

# 데이터 분석 - 기본

---

## - 목차 -

1. 개요
2. Multivariate Analysis
3. Cluster Analysis
4. Anomaly Detection
5. Association Analysis
6. Sequence Patterns
7. Classification
8. Prediction

# 데이터프레임 추가하기

통계함수를 사용할 때마다 데이터 프레임을 입력하는 번거로움을 방지하기 위해서는 `attach`를 이용한다.

## 데이터 프레임 추가

### □ 함수

`attach()`

### □ 사용예

```
> summary(mtcars$mpg)
```

```
> plot(mtcars$wt, mtcars$mpg)
```

```
> #attach를 이용하여 간단하게 코딩하기
```

```
> attach(mtcars)
```

```
> summary(mpg)
```

```
> plot(wt, mpg)
```

## 데이터 프레임 제거

### □ 함수

`detach()`

### □ 사용예

```
> attach(mtcars)
```

```
> summary(mpg)
```

```
> plot(wt,mpg)
```

```
> # detach를 이용하여 추가한 데이터프레임을 제거하기
```

```
> detach(mtcars)
```

# 통계함수 반복 적용하기

통계함수를 행/열을 대상으로 반복 적용하기 위해서는 하기와 같이 apply를 이용한다.

## 행 대상 반복 적용

□ 함수  
apply(x, margin, fun(),...) margin  
n이 1이면 행대상 적용

□ 사용예

```
> a <- array(1:12, dim = c(3,4))  
> a  
      [,1] [,2] [,3] [,4]  
[1,]    1    4    7   10  
[2,]    2    5    8   11  
[3,]    3    6    9   12
```

> a<-array(1:12,dim=c(3,4))  
> apply(a,1,max)

□ 결과

[1] 10 11 12

```
> apply(a, 1, max)
```

	[,1]	[,2]	[,3]	[,4]	결과값
[1,]	1	4	7	10	10
[2,]	2	5	8	11	11
[3,]	3	6	9	12	12

## 열 대상 반복 적용

□ 함수  
apply(x, margin, fun()) margin  
이 2이면 열대상 적용

□ 사용예

```
> apply(a, 2, max)
```

□ 결과

[1] 3 6 9 12

```
> apply(a, 2, max)
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	4	7	10
[2,]	2	5	8	11
[3,]	3	6	9	12

결과값 3 6 9 12

# 기초 연산

R에서는 하기와 같은 수학함수를 제공한다.

## 기초 수학 계산

### □log 함수

- log(3)            # 1.098612 (자연로그)  
- log10(100)      # 2 (상용로그)

### □버림/올림/반올림

- floor(3.14)        # 3 (버림)  
- ceiling(5.87)      # 6 (올림)  
- round(4.65)        # 5 (반올림)

### □최소/최대/합계

- min(c(1,5,7))      # 1 (최소값)  
- max(c(2,6,9))      # 9 (최대값)  
- sum(c(1,2,4,7))    # 14 (합계)

### □기타(sin, sqrt, runif)

- sin(pi)            # 1.224606e-16  
- sqrt(9)            # 3 (제곱근)  
- runif(2, 1, 10)    # 난수 2개를 벡터로 생성(1 초과 10 미만)

## 미분과 적분 계산

### □편미분

```
>f<-expression(2*x^3-y*x^2+2*y^2+1)
>D(f,"x")
2 * (3 * x^2) - y * (2 * x)
>D(f,"y")
2 * (2 * y) - x^2
```

### □정적분

```
>f <- function(x) 2*x^3 - 3*x^2 + 1
>integrate(f, 0, 3)
16.5 with absolute error < 1.8e-13
```

# 데이터 탐색 - 통계

데이터셋을 대상으로 통계값을 계산하는 방법은 하기와 같다.

## 주요 통계값 계산

### □ 평균/표준편차/분산/최빈값

- mean() : 평균
- colMeans() : 여러 열의 평균
- sd() : 표준편차
- var() : 분산
- Mode() : 최빈값(prettyR패키지 사용)
- mad() : 중위수 절대 편차(MedianAbsolute Deviation)
- cov() : 공분산

### □ 사분위수

- median() : 중위수
- IQR() : 사분위범위(3번째 분위값 - 1번째 분위값)
- range() : 범위
- quantile() : 사분위
- fivenum() : 사분위(Turkey five-number summary)

## 주요 통계값 계산

### □ 주요 통계값 계산

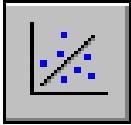
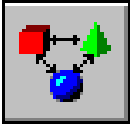

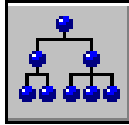
- scale() : 정규화(원래 값 - 평균 / 표준편차)
- sd() / mean() : 변동계수(표준편차 / 산술평균)
- cor() : 상관분석
- skewness() : 왜도(fBasics패키지 사용)
- kurtosis() : 첨도(fBasics패키지 사용)
- summary() : 최대, 최소, 사분위, 평균, 빈도 계산
- describe() : psych패키지 함수로서 표본수, 평균, 표준편차, 중위수, 최소, 최대, 범위, 왜도, 첨도 등을 계산함

### □ 사칙연산/관측치 개수

- sum() : 합
- prod() : 곱
- length() : 관측치 개수

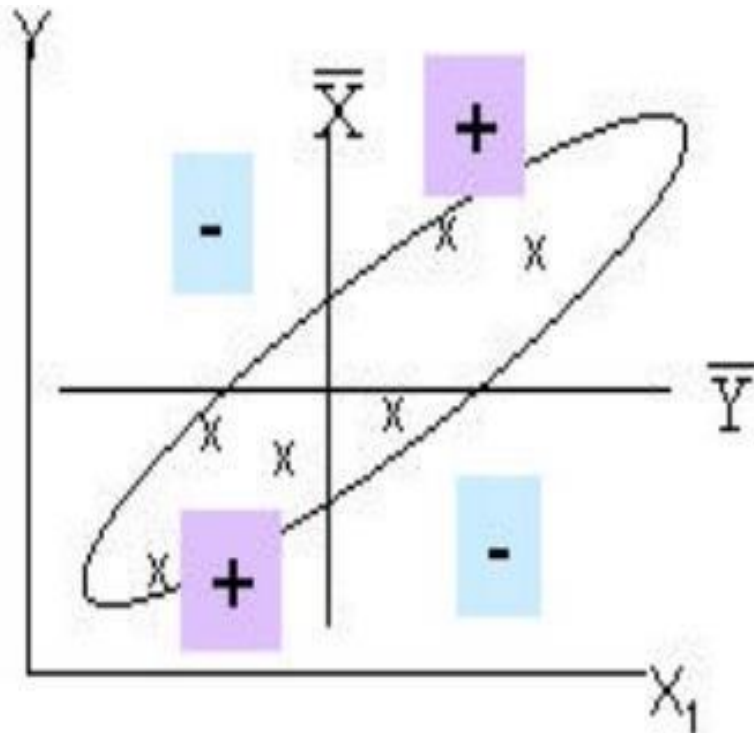
# 데이터 분석 유형

일반적으로 데이터 분석은 예측, 연관성, 분류/군집화, 인과관계 분석으로 구분할 수 있다.

	<b>Prediction</b> 	<b>Association</b> 	<b>Segmentation</b> 	<b>Cause &amp; Effect</b> 
분석 목적	<b>Prediction</b> 과거Data에 근거하여 제품품질특성이나 미래시장점유율 예측	<b>Association Rule</b> 빈도/확률에 근거하여 자주 발생하는 2개 이상의 사건을 도출	<b>Classify &amp; Cluster</b> 사전 정의된 기준이나 유사한 특성 기준에 근거하여 Data를 분할하고 특성을 파악	<b>Feature Selection</b> 목표변수에 대한 유의한 영향인자를 찾아냄
분석 문제 사례	부품의 설계Spec.을 A→B로 변경할 경우 성능값은 어떻게 변하게 될 것인가?	결혼한 20대 남자 회사원에게 쇼핑할인쿠폰을 보낼때 가장 효과적인 아이템은 무엇인가?	어떤 특성의 고객이 카드비용 연체를 자주 하는가?	제품의 수율저하/불량을 발생시키는 공정 or 장비는 무엇인가?
활용 분야	제품 성능/수율 예측 판매전략 수립 수요 예측	고객 구매행태 분석 제품불량원인 도출 S/W Bug Localization	제품 합격/불량 판정 신용평가 유사구매행태 고객분류 유사어 분류	성능 주인자도출 제품불량원인 도출

# 다변량 - Correlation

두 변수가 선형적으로 상호 어떠한 관계(정비례, 반비례)가 있는지를 분석한다.



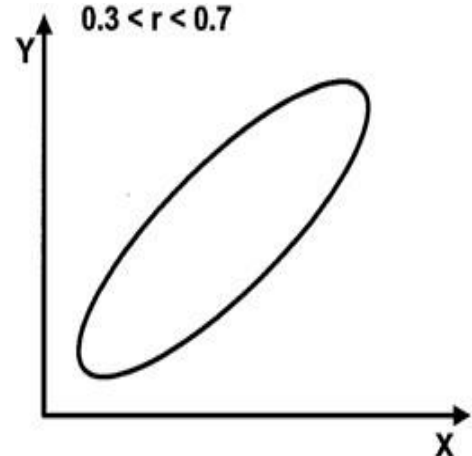
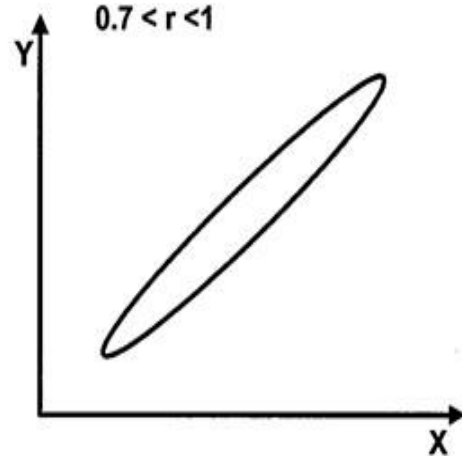
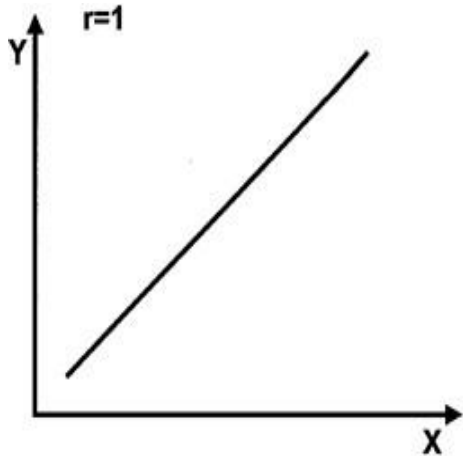
$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}$$

- ◆  $r=0$  이면 상관관계가 없으며,  
 $r=+1$  또는  $r=-1$ 이면 완전한 상관관계가 있다.
- ◆  $r=0.65$  이상이면 상관관계가 있다고 말할 수 있다.  
(이론적인 기준임)

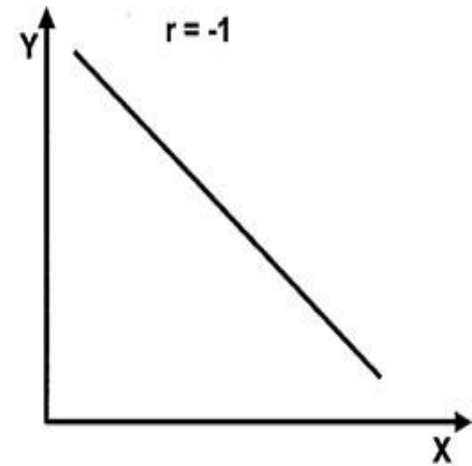
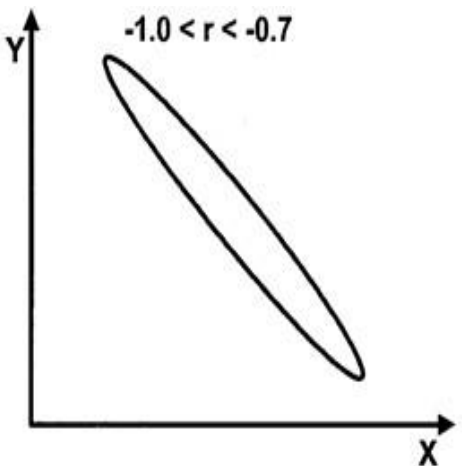
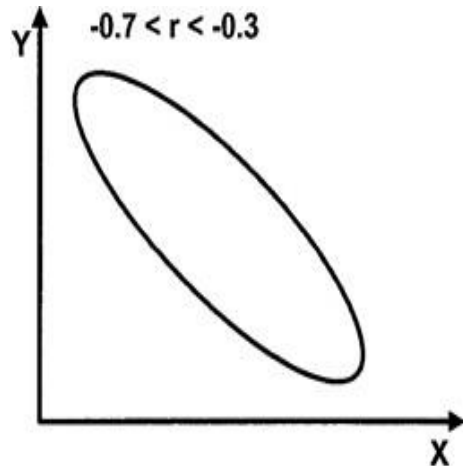
# 다변량 - Correlation

두 변수간의 상관관계는 양 또는 음의 상관관계가 존재한다.

양의  
상관  
관계



음의  
상관  
관계





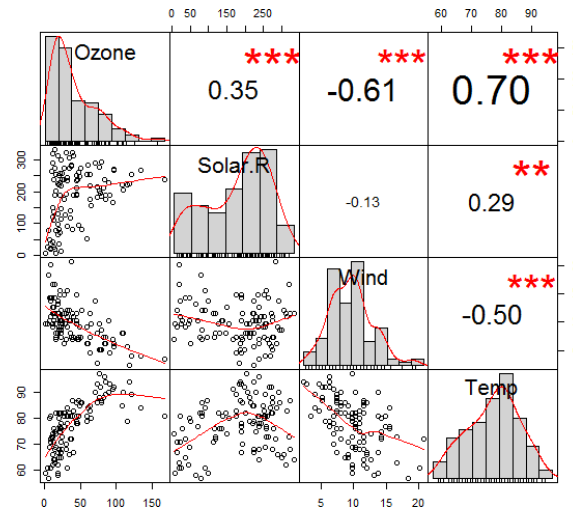
# 다변량 - Correlation

여러 개의 변수간 상관관계 분석을 위해서는 하기와 같은 명령을 실행한다.

## 코드

```
> aq1 <- airquality[,c(1:4)]
> View(aq1)
> #상관분석
> cor(aq1)
> #결측치제거하기
> aq2 <- na.omit(aq1)
> #상관분석
> cor(aq2)
> #산점도그리기
> plot(aq2)
> #추세선그리기
> pairs(aq2, panel=panel.smooth)
> 산점도에 히스토그램 그리기
> install.packages("PerformanceAnalytics")
> library(PerformanceAnalytics)
> chart.correlation(aq2, histogram=TRUE, pch=19)
> 그래프에 상관계수 표현하기
> library(corrplot)
> corrplot(cor(aq2), method="number")
```

## 실행결과

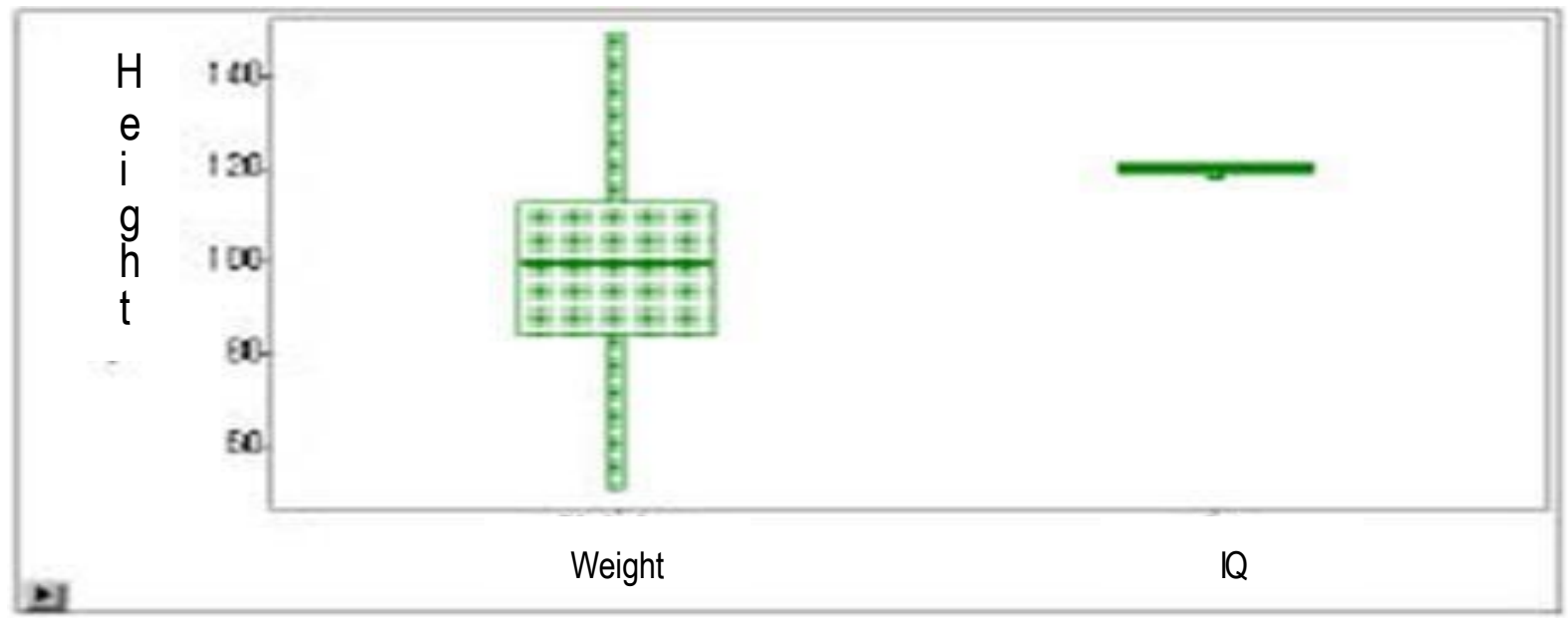


	Ozone	Solar.R	Wind	Temp
Ozone	1	0.35	-0.61	0.7
Solar.R	0.35	1	-0.13	0.29
Wind	-0.61	-0.13	1	-0.5
Temp	0.7	0.29	-0.5	1

# 다변량-PCA

변수의 차원을 축소하기 위한 목적으로 변동(분산, 공분산)을 잘 설명하는 주성분을 찾아내는 분석 방법이다.

정보의 Value측면에서 어느 변수가 더 바람직한가?



IQ정보보다는 Weight의 정보가 보다 더 활용성이 높다.  
왜냐하면 IQ로는 신장(Height)의 크고 작음을 분별하는 능력이 떨어지지만  
Weight로는 분별력이 있다.

주성분분석은 상관관계가 있는 변수들을 선형결합하여 변수를 축약하는 방법이다.

## 분석 목적

### □ 분석목적

- 소수의 주성분으로 차원을 축소함  
(여러 변수들 간에 내재하는 상관관계, 연관성을 이용)
- 다중공선성의 문제해결  
(상관성이 적은 주성분으로 변수들을 축소)
- 군집분석의 결과와 연산속도 개선  
(차원을 축소하여 군집분석 수행)
- 시계열 분포/추세의 변화 분석하여 고장징후 파악  
(다량의 센서 데이터를 주성분분석으로 차원축소)

## 주성분 분석 유의사항

### □ 유의사항

- 주성분 분석은 척도에 영향을 받음  
→ 변수들의 선형결합 유도시 분산을 이용함  
→ 측정단위의 크기에 좌우됨
- 공분산행렬: 모든 변수들이 동일 수준으로 점수화되는 경우 사용
- 상관계수행렬: scale이 다양한 경우 값이 큰 변수가 전체 경향을 좌우하는 것을 방지하기 위해 사용함

### □ 주성분 선택법

- 주성분 분석결과에서 누적기여율이 85% 이상인 주성분까지 선택
- Scree Plot에서 고유값이 수평을 유지하기 전단계까지의 주성분 개수를 선택함

우리는 아래와 같이 실생활에서 주성분 분석을 이용하고 있다.

## □기성복 바지를 살때, 우리 몸의 치수를 모두 알아야 하나?

일반적으로 허리 둘레와 기장만 알고 있으면 충분하다.

바지를 사려면 허리둘레, 기장 이외에 엉덩이 둘레, 허벅지 둘레, 무릎 높이 등

다른 하체에 대한 정보가 있어야 할 것 같지만 변수(허리,기장) 정보만 가지고 기성복을 사 입어도 잘 맞는다.

## □왜 허리,기장 변수만으로 구입이 가능한가?

하체에 대한 많은 체형 측정변수들이 PCA방법에 의해 2개의 변수로 축약되어 있고,

이 변수가 다른 체형의 정보를 대체하기 때문이다.

따라서 허리와 기장은 주성분이라고 할 수 있다.

## □주성분분석에서 이상치(Outlier)는?

하체에 대한 정보(허리, 기장)로 Large Size로 판단되었으나, 실제로 XL가 맞는다면

이 사람은 『이상치(Outlier)』에 해당한다고 볼 수 있다.

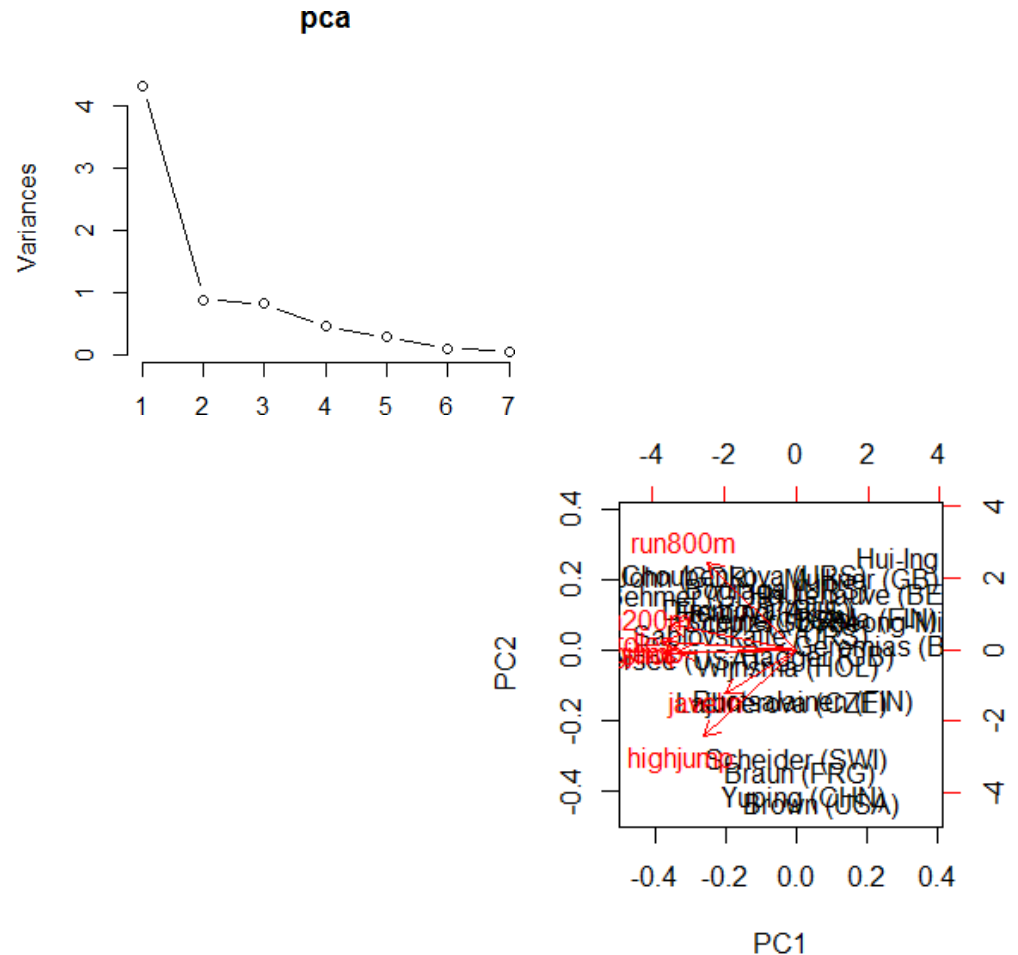
## 다변량-PCA

**주성분 분석을 위해서는 하기와 같은 명령을 실행한다.**

## 코드

```
> library(MVA)
> df<-heptathlon
> df
> #달리기 기록은 작을수록 좋으므로 이를 반대로 만든다.
> df$hurdles<-with(df,max(hurdles)-hurdles)
> df$run200m<-with(df,max(run200m)-run200m)
> df$run800m<-with(df,max(run800m)-run800m)
> df
> #주성분 분석 수행
> df2<-df[df$hurdles>0,]
> pca<-prcomp(df2[,-8],scale=T)
> summary(pca)
Importance of components:
               PC1    PC2    PC3    PC4    PC5
Standard deviation  2.0793 0.9482 0.9109 0.68320 0.54619
Proportion of Variance 0.6177 0.1284 0.1185 0.06668 0.04262
Cumulative Proportion 0.6177 0.7461 0.8646 0.93131 0.97392
.....
> pca
> #Scree Plot표시
> plot(pca,type="l")
> #pc1과 score 간 상관관계 분석
> cor(df2$score, pca$x[,1])
[1] -0.9931168
> biplot(pca)
```

## 실행결과



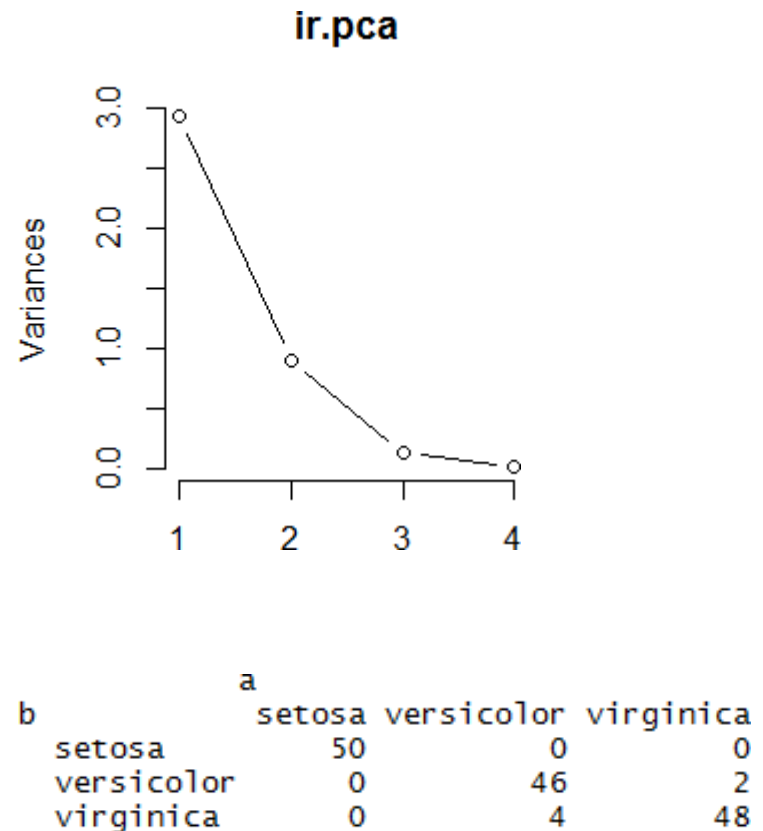
# 다변량-PCA

주성분 분석을 위해서는 하기와 같은 명령을 실행한다.

## 코드

```
> log.ir<-log(iris[,1:4])
> ir.species<-iris[,5]
>#변수별 scale이 다르므로 표준화하기 위해 scale.=T로 설정
> ir.pca<-prcomp(log.ir,center=T, scale.=T)
> plot(ir.pca,type='l')
> prc<-as.matrix(log.ir) %*% ir.pca$rotation
> head(prc)
      PC1      PC2      PC3      PC4
[1,] -0.2772209 -1.809493 1.604387 -1.0010840
[2,] -0.2507663 -1.654229 1.627078 -0.9946772
....
> train1<-cbind(ir.species,as.data.frame(prc))
> train1[,1]<-as.factor(train1[,1])
> colnames(train1)[1]<-'label'
> fit1<-lm(label~PC1+PC2,data=train1)
> fit1_pred<-predict(fit1,newdata=train1)
> b<-round(fit1_pred)
> b[b==0|b==1]<-'setosa'
> b[b==2]<-'Versicolor'
> b[b==3]<-'Virginica'
> a<-ir.species
> table(b,a)
```

## 실행결과



# 다변량-FA

데이터내의 변수들간의 내재된 상관관계를 이용하여 요인을 구하고 의미를 분석한다.

첫째, 변수들을 분류하여 변수그룹(요인, Factor)을 도출하고

둘째, 해당 변수그룹에 적절한 의미를 부여한다.

다변량 데이터는 개체들의 특성을 측정한 변수들이 유사한 경우가 많고 서로 상관관계를 맺고 있어 상관계수만으로는 과목의 구조적 특성을 파악하기 어렵다.

	Classic	French	English	Math	Discovery	Music
Classic	1	.83	.78	.7	.66	.63
French		1	.67	.67	.65	.57
English			1	.64	.54	.51
Math				1	.45	.51
Discovery					1	.4
Music						1

요인분석은 서로 관련이 있는 변수들을 대상으로 해당 변수들을 설명할 수 있는 새로운 공통변수를 파악하는 통계적 분석방법이다.

## 목적과 사례

### □ 분석목적

- 차원을 축소하여 대상을 파악함
- 차원축소를 통한 다중공선성의 문제 해결함

### □ 분석 사례

- Data
  - 100명의 학생에 대한 시험성적  
(국어, 영어, 수학, 사회, 지리, 역사, 물리, 화학, 생물)
- 사례
  - 과목 9개가 아닌 공통으로 설명가능한 인자 도출
  - 예) 언어능력(국어, 영어), 수리능력(수학, 물리) 등

## 주성분 분석과 비교

### □ 공통점

- 관측된 여러 개의 변수로부터 소수의 새로운 변수 생성
- 차원 축소방법으로 활용

### □ 차이점

- PCA: 원 변수의 변동을 가장 잘 설명하는 주성분을 찾음  
계산한 주성분간에 중요도 순서가 존재함  
(제1주성분이 제2주성분보다 변동을 더 잘 설명함)
- FA: 원 변수의 내재된 관계를 이용하여 변수를 분류함  
계산한 요인들은 기본적으로 대등한 관계를 가짐



FA모형의 종류 / 차이점은 하기와 같으며, 요인의 수와 유의성 판단기준은 하기와 같다.

## 요인모형 추정 종류 및 차이점

### □ 주성분인자법

- 관측값  $X$ 의 분산, 공분산 행렬, 상관계수행렬  $R$ 의 고유근과 고유벡터를 이용하여 인자부하값과 특수분산을 추정하는 방법
- 주성분 분석과 같은 과정을 가짐

### □ 최우추정법

- $X$ 가 다변량 정규분포를 따른다는 가정을 함
- 추정의 신뢰성이 높아 많이 사용되는 방법임

## 요인의 수와 유의성 판단 기준

### □ 요인의 수

- 상관계수행렬  $R$ 의 고유값이 1이상인 경우 채택한다.

### □ 요인의 유의성 기준

- 수학적 근거보다는 통상적 개체수  $n \geq 50$ 인 경우 절대값 기준으로
  - 요인부하(loadings)값  $> 0.3$  : 유의함
  - $> 0.4$  : 좀 더유의함
  - $> 0.5$  : 아주 유의함

주성분 요인법을 이용한 요인분석을 위해서는 하기와 같은 명령을 실행한다.

## 코드

```
> med.data<-read.table('d:/medFactor.txt',header=T)
> head(med.data)
> summary(med.data)
> str(med.data)
> # 요인분석을 위한 패키지 설치
> install.packages(c('psych','GPArotation'))
> library(psych)
> library(GPArotation)
> med.factor<-principal(med.data,rotate='none')
> names(med.factor)
> med.factor$values
> plot(med.factor$values, type='b')
```

- 데이터 : 무료 검진 프로그램인 Positive Health Inventory(PHI) 관련 데이터
- 변수개수 : 11개(검진 항목)
- 데이터 개수 : 128명의 검진결과값



medFactor.txt

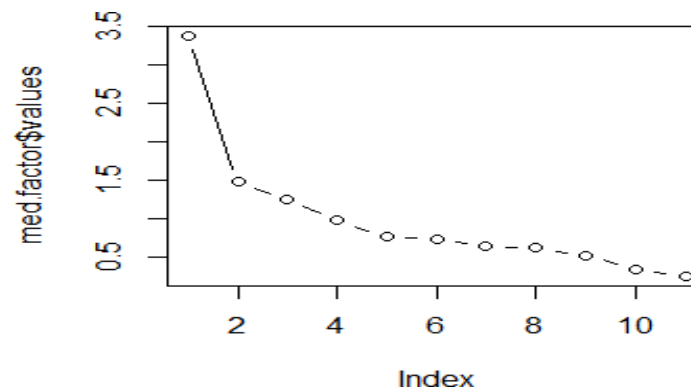
※ medFactor.txt파일 (<https://drive.google.com/open?id=1jVA5dyUkH1JdGOcYQDGAwIldKGSGfwi4>)

## 실행결과

### ■ med.factor\$values의 값

```
[1] 3.3791814 1.4827707 1.2506302 0.9804771 0.7688022 0.7330511
[7] 0.6403994 0.6221934 0.5283718 0.3519301 0.2621928
```

### ■ Plot(med.factor\$values, type='b')



- 위 그래프에서 y축값인 med.factor\$values는 고유근을 의미함
- 인자수 채택은 고유근의 값이 1이상인 경우까지만을 대상으로 함
- 따라서 예제에서는 3개 인자만 선택하도록 함  
(index 3의 값이 1.2506302이고 이후는 1보다 작음)

요인분석을 통하여 선정한 3개 요인에 대해 직교회전(varimax)을 실행하여 개별 변수에 대한 해석과 유사한 변수끼리 그룹화 할 수 있다.

## 코드

```
> # 요인은 앞에서 분석한대로 3개를 선정함
> # 회전방법은 직교회전방법중 하나인 varimax를 사용함
> # 회전이유는 개별변수의 해석을 용이하게 하기 때문임
> med.Varimax = principal(med.data, nfactors = 3, rotate="varimax")
> med.Varimax

Principal Components Analysis
Call: principal(r = med.data, nfactors = 3, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	RC1	RC2	RC3	h2	u2	com
lung	0.66	0.12	0.16	0.47	0.53	1.2
muscle	0.11	-0.09	0.79	0.64	0.36	1.1
liver	0.78	0.13	0.17	0.66	0.34	1.1
skeleton	0.19	0.29	0.76	0.70	0.30	1.4
kidneys	0.73	0.23	-0.14	0.61	0.39	1.3
heart	0.65	-0.11	0.19	0.46	0.54	1.2
step	0.49	0.48	0.10	0.48	0.52	2.1
stamina	0.02	0.62	0.29	0.47	0.53	1.4
stretch	0.18	0.65	0.34	0.57	0.43	1.7
blow	0.26	0.70	-0.04	0.56	0.44	1.3
urine	-0.07	0.65	-0.28	0.50	0.50	1.4

## 실행결과



medFactor.txt

	RC1	RC2	RC3
SS loadings	2.39	2.13	1.59
Proportion Var	0.22	0.19	0.14
Cumulative Var	0.22	0.41	0.56
Proportion Explained	0.39	0.35	0.26
Cumulative Proportion	0.39	0.74	1.00

The root mean square of the residuals (RMSR) is 0.1  
with the empirical chi square 142.78 with prob < 1.8e-18 Fit  
based upon off diagonal values = 0.85

- **Proportion Var에 의하면 개별 요인의 총 변동에 대한 설명력은 아래와 같음**
  - RC1 : 총 변동의 22%
  - RC2 : 총 변동의 19%
  - RC3 : 총 변동의 14%
  - 3개 요인이 총 변동의 55%를 설명한다고 볼 수 있음
- **RC1, RC2, RC3의 Standardized loadings값을 볼 때 다음과 같이 그룹화가 가능함**
  - RC1은 lung, liver, kidney, heart가 높은 값을 갖음 → 생물의학
  - RC2는 stamina, stretch, blow, urine이 높은 값을 갖음 → 인체기능
  - RC3은 muscle과 skeleton에서 높은 값을 갖음 → 근육골계통력

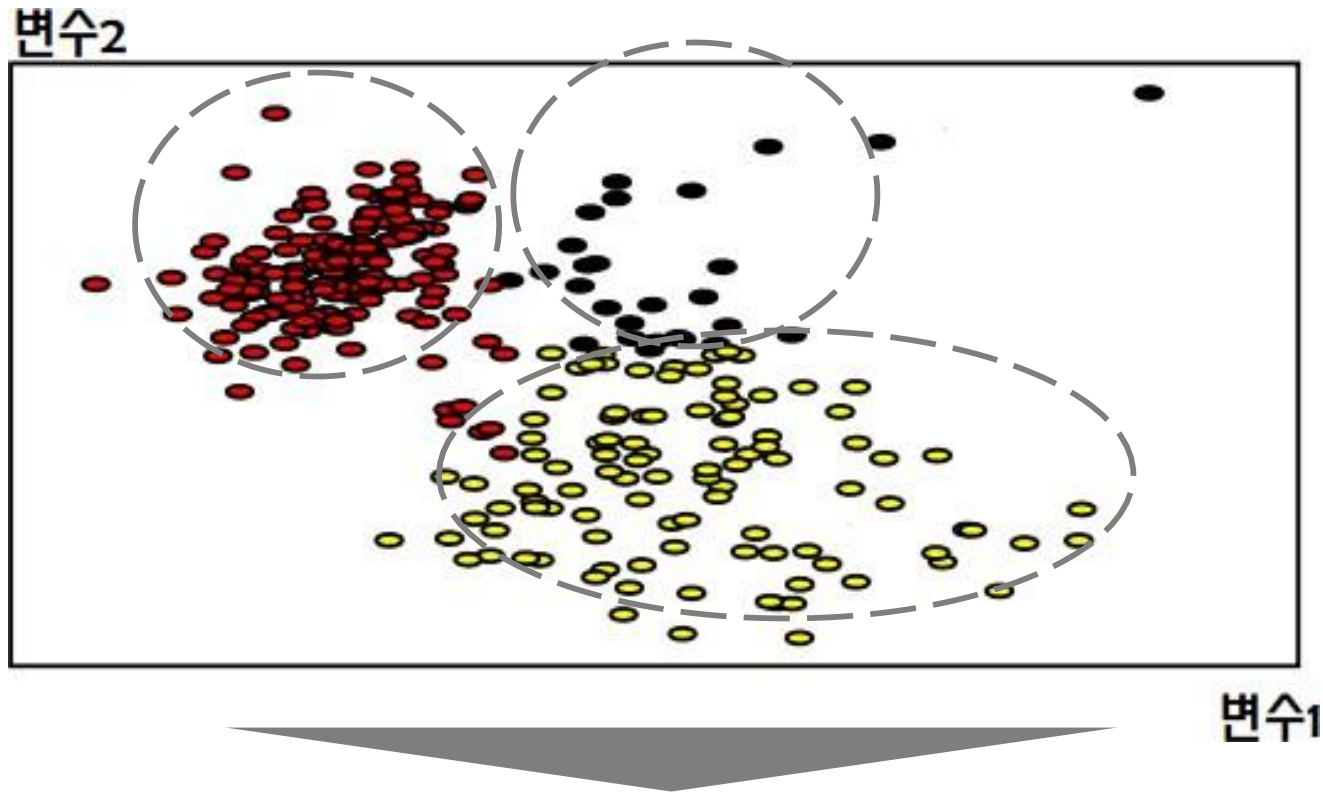
요인분석과 주성분분석의 차이는 아래와 같다.

주성분 분석	요인 분석
주성분은 원 변수의 직교 선형 결합으로 표현 $Y=LX$ , $L$ 은 선형계수 행렬	인자들의 직교 선형 결합으로 원 변수들을 표현 $X=LF$ , $L$ 은 부하행렬, $F$ 는 관측불가
주성분은 변수들의 변동을 설명한다.	요인은 변수들의 분산-공분산 구조 설명한다.
요인분석이나 주성분 분석의 $L$ 을 구하는 방법 유사하다. 공분산 행렬, 상관 행렬로부터 고유치 그에 대응하는 고유 벡터를 이용	
행렬 $L$ 은 변수의 개수 축약하는데 사용되며 는 주성분의 이름을 붙이는데 사용	행렬 $L$ 은 변수에 내재된 관계를 알아보는데 사용되며 는 변수들을 그룹화 하는데 사용한다.
적절한 주성분의 수를 구하고 주성분의 이름을 부여하고 주성분들간 산점도로 이상치 발견하거나 각 주성분 점수에 의해 개체 순위	적절한 인자의 수를 구하고 이를 이용하여 변수들을 그룹화 하고 그룹을 이용하여 변수에 내재된 관계를 알아본다.

변수들의 구조를 파악하기 위해 공분산 행렬을 이용하는 것은 동일하지만  
주성분 분석은 원변수의 차수 축약, 요인 분석은 원변수를 그룹화하는 목적으로  
수행하는 것이 서로 다르다.

# ClusterAnalysis

개별 데이터 개체를 유사도 기준에 근거하여 집단으로 그룹화하여 각 집단의 성격을 파악함으로써 데이터 전체의 구조에 대한 이해를 얻는 분석기법이다.



여러 개의 변수(변수1, 변수2, ...)를 기준으로 각 데이터의 유사도를 계산한 후 이에 근거하여 집단으로 그룹화하여 각 집단의 특성을 파악한다.

# Cluster Analysis

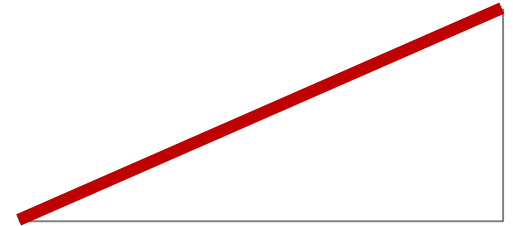
개체*i*, 개체*k*에 해당하는 대표적인 유사도 계산 방법은 아래와 같다.

(개체는 *p*개의 변수로 표현할 수 있다고 가정함)

유클리드(Euclidean)거리

최단거리, 가장 많이 사용하는 방법

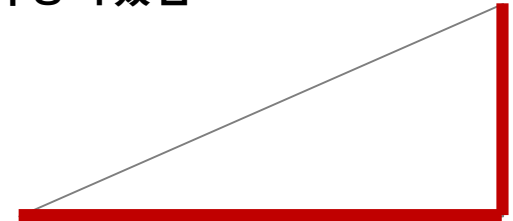
$$d(i,k) = \sqrt{\sum_{j=1}^p (x_{ij} - x_{kj})^2}$$



맨하탄(Manhattan) 거리

직선이동거리, 이상치 비중이 약해지는 특성이있음

$$d(i,k) = \sum_{j=1}^p |x_{ij} - x_{kj}|$$



피어슨(Pearson) 거리

거리를 변수의 분산으로 나누어 표준화함

$$d(i,k) = \sqrt{\sum_{j=1}^p \frac{(x_{ij} - x_{kj})^2}{s_j^2}}$$

# ClusterAnalysis - kmeans

군집 분석은 사전 정보 없이 자료를 컴퓨터에게 주고, “유사한 대상끼리 묶어보아라!” 라고 명령을 내리는 분석이다. 입력 Parameter는 하기와 같다.

## Parameter

```
kmeans(x, centers, iter.max = 10,  
       nstart = 1,  
       algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"),  
       trace=FALSE)
```

## Property

x : 분석대상 데이터 행렬

**centers : 군집수**

iter.max : 군집화 반복회수

nstart : 사용할 초기 랜덤셋

algorithm : 유사도 계산방법(Hartigan-Wong, Lloyd, Forgy, MacQueen)

trace : 현 알고리즘을 수행결과 정보

cluster : 원시데이터에 대한 군집배정결과

centers : 군집중심

totss : 총 제곱합

withinss : 군집내 제곱합 벡터

tot.withinss : withinss의 총합

betweenss : 군집간 제곱합

size : 각 군집내 데이터 개수

iter : 군집화 반복회수

auct : 전문가 진단

# ClusterAnalysis - kmeans

kmeans분석을 하기 위한 명령어와 실행결과는 하기와 같다.

## 코드

```
> x <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),  
              matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))  
> fit<-kmeans(x,10)  
> fit
```

## 실행결과

K-means clustering with 10 clusters of sizes 8, 4, 14, 9, 8, 20, 10, 5, 14, 8

Cluster means:

[,1] [,2]

1 0.9493415 0.5135318

2 -0.5423244 0.1040697

...

10 0.1114832 -0.4078619

Clustering vector:

[1] 8 3 3 3 6 6 8 3 6 6 6 3 6 3 6 6 6 2 10 6 8 2 3 2

[26] 10 8 3 10 10 3 8 3 6 3 6 6 10 10 6 6 6 2 6 6 6 3 10 3 10

[51] 4 5 9 9 9 1 4 5 9 9 4 7 1 5 7 3 5 9 1 5 4 1 4 4 9

[76] 4 7 1 1 9 1 5 9 7 5 4 7 9 7 9 5 9 7 7 9 1 4 7 9

Within cluster sum of squares by cluster:

[1] 0.3253477 0.1443898 0.6251724 0.4166249 0.2070878 0.7081641 0.1917797

[8] 0.1652702 0.5336141 0.4072970

(between\_SS / total\_SS = 94.6 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss"

[6] "betweenss" "size" "iter" "ifault"

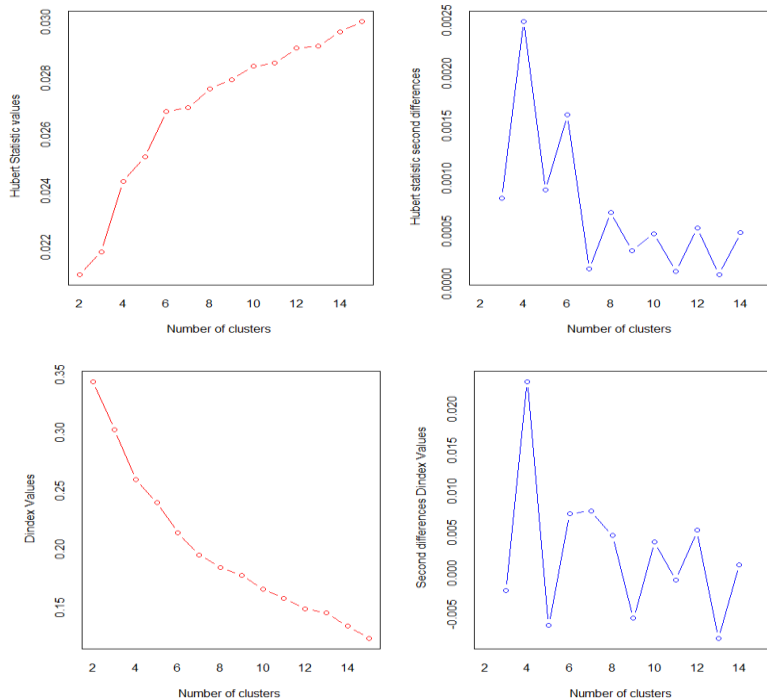


# ClusterAnalysis - kmeans

Kmeans에서 군집개수를 설정하는 방법은 하기와 같다.

## 코드

```
>#NbClust 패키지를 사용하는 방법
>install.packages('NbClust')
>library(NbClust)
>nc <- NbClust(x, min.nc = 2, max.nc = 15, method = "kmeans")
```



## 실행결과

\*\*\* : The Hubert index is a graphical method of determining the number of clusters.

In the plot of Hubert index, we seek a significant knee that corresponds to a significant increase of the value of the measure i.e the significant peak in Hubert index second differences plot.

\*\*\* : The D index is a graphical method of determining the number of clusters.

In the plot of D index, we seek a significant knee (the significant peak in Dindex second differences plot) that corresponds to a significant increase of the value of the measure.

\*\*\*\*\*

\* Among all indices:

\* 12 proposed 2 as the best number of clusters

\* 2 proposed 3 as the best number of clusters

...

\* 1 proposed 13 as the best number of clusters

\* 1 proposed 15 as the best number of clusters

\*\*\*\*\* Conclusion \*\*\*\*\*

\* According to the majority rule, the best number of clusters is 2

\*\*\*\*\*

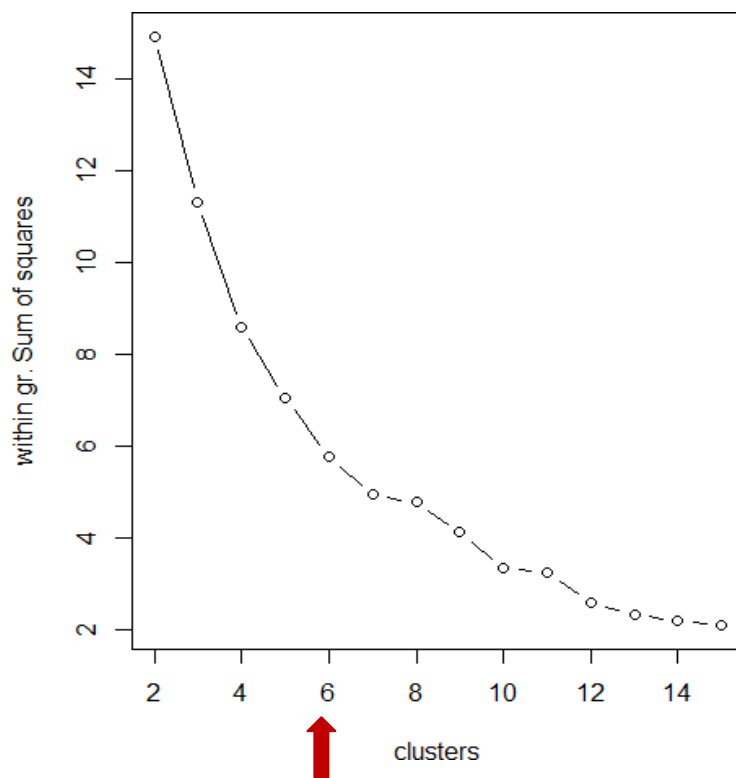
# ClusterAnalysis - kmeans

Kmeans에서 군집개수를 설정하는 방법은 하기와 같다.

## 코드

```
>#군집내 제곱합(withinss)를 이용하는 방법
>results<-matrix(nrow=14, ncol=2, dimnames=list(2:15,c("cluster s",
"within gr. Sum of squares"))))
>for(i in 2:15){
+   fit<-kmeans(x,i)
+   results[i-1,1]<-i
+   results[i-1,2]<-fit$tot.withinss
+ }
>plot(results, type='b')
```

## 실행결과



6개 이후 군집내 제곱합 변화가 적으므로  
군집개수 설정을 6으로 하는 것이 적합함

# ClusterAnalysis - kmedoids

kmeans분석은 군집중심을 평균값으로 취하기 때문에 이상치에 의한 영향을 많이 받는 이슈가 발생한다. 이와 같은 문제를 해결하기 위한 방법으로 kmedoids을 이용하며, 입력 parameter는 하기와 같다.

## Parameter

pam(x, k, diss, metric, medoids, stand, cluster.only, do.swap, keep, diss, keep.data, trace.lev)

x : 분석대상 데이터 행렬 or 유사도 행렬

k : 군집수

diss : x값이 데이터 행렬일 경우 False, 유사도 행렬인 경우 True

metric : 유사도 계산 방법(euclidean, manhattan)

medoids : Null값이 할당될 경우 여러 medoids가 도출된 상태임을 의미하고  
아닌 경우 초기 medoids상태임을 의미함

stand : x가 데이터 행렬이라면, 유사도 행렬 계산전에 x행렬값을 정규화함

cluster.only : True로 설정시, 군집만 계산하여 리턴함

do.swap : swap발생여부를 결정하는 불리언 값

keep.diss : 유사도 행렬을 결과객체에 포함시킬지 결정하는 불리언 값

keep.data : 분석 데이터를 결과객체에 포함시킬지 결정하는 불리언 값

trace.lev : 분석진행과정을 추적하는 수준을 결정하는 정수값으로서, 0은 추적안함

## 용어

### □ Centroid

- 군집내 데이터를 이용하여 인위적으로 생성한 군집의 중심
- kmeans의 결과
- centroid의 경우 이상치에 의한 영향을 많이 받는 단점이 있음

### □ Kmedoids

- 군집내 데이터중에서 선택하여  
해당 군집의 중심으로 선정함(kmedoids의 결과)

# ClusterAnalysis - kmedoids

kmedoids 분석을 하기 위한 명령어와 실행결과는 하기와 같다.

## 코드

```
> library(cluster)
> a<-c(1,2,1,2,1,3,2,2,3)
> b<-c(10,11,10,12,4,5,6,5,6)
> x<-data.frame(a,b)
> x
  a b
[1,] 1 10
[2,] 2 11
[3,] 1 10
[4,] 2 12
[5,] 1 4
[6,] 3 5
[7,] 2 6
[8,] 2 5
[9,] 3 6

> result<-pam(x,2,FALSE,'euclidean')
> summary(result)
> plot(result$data,col=result$clustering,pch=16)
```

## 실행결과

### Medoids:

ID a b

[1,] 3 1 10

[2,] 8 2 5

Clustering vector:

[1] 1 1 1 1 2 2 2 2 2

Objective function:

build swap  
1.0333959 0.9420787

Numerical information per cluster:

size max\_diss av\_diss diameter separation

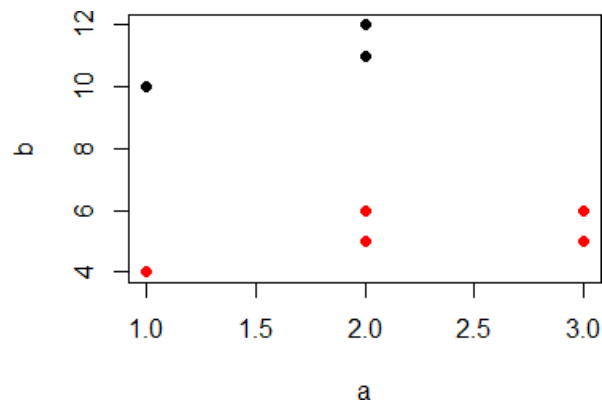
[1,] 4 2.236068 0.9125704 2.236068 4.123106

[2,] 5 1.414214 0.9656854 2.828427 4.123106

Isolated clusters:

L-clusters: character(0)

L\*-clusters: [1] 1 2



Silhouette plot information:

cluster neighbor sil\_width

2 1 20.7818773

1 1 20.7574186

3 1 20.7574186

4 1 20.7334521

8 2 10.7918605

6 2 10.7638253

7 2 10.7064211

9 2 10.6898229

5 2 10.6788165

Average silhouette width per cluster:

[1] 0.7575416 0.7261493

Average silhouette width of total data set:

[1] 0.7401014

36 dissimilarities, summarized :

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.000 1.414 4.298 3.819 6.000 8.062

Metric : euclidean

Number of objects : 9

Available components:

[1] "medoids" "id.med" "clustering"

[4] "objective" "isolation" "clusinfo"

[7] "silinfo" "diss" "call"

[10] "data"

# Cluster Analysis - Hierarchical

계층 형태로 data를 군집화하기 위한 분석방법은 Hierarchical Clustering로, 입력 parameter는 하기와 같다.

## Parameter

hclust(d, method = 'complete', members = null)

d : 행렬

method : 군집을 형성하는 방법으로

ward.D, ward.D2, single, complete, average,

mcquitty, median, centroid에서 선택

members : null, d(유사도 행렬)

- 최단연결법 : single로 설정
- 최장연결법 : complete로 설정
- 평균연결법 : average로 설정
- 중심연결법 : centroid로 설정
- 와드연결법 : ward로 설정
- 맥퀴티연결법 : mcquitty로 설정

## 방법

### ❑ Agglomerative(or Bottom up)

- 각 객체에서 시작하여 점차 짝(pair)끼리 묶어 상위단위까지 진행함

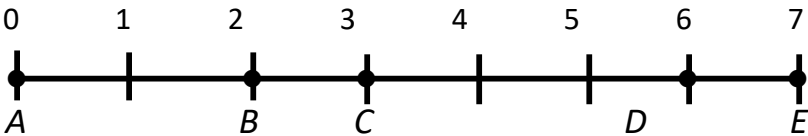
### ❑ Divisive(or Top down)

- 한 개의 군집에서 시작하여 군집을 쪼개어 최하위 단위인 객체까지 진행함

거리계산 방법은 하기와 같다.

□ 분석대상

- 데이터가 다음과 같다고 가정하고  
거리행렬(Distance Matrix)를 계산함



- 가장 가까운 거리계산 : B와 C

	A	B	C	D
B	2			
C	3	1		
D	6	4	3	
E	7	5	4	2

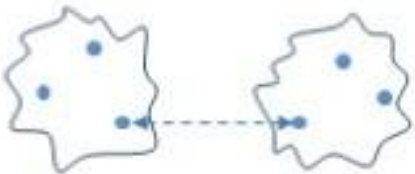
> c<-c(0,2,3,6,7)

> dist(c)

1 2 3 4  
2 2  
3 3 1  
4 6 4 3  
5 7 5 4 1

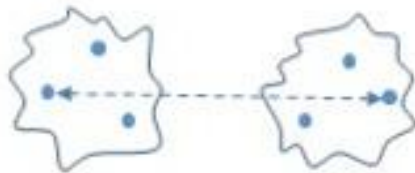
□ 최소연결법(단일결합법, Single linkage)

- 정의 : 가장 가까이에 있는 두 관측치 사이의 거리를 유사도로 계산함
- 계산 :  $d\{(BC), A\} = \min\{d_{BA}, d_{CA}\} = \min\{2, 3\} = 2$



□ 최장연결법(완전결합법, Complete linkage)

- 정의 : 가장 멀리 떨어진 두 관측치 사이의 거리를 유사도로 계산함
- 계산 :  $d\{(BC), A\} = \max\{d_{BA}, d_{CA}\} = \max\{2, 3\} = 3$



거리계산 방법은 하기와 같다.

## □ 평균연결법(Average linkage)

- 정의 : 하나의 군집 내에 있는 관측치들과 다른 군집 내에 있는 관측치들 사이의 모든 가능한 거리의 평균값을 유사도로 계산함  
극단값을 사용하지 않고 각 군집에 포함된 모든 구성원들의 값을 사용한다는 점에서 보다 합리적인 방법이라고 할 수 있음

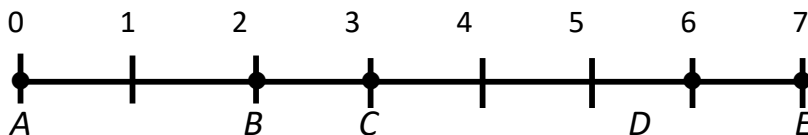
- 계산 :  $d\{(BC), A\} = 1/(\text{군집BC의 개체수} * \text{군집A의 개체수}) * (d_{BA} + d_{CA})$   
 $= 1/(2*1) * (2+3) = 5 / 2 = 2.5$



## □ 와드연결법(Ward linkage)

- 정의 : 두 군집간에 속하는 개체들의 평균으로부터 떨어진 정도인 “편차”의 제곱을 계산한 후 모두 합한 것을 유사도로 계산함.  
연결된 후에 군집 내 제곱합을 계산하여 합한 것을 유사도로 계산함

- 계산 :  $d\{(BC), A\} = (A-\text{avg})^2 + (B-\text{avg})^2 + (C-\text{avg})^2 = (0 - 1.67)^2 + (2 - 1.67)^2 + (3 - 1.67)^2 = 4.667$   
 $\text{average}(A, B, C) = (0 + 2 + 3) / 3 = 1.67$



## □ 맥퀴티연결법(Mcquitty linkage)

- 정의 : 산술평균으로 유사도를 계산함
- 계산 :  $d\{(BC), A\} = (d_{BA} + d_{CA}) / 2 = (2 + 3) / 2 = 2.5$

# Cluster Analysis - Hierarchical

계층 형태로 data를 군집화하기 위한 명령어와 실행 결과는 하기와 같다

## 코드

```
> dat<-matrix(rnorm(100),nrow=10, ncol=10)

[,1] [,2] [,3] [,4] [,5] [,6]
[1,] -0.41898010 0.1197176 0.57671878 1.0537509 1.580091684 -0.95583910
[2,] 0.99698686 -0.2821739 -1.28074943 -1.1195991 1.497818761 -1.23170706
[3,] -0.27577803 1.4559884 1.62544730 0.3356172 0.262645459 -0.95689188
[4,] 1.25601882 0.2290196 -0.50069660 0.4947958 -1.232901200 -0.86978287
[5,] 0.64667439 0.9965439 1.67829721 0.1380527 -0.003723534 -0.91068068
[6,] 1.29931230 0.7818592 -0.41251989 -0.1187920 1.511672283 0.74127631
[7,] -0.87326211 -0.7767766 -0.97228684 0.1976843 -0.475698284 0.06851153
[8,] 0.00837096 -0.6159899 0.02538287 -1.0686927 0.797916438 -0.32375075
[9,] -0.88087172 0.0465803 0.02747534 -0.8032132 -0.974002561 -1.08650305
[10,] 0.59625902 -1.1303858 -1.68018272 -1.1137651 0.689372698 -1.01592895

[,7] [,8] [,9] [,10]
[1,] -0.76779018 -0.25218316 0.7132405 0.59897515
[2,] -1.11972006 -0.86576375 -0.5428819 -1.52361488
[3,] -0.44817424 0.58258600 0.8857784 -0.20618900
[4,] 0.47173637 -0.01252935 -0.3485947 -0.57429541
[5,] -1.18049068 -0.37485476 -1.0080546 -1.39016604
[6,] 1.47025700 0.31788574 1.8831825 -0.07041738
[7,] -1.31142059 -0.48880563 -0.9289711 -0.43087953
[8,] -0.09652492 2.65865803 -0.2941965 -0.59222537
[9,] 2.36971991 1.68027820 -0.6149503 0.98111616
[10,] 0.89062648 0.77958401 -0.9470758 0.53240936
```

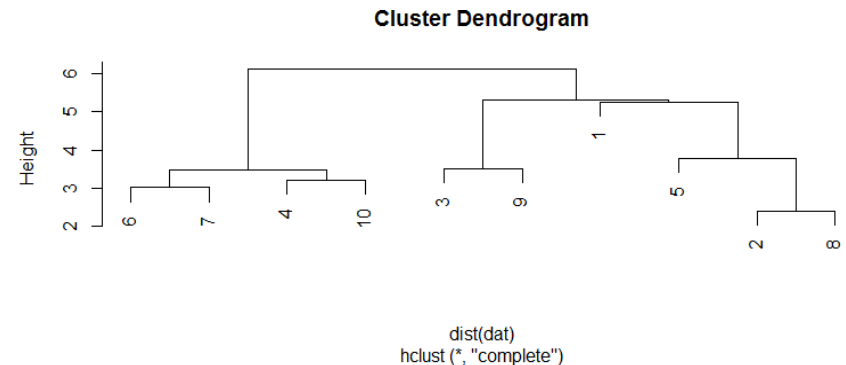
## 실행 결과

```
> hc<-hclust(dist(dat))
> hc
```

Call:  
hclust(d = dist(dat))

Cluster method : complete  
Distance : euclidean  
Number of objects: 10

```
> plot(hc)
```





# Cluster Analysis - Density Estimation

KMeans나 Hierarchical 군집화는 군집간의 거리를 이용하여 군집화하는 방법인 반면에 밀도기반의 군집화는 점이 몰려 있어 밀도가 높은 부분을 군집화하는 방법이다.

## 밀도추정 방법

### Pazen windows

윈도우내에 위치한 관측값을 이용하여 밀도추정을 하는 방법이다.

### Vector Quantization

관측값 분포에 따라 확률밀도함수를 모델링하는 방법이다.

### Histograms

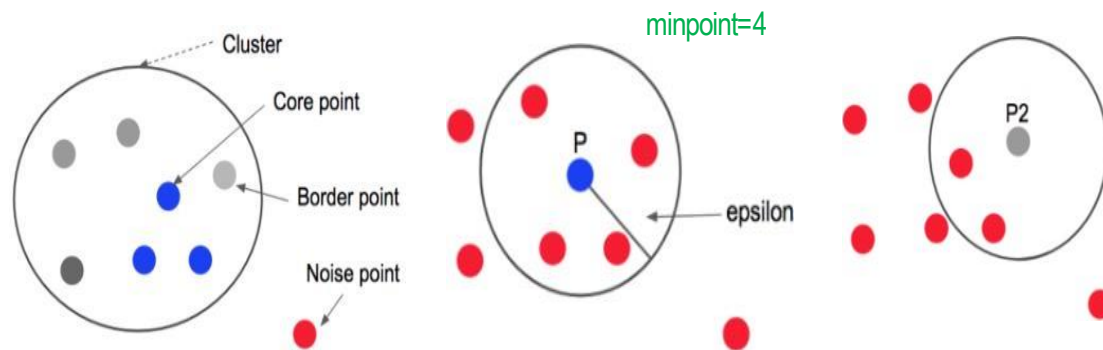
히스토그램을 이용하여 밀도값을 얻어낼 수 있으며, Bin의 개수에 따라 밀도추정결과가 달라진다.

## R에서 밀도추정

### DBSCAN(Density-based spatial clustering of applications with noise)

- 고정된 점 군집에 대한 클러스터링을 정해진 윈도우내에 위치한 관측값을 이용하여 밀도추정을 하는 방법이다.
- 어느점을 기준으로 반경  $x$  내에 점이  $n$ 개 이상 있으면 하나의 군집으로 인식하는 방식이다.

(고정된 point 군집에 대한 군집화 수행함)



# Cluster Analysis - Density Estimation

밀도 추정 군집화 알고리즘의 Parameter는 하기와 같다.

## Parameter

density(x, bw = "nrd0", adjust = 1, kernel = c("gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine", "optcosine"),  
weights = NULL, window = kernel, width, give.Rkern = FALSE, n = 512, from  
, to, na.rm = FALSE, ...)

x : 행렬

bw : 사용할 밴드폭

adjust: 밴드폭을 조정하기 위한 조정자

kernel : 사용대상 smoother 커널

(gaussian, rectangular, triangular, epanechnikov, biweight, cosine, optcosine)

weights : x와 동일한 길이의 관측 가중치 벡터

window : 사용 커널

width : S호환 파라미터

give.Rkern : TRUE이면 어떤 밀도도 추정하지 않음

n : 추정대상 밀도 포인트의 개수

from, to : 좌측, 우측의 최말단 포인트

na.rm : TRUE이면 결측치가 제거됨

## bw 함수 parameter

x : 데이터셋

nb : 빈의 개수

lower, upper: 최소화대상 밴드폭의 범위

method: 방정식을 해결하는데 사용하는 ste 메소드이나  
직접 플래그인에 사용되는 dpi 메소드

tol : ste에 대한 수렴오차

# Cluster Analysis - Density Estimation

밀도추정 방법으로 군집화하기위한 명령어와 실행결과는 하기와 같다.

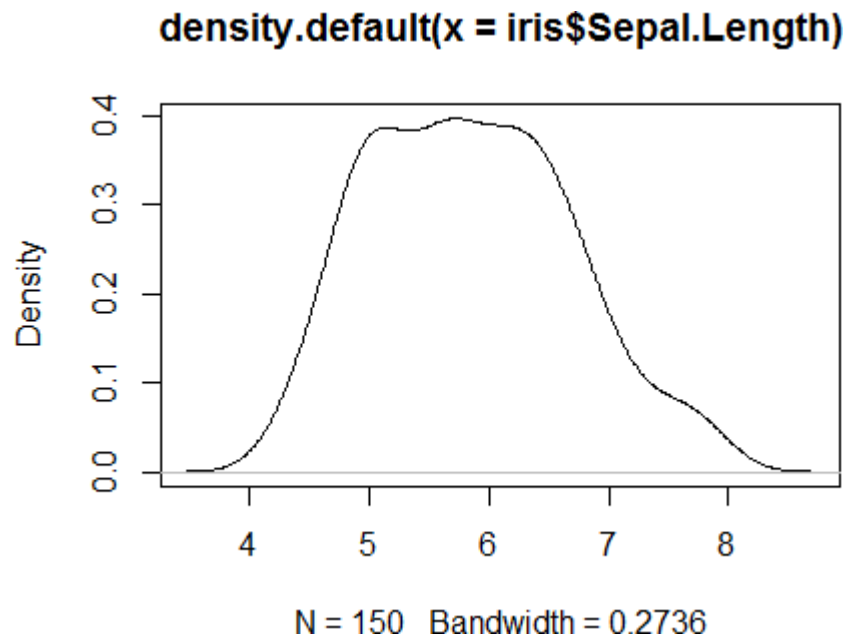
## 코드

```
> d<-density(iris$Sepal.Length)
> d
> plot(d)
```

대부분의 데이터는 5~7에 위치함

Sepal.Length의 평균은 대략 6에 해당함

## 실행결과



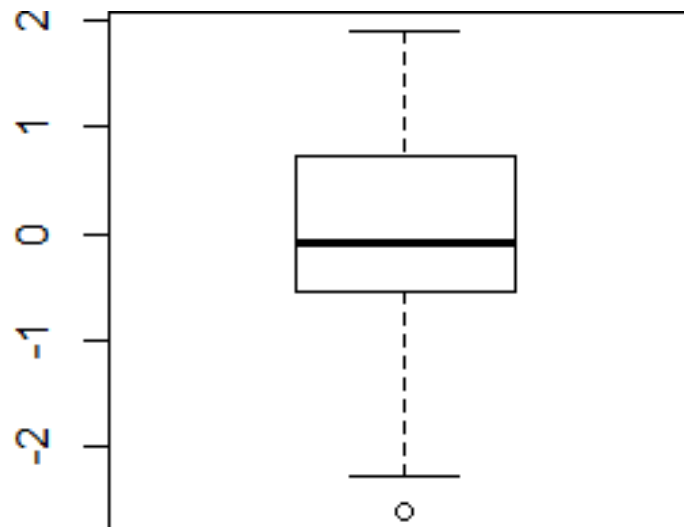
# Anomaly Detection – Show outliers

1차원의 경우 R에서 outlier를 확인하기 위해서는 Boxplot을 활용한다.

## 코드

```
> x<-rnorm(100)
> summary(x)()
> boxplotstats(x)$out
> boxplot(x)
```

## 실행결과



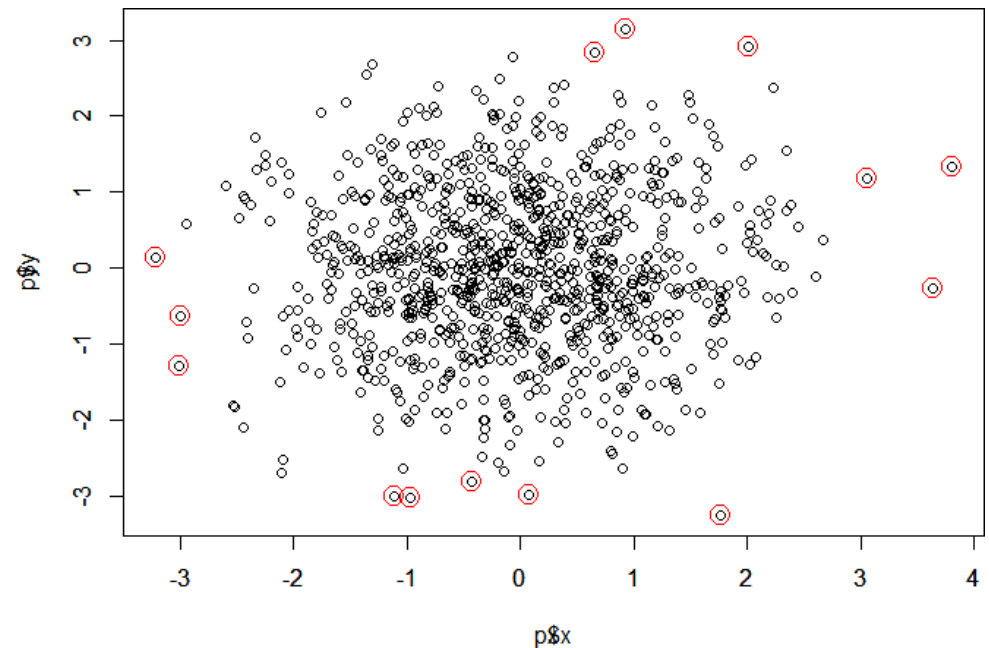
# Anomaly Detection – Show outliers

2차원의 경우 R에서 outlier를 확인하기 위해서는 scatter plot을 활용한다.

## 코드

```
> x <- rnorm(1000)
> y <- rnorm(1000)
> f <- data.frame(x,y)
> a <- boxplot.stats(x)$out
> b <- boxplot.stats(y)$out
> list <- union(a,b)
> plot(f)
> px <- f[f$x %in% a,]
> py <- f[f$y %in% b,]
> p <- rbind(px,py)
> par(new=TRUE)
> plot(p$x, p$y,cex=2,col=2)
```

## 실행결과



# Anomaly Detection – Calculating anomalies

R언어의 사용자 함수 정의기능을 이용하여 Anomalies계산을 자동화 한다.

## 코드

```
# Sepal.Length가 4.5미만, 7.5초과하면 이상치로 처리
outliers<-function(data, low, high){
  outs<-subset(data, data[,1] < low | data[,1] > high)
  return(outs)
}
outliers(iris, 4.5, 7.5)
```

## 실행결과

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
9	4.4	2.9	1.4	0.2	setosa
14	4.3	3.0	1.1	0.1	setosa
39	4.4	3.0	1.3	0.2	setosa
43	4.4	3.2	1.3	0.2	setosa
106	7.6	3.0	6.6	2.1	virginica
118	7.7	3.8	6.7	2.2	virginica
119	7.7	2.6	6.9	2.3	virginica
123	7.7	2.8	6.7	2.0	virginica
132	7.9	3.8	6.4	2.0	virginica
136	7.7	3.0	6.1	2.3	virginica

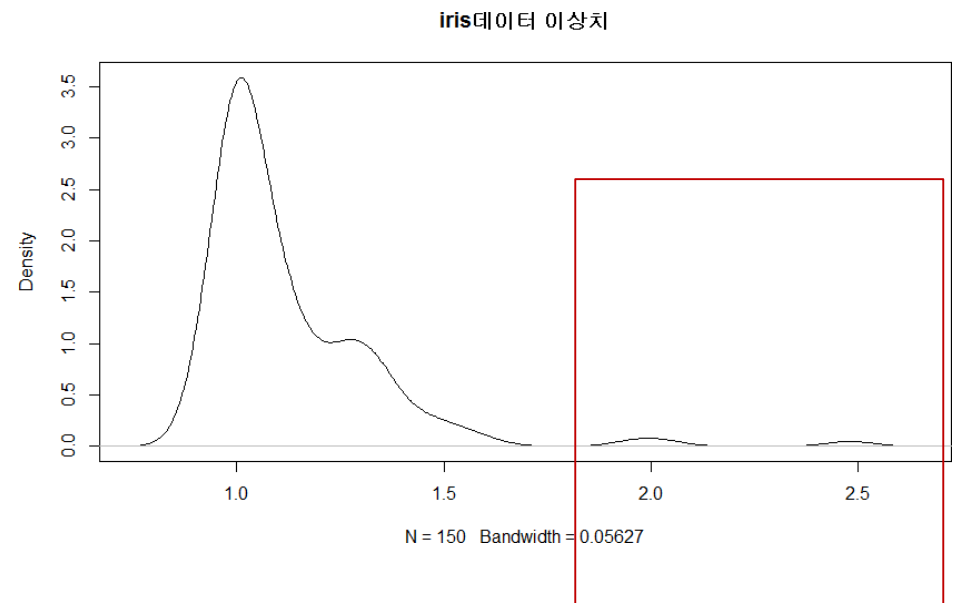
# Anomaly Detection – DMwR package

모든 변수를 고려한 outlier을 찾기 위해서는 DMwR 패키지의 lofactor(local outlier factor algorithm)함수를 이용한다.

## 코드

```
> install.packages("DMwR")
> library(DMwR)
> nospecies <- iris[,1:4]
> # k : outlier계산을 위한 이웃 갯수
> scores <- lofactor(nospecies,k=5)
# 밀도그래프를 그려보고 이상치 기준을 판단한다.
> plot(density(scores), main= ' iris데이터 이상치 ' )
# score값을 내림차순으로 정렬하여 보고
# 이상치 기준을 판단한다.
> sort(scores, decreasing=T)[1:10]
[1] 2.479960 2.029263 1.959143 1.602584 1.548281
[6] 1.528656 1.480122 1.463896 1.455104 1.451772
# socre 1.9를 기준이상인 경우 outlier처리
> outlier<-order(scores,decreasing=T)[1:3]
> outlier
[1] 42 107 23
```

## 실행결과



이상치 처리

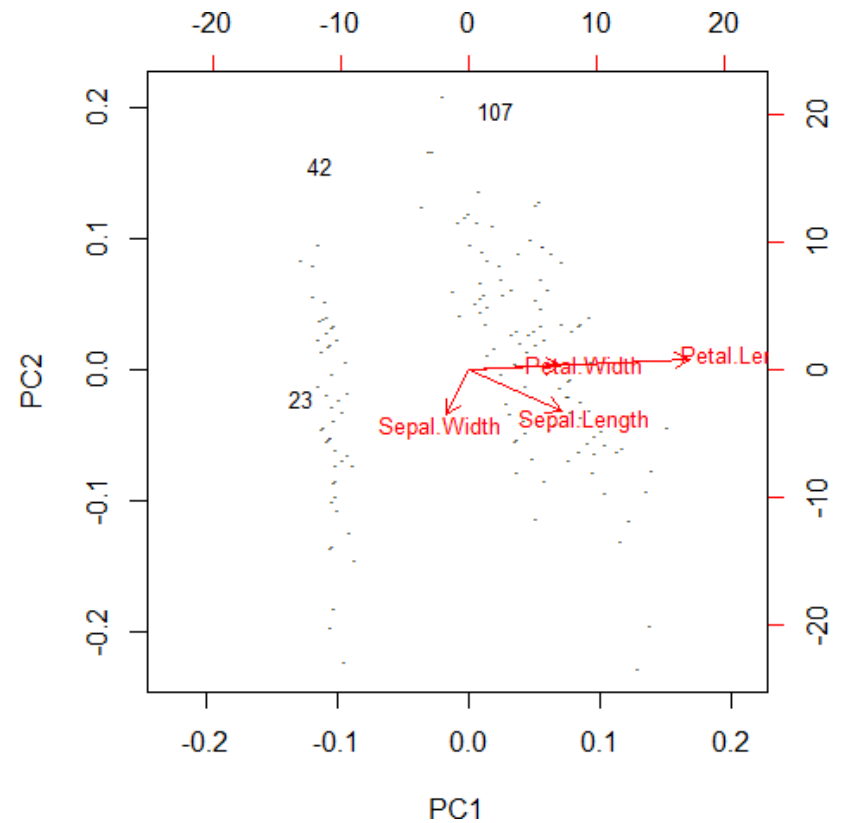
# Anomaly Detection – PCA

PCA를 이용하여 outlier를 분석해볼 수 있다.

## 코드

```
# 앞 페이지의 코드를 먼저 실행할 것  
> labels<-1:nrow(nospecies)  
> labels[-outlier]<-"."  
> biplot(prcomp(nospecies),cex=0.8, xlab=labels)
```

## 실행결과





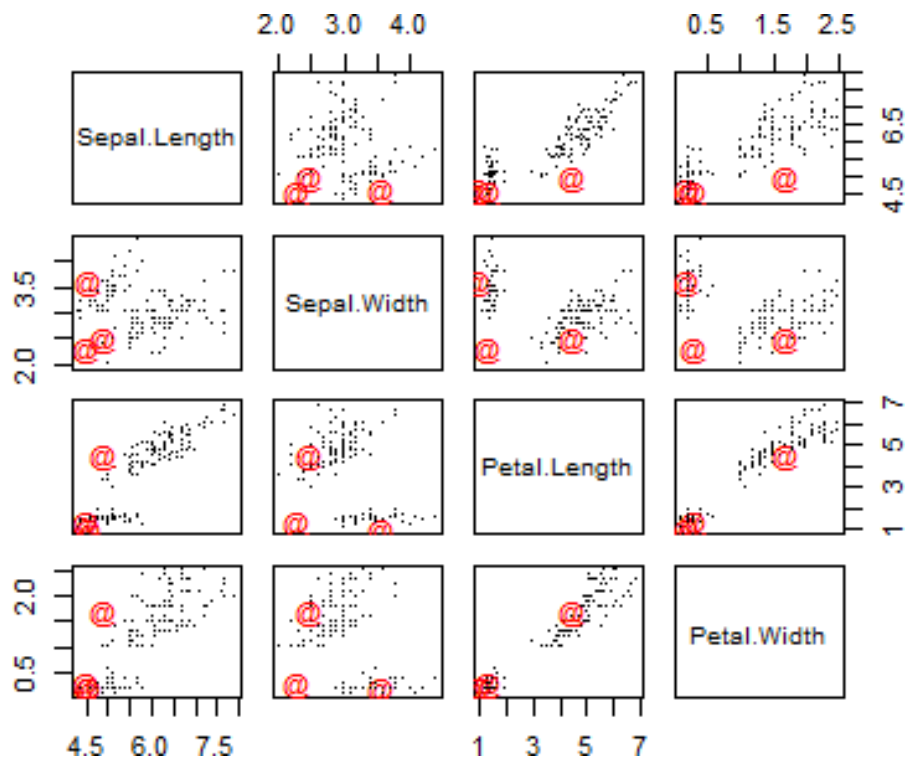
# Anomaly Detection – pairs

산점도를 이용하여 outlier를 분석해볼 수 있다.

## 코드

```
# 앞 페이지의 코드를 먼저 실행할 것  
> pch<-"."  
> pch[outlier]<-"@"  
> col<-rep("black",nrow(nospecies))  
> col[outlier]<-"red")  
> pairs(nospecies,pch=pch,col=col)
```

## 실행결과



# Association rules – Associations

연관성 두 데이터셋간의 연관성을 의미한다. 시장바구니 분석에서 가장 일반적으로 사용되며, 연관도를 표현하는 지표에는 하기와 같은 값이 있다.

## 연관성 지표

### □ 지지도(support)

전체 거래중에서, 품목A와 B가 모두 포함된 거래비율  
 $P(A \cap B)$

### □ 신뢰도(Confidence)

품목 A가 포함된 거래 중에서, 품목 B도 포함된 거래비율  
 $P(B|A) = P(A \cap B) / P(A)$

### □ 향상도(Lift)

연관규칙이 오른쪽 항목을 예측하는 능력이 얼마나 향상되었는지를 정량적으로 표현한 값

$$P(B|A) / P(B) = P(A \cap B) / (P(A) P(B))$$

값이 1이라면, A와 B가 독립임을 의미하며

값이 1보다 크면, A와 B는 양의 상관관계이고

값이 1보다 작으면, A와 B는 음의 상관관계이다.



품목 B가 발생할 확률대비

품목 A가 포함된 거래 중에서 품목 B가 발생할 확률의 비율

## Parameter

apriori(data, parameter = NULL, appearance = NULL, control = NULL)

Data : 트랜잭션 데이터

Parameter : 마이닝을 수행하는 기본값

지지도 : 0.1, 신뢰도 0.8, 최대길이 10

Appearance : 계산한 룰을 표시하는데 사용함

Control : 사용된 알고리즘의 성능을 조정하는데 사용함

# Association rules - Apriori

연관규칙을 알아내는데 사용하는 알고리즘으로 parameter는 하기와 같다.

## Parameter

---

apriori(data, parameter = NULL, appearance = NULL, control = NULL)

Data : 데이터셋

Parameter : 프로세스를 제어하기 위한 파라미터 리스트  
(기본설정값 지지도 : 0.1, 신뢰도 0.8, 최대길이 10)

Appearance : 사용대상 데이터 선정

Control : 알고리즘의 성능 제어(특히 정렬 기능)

# Association rules – Associations

연관성 분석은 하기와 같이 arules패키지의 apriori 명령을 이용하여 분석을 수행한다.

## 코드

```
> install.packages("arules")
> library(arules)
> data <- read.csv("c:/temp/groceries.csv")
> rules<-apriori(data)
> rules
> inspect(rules)
# apriori의 parameter를 수정하고자할 경우
> rules <- apriori(data, parameter = list(supp = 0.001, conf = 0.8))
#조회하고 싶은 Rule의 개수를 설정후 조회함
> Inspect(rules[1:5])
```

## 실행결과

```
Apriori
Parameter specification:
confidence minval smax arem aval originalSupport maxtime support minlen maxlen target
0.8 0.1 1 none FALSE ext TRUE 5 0.1 1 10 rules
FALSE
```

```
Algorithmic control:
filter tree heap memopt load sort verbose
0.1 TRUE TRUE FALSE TRUE 2 TRUE
```

```
Absolute minimum support count: 1529
set item appearances ... [0 item(s)] done [0.00s].
set transactions ... [655 item(s), 15295 transaction(s)] done [0.01s]. sort
ing and recoding items ... [3 item(s)] done [0.00s].
creating transaction tree ... done [0.00s]. chec
king subsets of size 1 2 3 done [0.00s]. writin
g ... [5 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
```

	lhs	rhs	support	confidence	lift
1	{semi.finished.bread=}	=> {margarine=}	0.2278522	1	2.501226
2	{semi.finished.bread=}	=> {ready.soups=}	0.2278522	1	1.861385
3	{margarine=}	=> {ready.soups=}	0.3998039	1	1.861385
4	{semi.finished.bread=margarine=}	=> {ready.soups=}	0.2278522	1	1.861385
5	{semi.finished.bread=,ready.soups=}	=> {margarine=}	0.2278522	1	2.501226

```
count
[1]3485
[2]3485
[3]6115
[4]3485
[5]3485
```

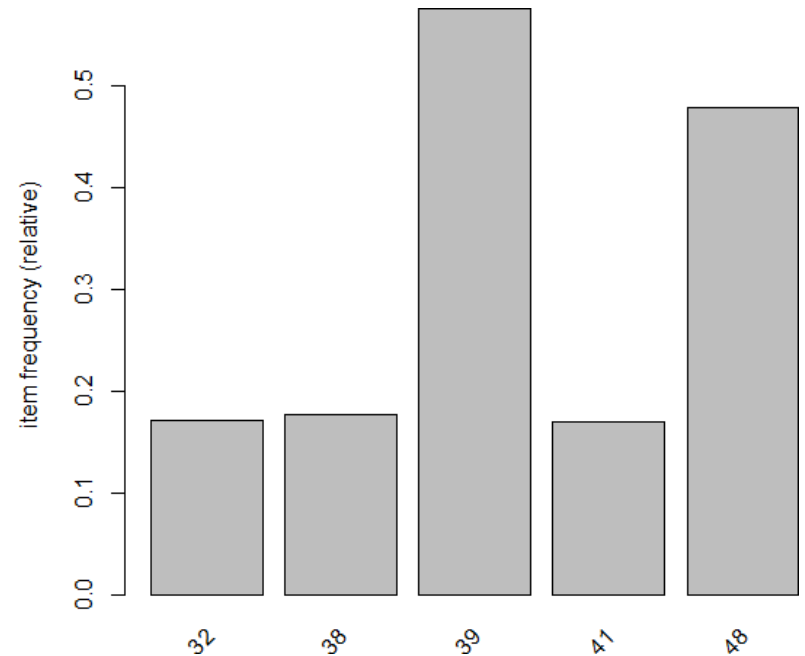
# Association rules - Apriori

데이터셋내의 연관규칙을 알아내기 위한 목적으로 apriori 분석을 한다.

## 코드

```
> install.packages('arules')
> library(arules)
> #데이터 읽어오기
> tr<-read.transactions("http://fimi.ua.ac.be/data/retail.dat", format='basket')
> #읽어온 데이터 요약정보 보기
> summaryr(tr)
> #Item빈도 그래프 그려보기
> itemFrequencyPlot(tr, support=0.1)
> #연관규칙 찾아내기(apriori사용)
> rules<-apriori(tr,parameter=list(supp=0.5, conf=0.5))
> #룰정보요약해서 보기
> summary(rules)
> #룰목록을 보기
> inspect(rules)
> #연관규칙에 대한 지지도, 신뢰도, 향상도등의 정보 보기
> interestMeasure(rules, c('support','chiSquare','confidence','conviction',
  'cosine','leverage','lift','oddsRatio'),tr)y
```

## 실행결과



lhs	rhs	support	confidence	lift	count
[1] {}	=> {39}	0.5747941	0.5747941	1	50675

누구나 대부분 item 39를 장바구니에 갖고 있음을 알 수 있다.

# Association rules - Eclat

Itemset 패턴(장바구니 분석)을 감지하기 위한 목적으로 Eclat분석을 하며, parameter는 하기와 같다.

## Parameter

---

eclat(data, parameter = NULL, control = NULL)

data : 분석대상 데이터 행렬

parameter : ECPParameter이나 리스트 객체

control : Eccontrol 이나 리스트 객체

## ECP & Eccontrol parameter

---

### ECP Parameter

- support: itemset에 대한 최소 지지도, 기본설정값 0.1
- minlen : itemset에 대한 최소 사이즈, 기본설정값 1
- maxlen : itemset에 대한 최대 사이즈, 기본설정값 10
- target : 연관성 유형
  - . Frequent itemsets
  - . Maximally Frequent itemsets
  - . Closed Frequent itemsets

### Eccontrol Values

- sort : 1(오름차순), -1(내림차순), 0(정렬안함), 2(오름차순), - 2(내림차순)
- verbose : 진행정보 표시

# Association rules - Eclat

Dataset에서 발생빈도가 많은 items를 찾으려면 **arules**패키지의 **eclat**함수를 이용하여 분석을 수행한다.

## 코드

```
> data("Adult") http://archive.ics.uci.edu/ml/datasets/Adult  
(인구조사 데이터)  
> dim(Adult) [1]  
48842 115  
> summary(Adult)  
transactions as itemMatrix in sparse format with 4  
8842 rows (elements/itemsets/transactions) and  
115 columns (items) and a density of 0.1089939  
  
most frequent items: ca  
pital-loss=None 46560 capital-gain=None 44807  
native-country=United-States 43 race=White 41762  
832 (Other)  
workclass=Private 401333  
33906  
.....  
> itemsets<-eclat(Adult)  
> # set of 2616 itemsets  
> itemsets.sorted<-sort(itemsets)  
> itemsets.sorted[1:5]  
> inspect(itemsets.sorted[1:5])  
> inspect(itemsets.sorted[100:100])
```

## 실행결과

전체 2,616개의 Itemsets에서

지지도(support) 기준으로 정렬하여 상위 5개를 분석하면 다음과 같다.

	items	support	count
1	{capital-loss=None}	0.9532779	46560
2	{capital-gain=None}	0.9173867	44807
3	{native-country=United-States}	0.8974243	43832
[4]	{capital-gain=None,capital-loss=None}	0.8706646	42525
[5]	{race=White}	0.8550428	41762
...			
[100]	{workclass=Private, capital-gain=None, hours-per-week=Full-time}	0.3952131	19303



- 조사대상자는 대부분은 capital-loss(양도손실)이 나 capital-gain(양도소득)을 얻지 못했다.
- 조사대상자 대부분은 국적이 미국이다.
- 조사대상자 대부분은 백인이다.

# Association rules - Eclat

Dataset에서 발생빈도가 많은 items를 찾기 위해서 Eclat함수를 이용하여 분석을 수행한다.

## 코드

```
> data("Adult")
> itemsets <- eclat(Adult, parameter=list(minlen=9))
> inspect(itemsets)
```

## 실행결과

items	Support	count
[1] {age=Middle-aged, workclass=Private, marital-status=Married-civ-spouse, relationship=Husband, race=White, sex=Male, capital-gain=None, capital-loss=None, native-country=United-States}	0.1056673	5161

**인구조사결과 데이터에서 Itemset의 길이가 9이상인 경우**

빈도가 가장 많은 or 유의미한 연관규칙은

1) 중년, 기혼 & 남편, 백인, 남자, capital-gain/loss 없음, 국적 미국인  
경우이다.

- 지지도는 0.106이며

- 빈도수는 5161이다.



순차 패턴(sequential pattern)분석은 데이터에 공통으로 나타나는 순차적인 패턴을 찾는 방법이다.

## □ 고객('정우성')의 마트 구매 품목 정보

- 월요일 : 담배, 술 구매
- 화요일 : 담배, 신문 구매
- 수요일 : 음료수, 과자 구매

## □ 트랜잭션이란?

- 동시에 이루어지는 Event에 대한 정보를 데이터 관리/분석 측면에서는 트랜잭션(Transaction)이라 한다.
- 예) {담배, 술} 트랜잭션, {담배, 신문} 트랜잭션, {음료수, 과자} 트랜잭션

## □ 시퀀스란?

- 개별 트랜잭션의 시간적 순서를 고려한 정보를 시퀀스(Sequence)라고 한다.
- 예) ( {담배, 술}, {담배, 신문}, {음료수, 과자} ) 시퀀스  
→ 이와 같은 이러한 시퀀스를 사용자 시퀀스라고 한다.

## □ 순차 패턴 마이닝이란?

- 모든 사용자 시퀀스 중 몇 % 이상 공통으로 나타내는 시퀀스를 찾는 것이 순차 패턴 마이닝이다.

# Sequence Patterns - Determining sequences

TraMineR 패키지의 seqdef는 시퀀스 형태의 데이터를 분석하고 가시화하는데 사용한다.

## Parameter - seqdef

```
seqdef(data, var=NULL, informat="STS", stsep=NULL, a  
lphabet=NULL, states=NULL, id=NULL, weights=NULL,  
start=1, left=NA, right="DEL", gaps=NA,  
missing=NA, void="%", nr="*", cnames=NULL,  
xstep=1, cpal=NULL, missing.color="darkgrey",  
labels=NULL, ...)
```

- data : 데이터(행렬)
- var : 시퀀스를 포함하는 열 리스트, NULL은 모든 열을 의미
- informat : 원시데이터의 형태(STS, SPS, SPELL)
- stsep : 구분자
- alphabet : 모든 가능 상태 리스트
- states : 짧은 상태 레이블(labels)

## Parameter - seqXXXplot

가시화기능(seqXXXplot)은 시퀀스 분포, 빈도, 이상치 등에 대한 그래프를 만들어 준다.

```
seqXXXplot(seqdata, group=NULL, type="i", title=NULL,  
cpal=NULL, missing.color=NULL,  
ylab=NULL, yaxis=TRUE, axes="all", xtlab=NULL, cex.plot=1,  
withlegend="auto", ltext=NULL, cex.legend=1, use.layout=(l  
s.null(group) | withlegend!=FALSE), legend.prop=NA, rows=  
NA, cols=NA, ...)
```

### - XXX 대신 사용해야 하는 parameter

- i : 인덱스 plot을 표시함
- f : 빈도 plot을 표시함
- d : 분포 plot을 표시함

## Sequence Patterns - Determining sequences

TraMineR 패키지를 이용하여 개별 데이터의 Seq.와 Seq. 빈도를 표시한 방법은 하기와 같다.

## 코드

```
>install.packages('TraMineR')
>library('TraMineR')
>#93/7~99/ 6까지 학생진로추적data
>data(mvad)
>summary(mvad)
>#17번째 열~86번째열까지 데이터 대상 분석
>myseq<-seqdef(mvad, 17:86)
># 인덱스 indexplot 그리기
># 가장빈도가많은순서로
># 디폴트는 1:10이며
># 옵션 idx=1:20과 같이 수정가능함
>seqiplot(myseq)
># 빈도 Frequencyplot그리기
>seqfplot(myseq)
```

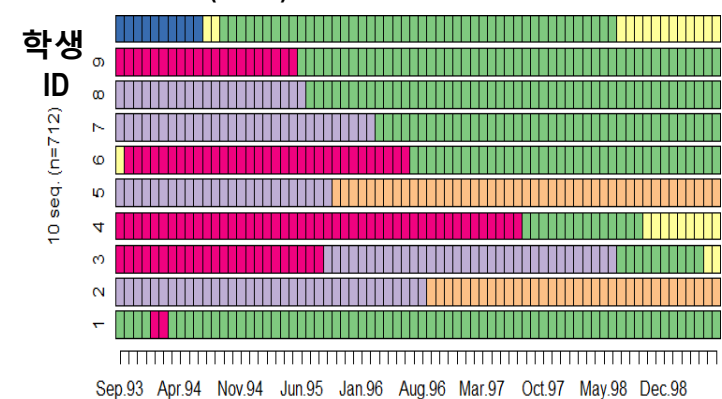
## 실행결과

```
> myseq<-seqdef(mvad, 17:86)
[>] 6 distinct states appear in the data:
1 = employment
2 = FE
3 = HE
4 = joblessness
5 = school
6 = training
[>] state coding:
      [alphabet] [label]      [long label]
1 employment    employment    employment
2 FE            FE            FE
3 HE            HE            HE
4 joblessness    joblessness    joblessness
5 school         school        school
6 training       training      training
[>] 712 sequences in the data set
[>] min/max sequence length: 70/70
```

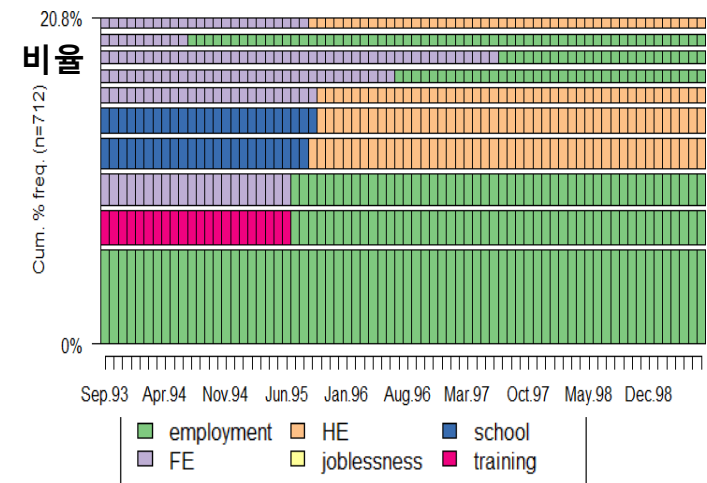
<학교 졸업후 취업상태 여부 데이터 Label>

- Employment: 취직
- FE : Further Education(졸업후 추가학업, Only in British)
- HE : Higher Education(대학원 진학)
- Joblessness : 무직
- School : 학교
- Training : 직업훈련

### <학생 ID(1~10)별 조사기간의 state 표시결과>



**<빈도 순서별 state sequence data 표시결과>**

[illegible]

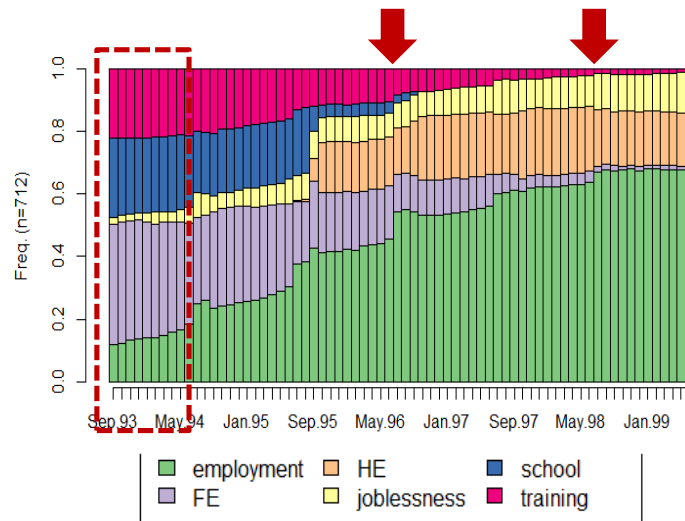
# Sequence Patterns - Determining sequences

TraMineR 패키지를 이용하여 데이터를 분석한 방법은 하기와 같다.

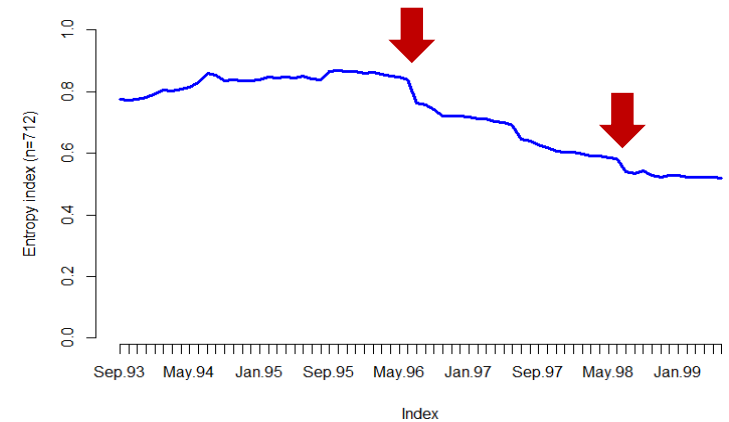
## 코드

```
># 분포 Distribution plot 그리기  
>seqdplot(myseq)  
># 엔트로피 Entropychart 그리기  
>seqHtplot(myseq)
```

<각 기간별 state의 분포도 가시화 결과>



## 실행결과



초기('93년 3월 시점)에는 Entropy값이 증가하지만 시간이 경과하여 말기('99년 6월 시점)에 가까워지면 Entropy값이 감소함

## □ 정보공학에서 엔트로피 값의 의미

- 엔트로피값이 작다는 것은 분류결과 데이터의 동질성이 높음을 의미함  
(극소수의 데이터 유형이 존재함을 의미)
- 엔트로피값이 높다는 것은 분류결과 데이터의 동질성이 낮음을 의미함  
(많은 종류의 데이터 유형이 존재함을 의미)

초기에 학생의 상태는  
여러 가지 (school, training, joblessness 등)이나  
시간이 지나면 대부분 employment 상태가 됨

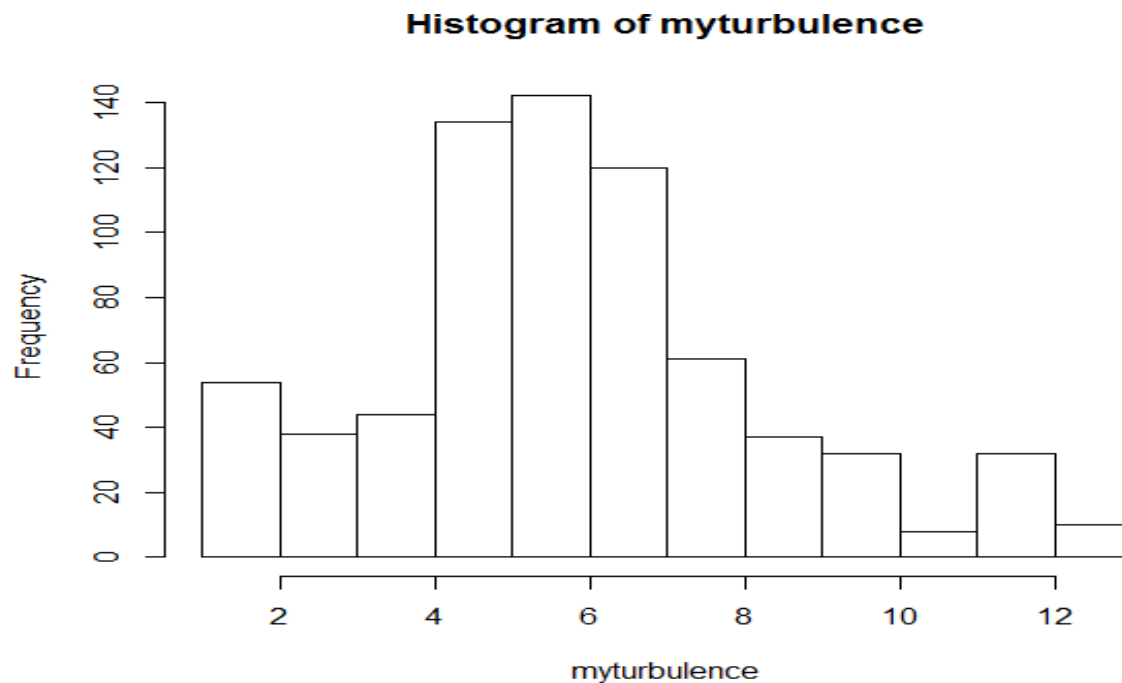
# Sequence Patterns - Determining sequences

TraMineR 패키지를 이용하여 데이터를 분석한 방법은 하기와 같다.

## 코드

```
> #개 별 state sequence data에 대해  
> #몇 개의 sub sequence data로 구분가능한지를  
> #계산한 지표임  
> myturbulence<-seqST(myseq)  
[>] extracting symbols and durations ...  
[>] computing turbulence for 712 sequence(s) ...  
> hist(myturbulence)
```

## 실행결과



대부분의 state sequence data는  
소수의 state sequence 상태로 존재하며  
몇몇의 경우에만 여러 가지의 state sequence 상태로 존재함

# Sequence Patterns - Similarities in the sequence

TraMineR 패키지의 seqdist는 시퀀스 형태의 데이터에 대한 유사도를 계산하는데 사용한다.

## Parameter - seqdist

```
seqdist(seqdata, method, refseq=NULL, norm=FALSE,  
indel=1, sm=NA, with.missing=FALSE, full.matrix=TRUE)
```

- seqdata : 상태 시퀀스(seqdef를 이용하여 계산함)
- method : LCP메소드
- refseq : 참조 시퀀스(선택parameter)
- norm : 거리를 정규화함
- indel : OM에만 사용됨
- sm : 대체 매트릭스(LCP의 경우 무시함)
- with.missing : 결측 gap이 있는 경우 TRUE
- full.matrix : TRUE이면 full 매트릭스가 반환됨

## 용어

### ❑ LCP : Longest Common Prefix

유사도 계산하기 위해 동일한 최대 길이의 시퀀스 프리픽스를 비교할 수 있다.

### ❑ LCS : Longest Common subsequence

유사도 계산을 위해 두 시퀀스 사이에서 동일한 가장 긴 부분을 찾아비교할 수 있다.

### ❑ OM : Optimal Matching Distance

하나의 시퀀스를 다른 시퀀스에서 생성하기 위해 삽입/삭제 비용면에서 최적의 편집거리이다.

# Sequence Patterns - Similarities in the sequence

TraMineR 패키지의 seqdist는 sequence에 대한 유사도를 계산해주며 다음과 같이 사용할 수 있다.

## 코드

```
> data(famform)
> seq<-seqdef(famform)
[>] found missing values ('NA') in sequence data
[>] preparing 5 sequences
[>] coding void elements with '%' and missing values with '*'
[>] 5 distinct states appear in the data:
```

```
1 = M
2 = MC
3 = S
4 = SC
5 = U
```

```
[>] state coding:
```

	[alphabet]	[label]	[long label]
1	M	M	M
2	MC	MC	MC
3	S	S	S
4	SC	SC	SC
5	U	U	U

```
[>] 5 sequences in the data set
```

```
[>] min/max sequence length: 2/5
```

```
> seq
```

```
Sequence
```

```
1 S-U
2 S-U-M
3 S-U-M-MC
4 S-U-M-MC-SC
5 U-M-MC
```

## 실행결과

```
> #시퀀스 3과 4에 대해 동일한 최대길이
```

```
> seqLLCP(seq[3,],seq[4,])
```

```
[1] 4
```

```
> #시퀀스 1과 2를 비교한 결과 일치하는 sub-seq중 가장 긴 길이가 2임
```

```
> seqLLCS(seq[1,],seq[2,])
```

```
[1] 2
```

```
> cost<-seqsubm(seq, method='CONSTANT', cval=2)
```

```
[>] creating 5x5 substitution-cost matrix using 2 as constant value
```

```
> cost
```

	M->	MC->	S->	SC->	U->
M->	0	2	2	2	2
MC->	2	0	2	2	2
S->	2	2	0	2	2
SC->	2	2	2	0	2
U->	2	2	2	2	0

```
> LCS.ex <- c("S-U-S-M-S-U", "U-S-SC-MC", "S-U-M-S-SC-UC-MC")
```

```
> LCS.ex <- seqdef(LCS.ex)
```

```
> seqLLCP(LCS.ex[1,],LCS.ex[3,])
```

```
[1] 2
```

```
> seqLLCS(LCS.ex[1,],LCS.ex[3,])
```

```
[1] 4
```

의사결정규칙(Decision Rule)을 나무구조로 도표화하여 분류(Classification)와 예측(Prediction)을 수행하는 분석방법이다.

## □ 방법

- 의사결정나무 생성방법 : 분리기준에 근거하여 현 단계에 존재하는 Data를 분리
- 분리기준 : 순수도(Uniformity)

재래시장



대형Mart

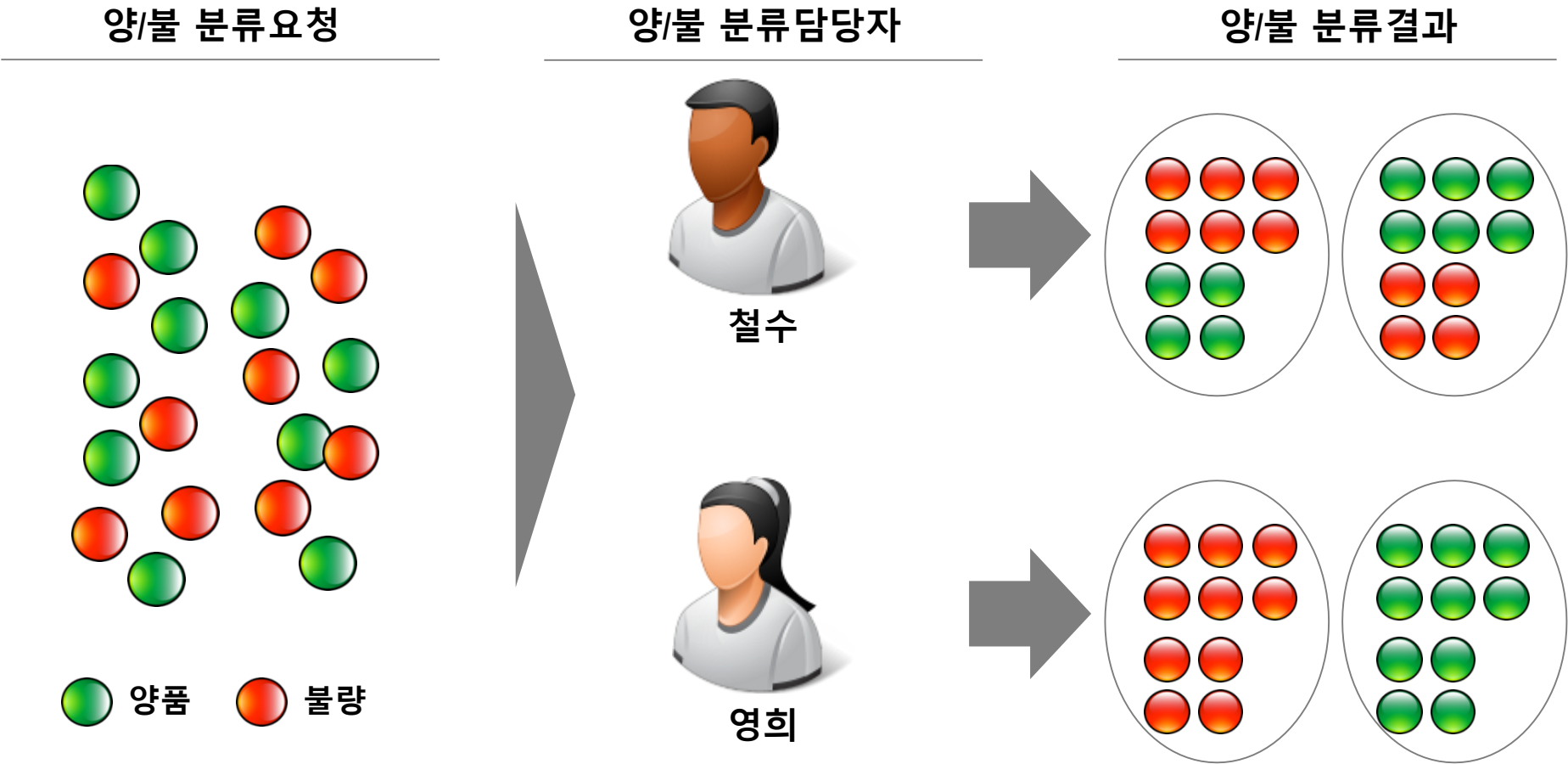


고객이 구매하려는 물건을 손쉽게 편리하게 구매할 수 있는 곳은 어디일까?  
왜 그렇게 생각하는지?



# Classification – DT

철수와 영희가 아래와 같이 공을 분류(Classification)했다고 하자. 누가 더 잘했다고 할 수 있는가?  
이에 대한 판단기준은 무엇일까? (Hint : 분류후의 자료의 동질성 → 순수도)

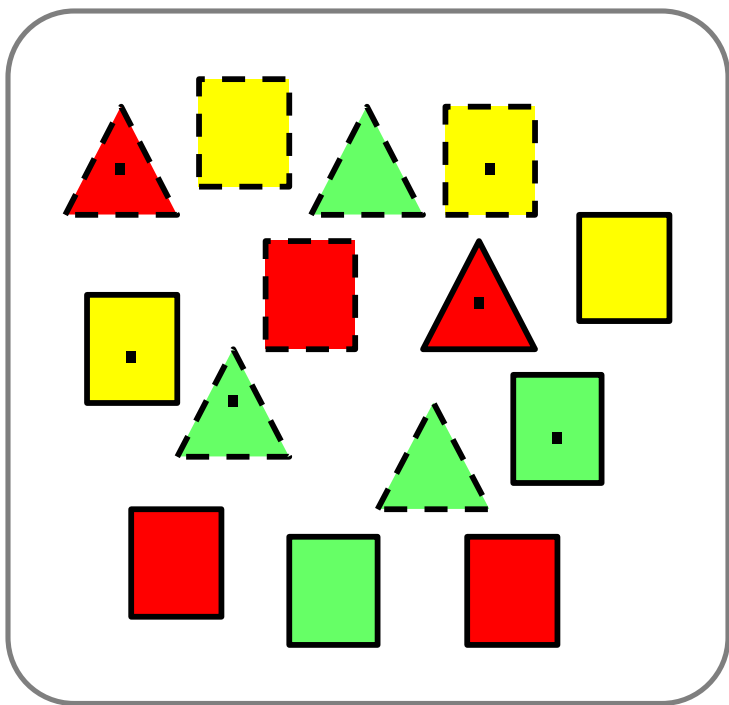


분류정확도 판단기준에 의하여 알고리즘을 크게 3개로 구분할 수 있다.

CART	<p>CART 알고리즘은 <b>지니지수(Gini Index)</b> 또는 <b>분산의 감소량</b>을 사용하여 나무의 가지를 <b>이진(Binary) 분리</b>한다. (범주형 변수에 대해서는 지니지수를 사용하고, 연속형 변수에 대해서는 분산의 감소량을 사용한다.)</p> $Gini = 1 - \sum_{i=1}^k \left( \frac{\# \text{ of } O_i}{n} \right)^2$ <p>n : 전체자료의 수(해당노드의 크기) # of O(i) : 부류i에 속한 자료의 수 k : 부류(서로다른 개체의 가지수)</p>
CHAID	<p>“CH”는 카이제곱(chi-squared)을 의미하며 CHAID의 분할기준은 카이제곱 검정에 근거하고 있다. 분할의 효과를 판단하기 위한 CHAID의 검정법은 열의 분포(목적변수값인 부류의 비율)가 각 행 (자식 마디)에서 서로 같은가를 검정함으로써 분할의 가치를 판단하는 방법이다. 최적분할 기준을 선택하기 위해서는 카이제곱 검증을 통하여 p값을 구하고, 구한 값 중에서 가장 낮은 p값을 갖는 변수를 분할기준으로 선택한다.</p> <p>X<sup>2</sup>을 계산하여 Chi-square검정결과 p-value에 의한 불순도를 평가한다.</p> $\chi^2 = \sum \frac{(O - E)^2}{E}$
C5.0	<p>C4.5 또는 C5.0은 엔트로피(Entropy)기반 이득율(gain ratio)에 의한 동질성 측도를 이용하여 분류가 이루어진다. 특정 노드의 엔트로피는 다음 식에 의해 구한다.</p> $I(t) = - \sum p(j   t) \times \log_2 \{ p(j   t) \}$ <p>I(t) : 해당 노드의 엔트로피(불순도)</p>

지니지수(Gini Index)에 근거한 초기 불순도 계산결과는 하기와 같다.

## □ 초기 불순도 계산



- 5 Triangles
- 9 Squares
- Gini지수

$$p(\square) = \frac{9}{14}$$

$$p(\triangle) = \frac{5}{14}$$

$$Gini = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

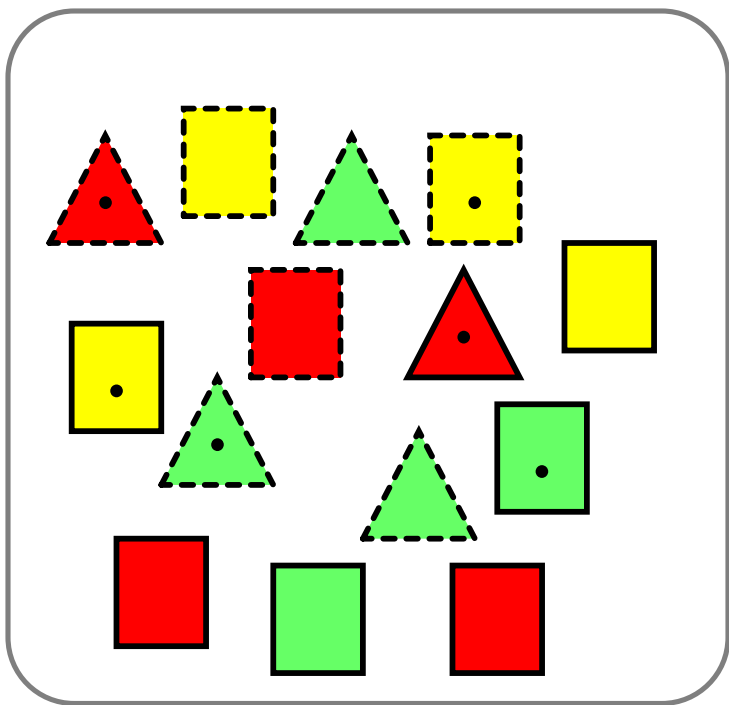
## □ 최적 분류기준 선정방법

- 가능한 신규 분류 기준을 적용한 후 Gini Index를 계산한 결과

이전 단계의 Gini Index대비 개선 효과가 가장 큰 분류기준을 선택하여 분류함

엔트로피 지수(Entropy Index)에 근거한 초기 불순도 계산결과는 하기와 같다.

## □ 초기 불순도 계산



- 5 Triangles
- 9 Squares
- Entropy지수

$$p(\square) = \frac{9}{14}$$

$$p(\triangle) = \frac{5}{14}$$

$$Entropy = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940 bits$$

## □ 최적 분류기준 선정방법

- 가능한 신규 분류 기준을 적용한 후 Entropy Index를 계산한 결과

이전 단계의 Entropy Index대비 개선 효과가 가장 큰 분류기준을 선택하여 분류함

R언어에서 Decision Tree 분석을 위한 방법은 tree, rpart, party 패키지를 이용한다.

## 패키지별 특성

### ❑ tree

- Binary recursive partitioning을 수행함
- 과적합의 위험이 있어 Pruning과정이 필요함

### ❑ rpart

- CART(Classification and Regression Trees) 방법론을 사용함
- 과적합의 위험이 있어 Pruning 과정이 필요함

### ❑ party

- Unbiased recursive partitioning based on permutation tests 방법론을 사용함
- 과적합의 위험이 없으나, 입력변수의 레벨이 31개 까지로 제한되어 있음

## DT분석의 장점

의사결정나무 분석은

회귀분석이나 랜덤포레스트 등의 알고리즘에 비해

- 1) 직관적인 이해
- 2) 설명이 용이하다

장점을 갖고있는 지도학습 분석 방법임

# Classification – DT(tree)

의사결정 나무 형태의 분류분석을 하려면 tree 패키지를 사용하며 사용법은 하기와 같다.

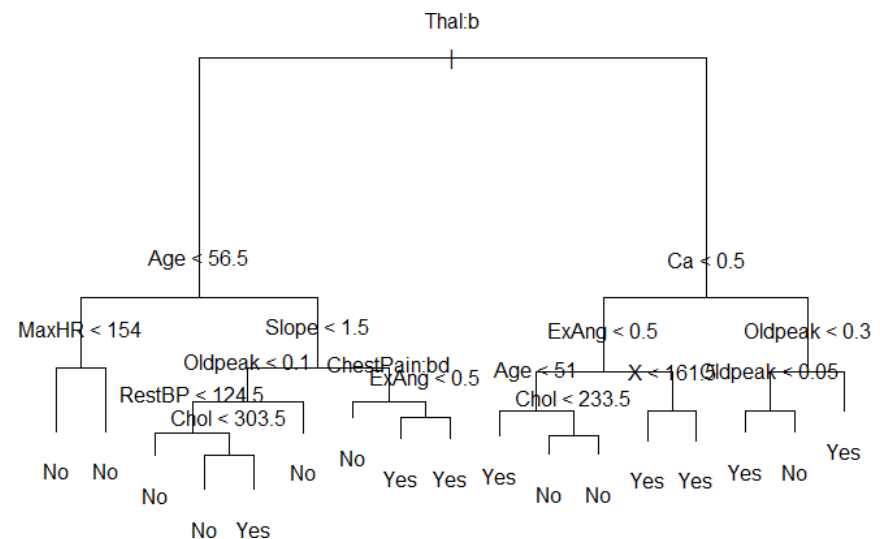
## 코드

```
>df<-read.csv('c:/temp/Heart.csv')
>str(df)
>head(df)

>library(caret)
>set.seed(1000) #reproducibility setting
>intrain<-createDataPartition(y=df$AHD, p=0.7, list=FALSE)
>train<-df[intrain, ]
>test<-df[-intrain, ]

>library(tree)
>treemod<-tree(AHD~. , data=train)
>plot(treemod)
>text(treemod)
```

## 실행결과



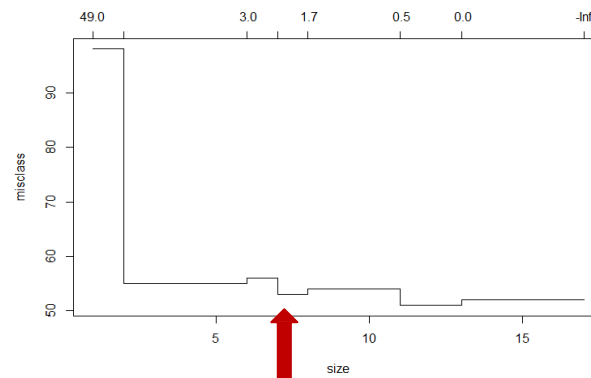
# Classification – DT(tree)

과적합의 문제를 해결하기 위해서 가지치기(pruning단계)가 필요하며, 이를 위해 최적 가지수를 결정하려면 k-fold crossvalidation을 수행하여 분산이 가장 낮은 가지수를 구한다.

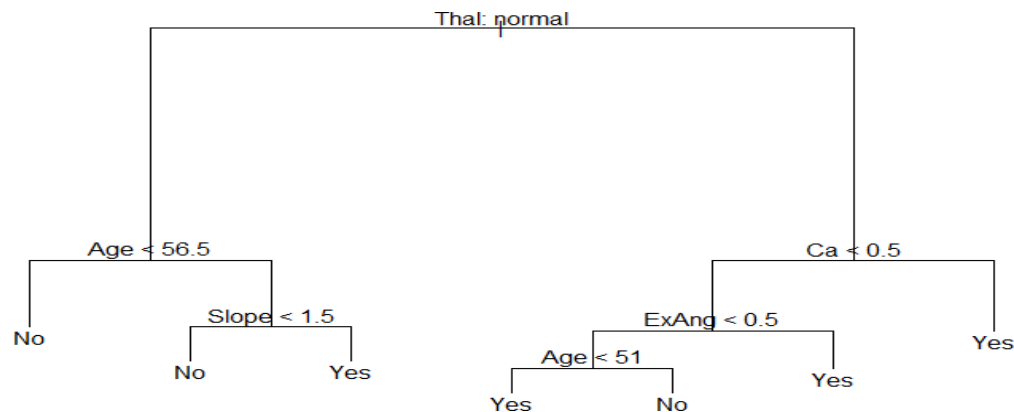
## 코드

```
> #Tree패키지의 cv.tree를 사용하여 k-fold crossvalidation을 수행한다.  
> cv.trees<-cv.tree(treemod, FUN=prune.misclass )  
> plot(cv.trees)  
  
> #가지치기를 위해서 prune.trees를 사용한다.  
> prune.trees <- prune.misclass(treemod, best=7)  
> plot(prune.trees)  
> text(prune.trees, pretty=0)
```

## 실행결과



분석결과 size가 7인 경우  
낮은 값을 갖으므로  
이 값으로 가지수를  
설정한다.



# Classification – DT(tree)

의사결정 모델링의 정확도를 측정한 결과는 다음과 같다.

## 코드

```
> treepred <- predict(prune.trees, test, type='class')  
> confusionMatrix(treepred, test$AHD)
```

## 실행결과

### Confusion Matrix and Statistics

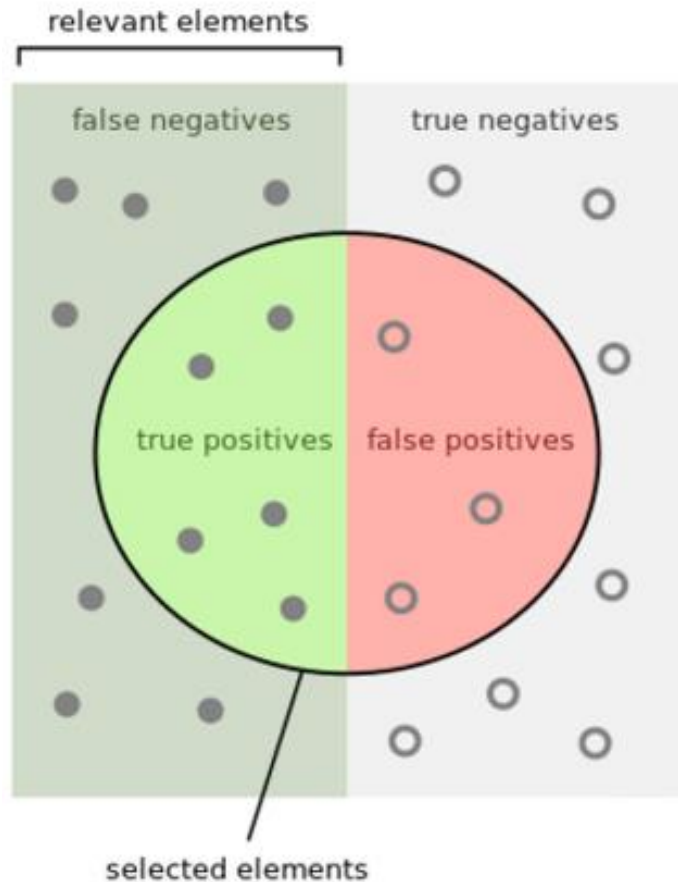
```
              Reference  
Prediction No Yes  
No      36    9  
Yes     13   32  
  
Accuracy : 0.7556  
95% CI : (0.6536, 0.84)  
No Information Rate : 0.5444  
P-value [Acc > NIR] : 2.879e-05  
  
Kappa : 0.5111  
McNemar's Test P-Value : 0.5224  
  
Sensitivity : 0.7347  
Specificity : 0.7805  
Pos Pred Value : 0.8000  
Neg Pred Value : 0.7111  
Prevalence : 0.5444  
Detection Rate : 0.4000  
Detection Prevalence : 0.5000  
Balanced Accuracy : 0.7576  
  
'Positive' Class : No
```

분석결과 tree패키지를 이용한 의사결정 모델링의 정확도는 0.76이 된다.



# Classification – DT(tree)

일반적으로 분류(classification)모델의 성능은 하기와 같은 항목으로 측정한다.

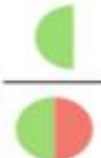


Accuracy



$$= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$$

Precision



$$= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$$

Recall



$$= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$$

False Positive Rate =  
(Fall-out)



$$= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$$

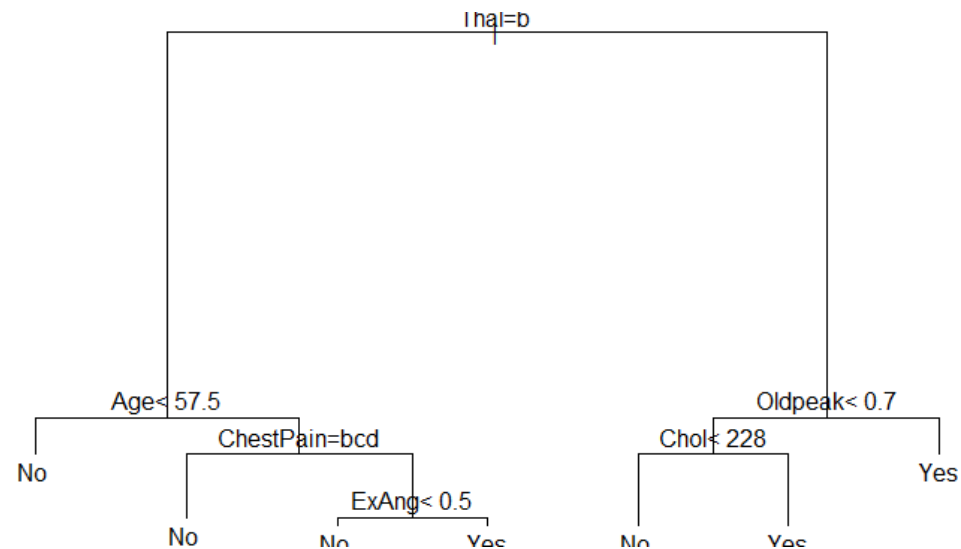
# Classification – DT(rpart)

의사결정 나무 형태의 분류분석을 하려면 rpart 패키지를 사용하며 사용법은 하기와 같다.

## 코드

```
> library(rpart)
> rpartmod<-rpart(AHD~., data=train, method='class')
> plot(rpartmod)
> text(rpartmod)
```

## 실행결과



# Classification – DT(rpart)

과적합의 문제를 해결하기 위해서 가지치기(pruning단계)가 필요하며, 이를 위해 최적 가지수를 결정하려면 k-fold crossvalidation을 수행하여 분산이 가장 낮은 가지수를 구한다.

## 코드 및 실행결과

```
> printcp(rpartmod)
```

```
Classification tree:
rpart(formula = AHD ~ ., data = train, method = "class")

Variables actually used in tree construction:
[1] Age      ChestPain Chol      ExAng      Oldpeak  Thal

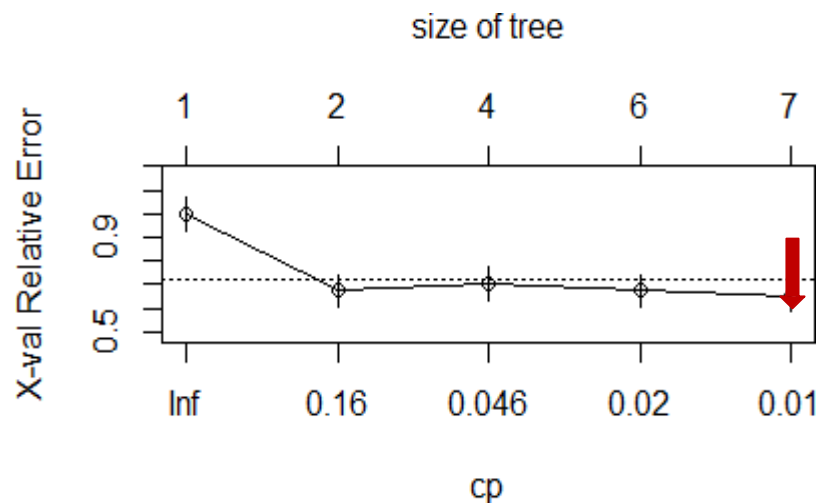
Root node error: 98/213 = 0.46009

n= 213
```

	CP	nsplit	rel error	xerror	xstd
1	0.489796	0	1.00000	1.00000	0.074224
2	0.051020	1	0.51020	0.67347	0.068868
3	0.040816	3	0.40816	0.70408	0.069693
4	0.010204	5	0.32653	0.67347	0.068868
5	0.010000	6	0.31633	0.65306	0.068276

## 코드 및 실행결과

```
> plotcp(rpartmod)
```



분석결과 size가 7인 경우 가장 낮은 값을 갖으므로 이 값으로 가지수를 설정한다.

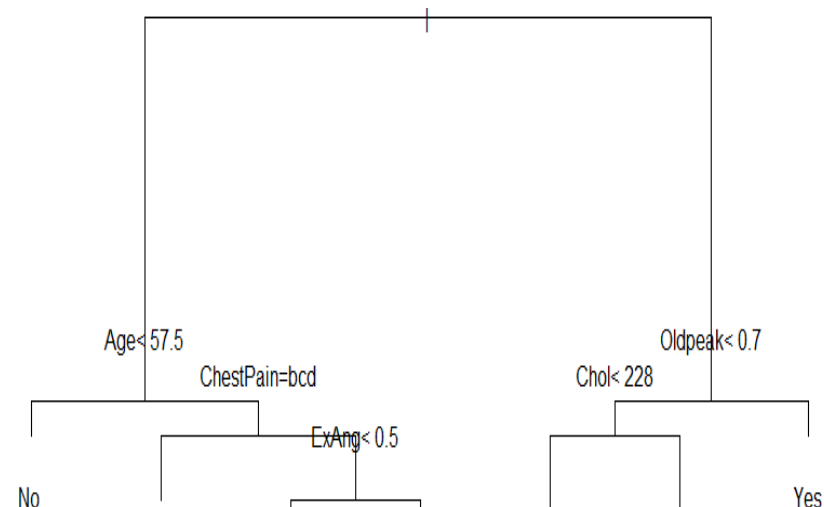
## Classification – DT(rpart)

과적합의 문제를 해결하기 위해서 가지치기(pruning단계)가 필요하며, 이를 위해 최적 가지수를 결정하려면 k-fold crossvalidation을 수행하여 분산이 가장 낮은 가지수를 구한다.

## 코드

```
>ptree<-prune(rpartmod, cp = rpartmod$cpstable[which.min(rpartmod$cpstable[, "xerror"]), "CP"])
>plot(ptree)
>text(ptree)
```

## 실행결과



# Classification – DT(rpart)

의사결정 모델링의 정확도를 측정한 결과는 다음과 같다.

## 코드

```
> rpartpred<-predict(ptree, test, type='class')  
> confusionMatrix(rpartpred, test$AHD)
```

## 실행결과

### Confusion Matrix and Statistics

```
              Reference  
Prediction No Yes  
No      39  12  
Yes     10  29  
  
Accuracy : 0.7556  
95% CI : (0.6536, 0.84)  
No Information Rate : 0.5444  
P-Value [Acc > NIR] : 2.879e-05  
  
Kappa : 0.5052  
McNemar's Test P-Value : 0.8312  
  
Sensitivity : 0.7959  
Specificity : 0.7073  
Pos Pred Value : 0.7647  
Neg Pred Value : 0.7436  
Prevalence : 0.5444  
Detection Rate : 0.4333  
Detection Prevalence : 0.5667  
Balanced Accuracy : 0.7516  
  
'Positive' Class : No
```

분석결과 rpart패키지를 이용한 의사결정 모델링의 정확도는 0.76이 된다.

# Classification – DT(party)

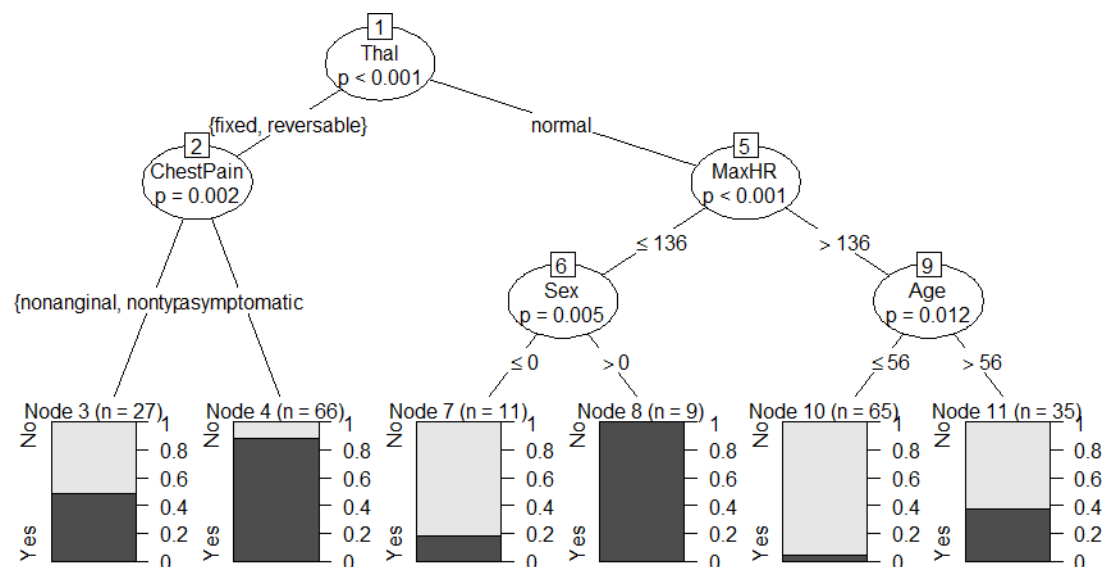
의사결정 나무 형태의 분류분석을 하려면 party 패키지를 사용하며 사용법은 하기와 같다.

Party패키지 방법의 경우 가지치기(pruning)를 significance를 이용하여 실행하므로 별도작업이 필요하지 않다.

## 코드

```
> library(party)
> partymod<-ctree(AHD~., data=train)
> plot(partymod)
```

## 실행결과



# Classification – DT(party)

의사결정 모델링의 정확도를 측정한 결과는 다음과 같다.

## 코드

```
> partypred<-predict(partymod, test)
> confusionMatrix(partypred, test$AHD)
```

## 실행결과

### Confusion Matrix and Statistics

	Reference	
Prediction	No	Yes
No	44	16
Yes	5	25

Accuracy : 0.7667  
95% CI : (0.6657, 0.8494)  
No Information Rate : 0.5444  
P-Value [Acc > NIR] : 1.061e-05

Kappa : 0.5191  
McNemar's Test P-Value : 0.0291

Sensitivity : 0.8980  
Specificity : 0.6098  
Pos Pred Value : 0.7333  
Neg Pred Value : 0.8333  
Prevalence : 0.5444  
Detection Rate : 0.4889  
Detection Prevalence : 0.6667  
Balanced Accuracy : 0.7539

'Positive' Class : No

분석결과 party패키지를 이용한 의사결정 모델링의 정확도는 0.77이 된다.

변수들간의 인과 관계를(종속변수, 설명변수) 설정하여 변수들간의 함수형태, 영향을 미치는 변수, 종속 변수에 대한 예측값을 얻는 분석방법이다.

## 회귀분석을 통하여 선형 함수 관계를 설정

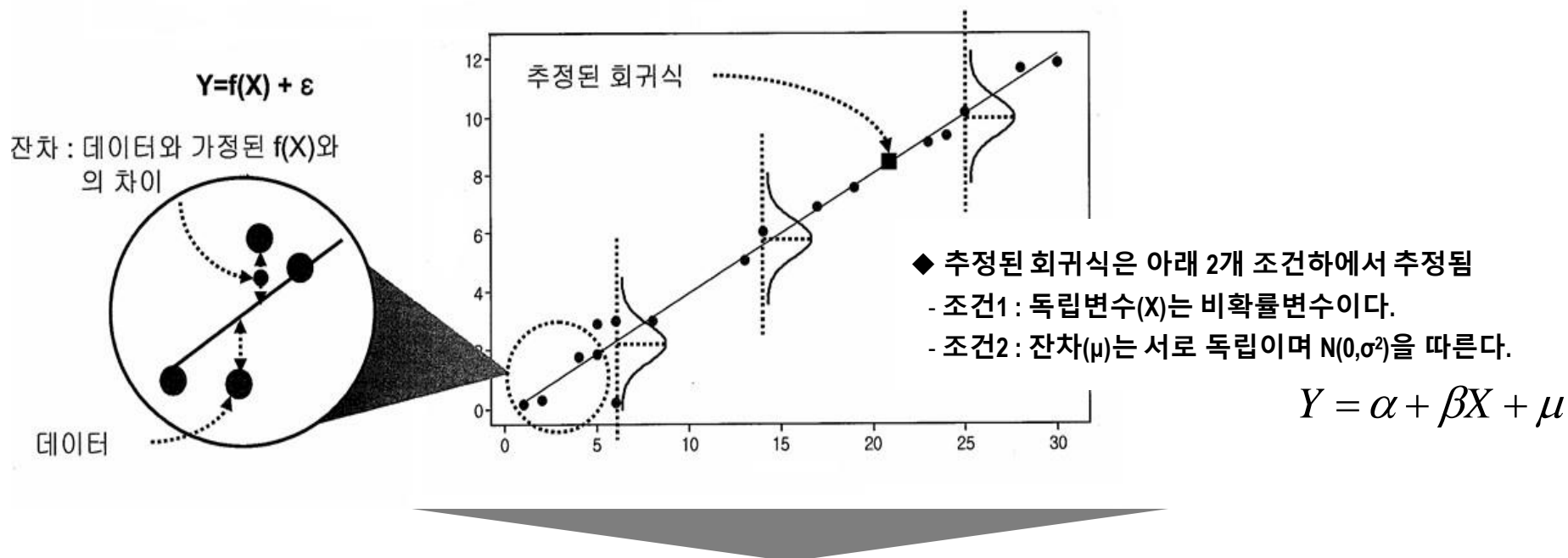
- ◆ 주요 영향인자 분석: 설명변수는 종속 변수에 영향을 미치는가?
- ◆ 회귀계수 추정: 영향을 미치는 경우 어떠한 영향을 미치는가?
- ◆ 회귀모델링을 이용한 목표변수 값 추정

<div>종속 변수(Y) 변동 <math>\Sigma(y_i - \bar{y})^2</math></div>	$=$	<div>모형이 설명하는 변동 <math>\Sigma(\hat{y}_i - \bar{y})^2</math></div>	$+$	<div>모형이 설명 못하는 변동 <math>\Sigma(y_i - \hat{y}_i)^2</math></div>
---	-----	---	-----	---

통계모형은 과학적 진실이기 보다는 사실에 대한 대표적 모형이므로 설명변수에 의해 설명되지 못하는 부분인 잔차의 분포는  $iid \sim N(0, \sigma^2)$ 을 가정한다.



## 회귀분석에대한 가정



### 조건2의 준수여부를 확인하려면 잔차분석이 필요함

- 정규성 : 주어진 함수식에 의해 설명된 체계적 관계를 제외하면 측정/관측에서 발생한 오차에는 특정한 패턴이 없다.
- 등분산성 : 고려한 X의 범위에서 추정의 정도(Precision)는 동일하다.
- 독립성 : 개별관측값들은 서로 아무런 관계가 없다.(시계열 Data에 대해서만 독립성 여부 검증 실시)

# Prediction - Regression

다중 회귀모형에 대해 회귀계수의 유의성 검정( $H_0: \beta_0 = \beta_1 = \dots \beta_p = 0 \rightarrow$  설명변수가 모두 유의하지 않다)은 F-검정을 실시한다.

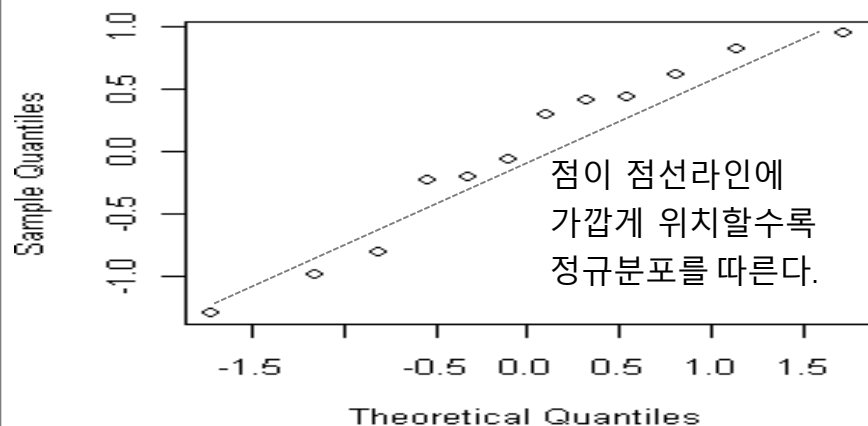
변동 (source)	SS(자승합)	Df (자유도)	MS (평균 자승합)	EMS (기대 평균 자승합)
Regression (모형, 회귀)	$SSR = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2$	$p = 1$	$MSR = SSR / p$	$E(MSR) = \sigma^2 + \beta^2 \sum (X_i - \bar{X})^2$
Error (오차)	$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$	$n - p - 1 = n - 2$	$MSE = SSE / (n - 2)$	$E(MSE) = \sigma^2$
Total (총변동)	$SST = \sum_{i=1}^n (Y_i - \bar{Y})^2$	$n - 1$		$F = \frac{MSR}{MSE} \sim F(1, n - 1)$

# Prediction - Regression

회귀분석에 있어 잔차분석은 모형의 적합성을 판단하는데 중요한 역할을 한다.

정규성검정

Normal Q-Q Plot



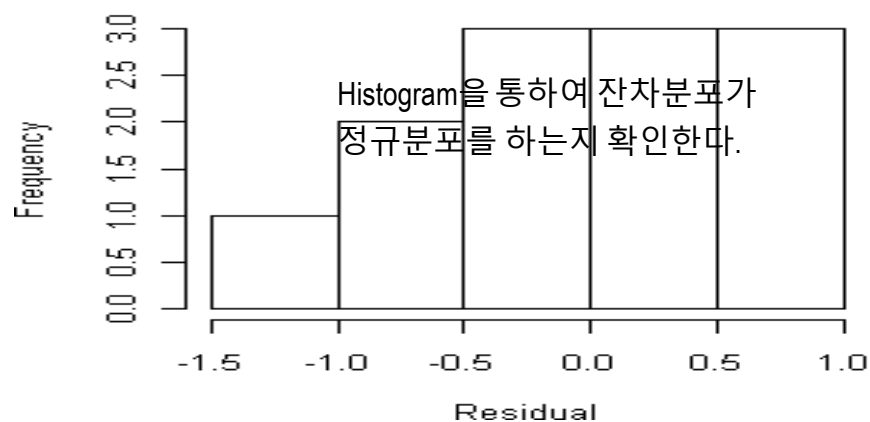
등분산성

Versus Fit



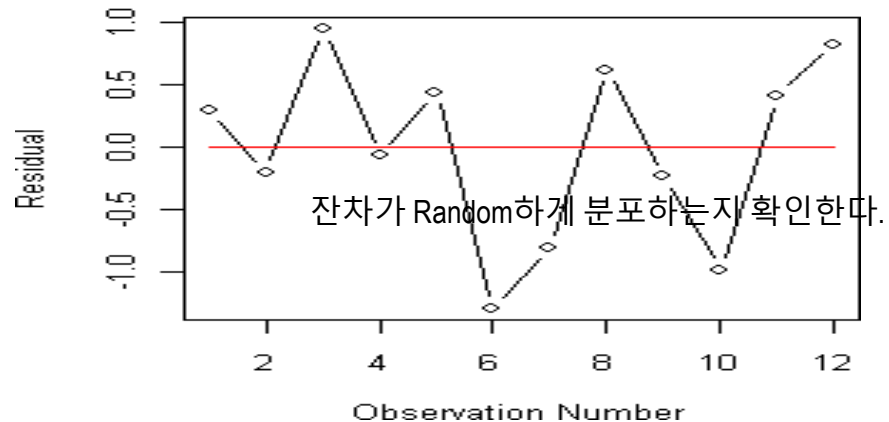
정규성검정

Histogram of residual



독립성

Versus Order



R에서 회귀분석은 다음과 같이 실행한다.

## □ 회귀분석 모델을 생성

- lm함수를 사용하여 회귀분석 모델을 생성한다.

## □ 해당 모델에 대한 통계적 유의성 여부 확인

1) F-test : 모델의 통계적 유의성 판단

- F통계량의 p-value의 값이 0.05보다 작으면, 해당 모델이 유의하다고 볼 수 있음

2) p-value : 입력변수의 통계적 유의성 판단

- p-value의 값이 0.05보다 작으면 해당 변수는 유의하게 결과 변수를 설명한다고 볼 수 있음

3) adjusted-r 제곱

- 모델이 예측대상 변수의 변동량을 몇 %를 설명하는지 확인함

## □ partial F-test를 통하여 입력변수 추가/삭제에 따른 유의차를 분석함

## □ predict함수를 이용하여 테스트 데이터 셋에 대한 예측값을 계산하고 검토함

# Prediction - Regression

R에서 회귀분석 모델링 방법과 결과 해석(모델유효성, 입력변수 유효성 여부) 방법은 하기와 같다.

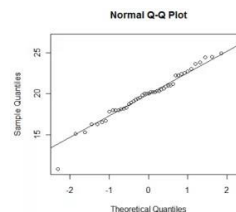
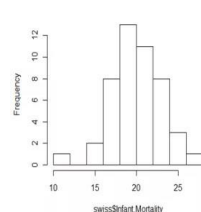
## 코드

```
> require(datasets); require(ggplot2)
> data(swiss)
> str(swiss)
> summary(swiss)

> hist(swiss$Infant.Mortality)
> qqnorm(swiss$Infant.Mortality)
> qqline(swiss$Infant.Mortality)

> model<-lm(Infant.Mortality~. ,data=swiss)
> summary(model)
```

## 실행결과



예측값  
변수의  
정규성 검토

```
Call:
lm(formula = Infant.Mortality ~ ., data = swiss)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.2512	-1.2860	0.1821	1.6914	6.0937

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	8.667e+00	5.435e+00	1.595	0.11850
Fertility	1.510e-01	5.351e-02	2.822	0.00734 **
Agriculture	-1.175e-02	2.812e-02	-0.418	0.67827
Examination	3.695e-02	9.607e-02	0.385	0.70250
Education	6.099e-02	8.484e-02	0.719	0.47631
Catholic	6.711e-05	1.454e-02	0.005	0.99634

---  
signif. codes:

0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.683 on 41 degrees of freedom  
Multiple R-squared: 0.2439, Adjusted R-squared: 0.1517  
F-statistic: 2.645 on 5 and 41 DF, p-value: 0.03665

F-test결과 p-value의 값이 0.3665로 유의수준 0.05보다 작다. → 모델이 infant.Mortality를 예측하는데 사용가능하다. 입력변수 중에서 Fertility만이 p-value의 값이 0.00734로 유의수준 0.05보다 작다. → infant.Mortality를 설명하는데 사용할

R에서 복수개의 회귀분석 모델링에 대한 유의차 분석은 하기와 같이 실행한다.

## 코드

```
#Fertility만을 입력변수로 고려한 회귀예측 모델 생성
model_simple<-lm(Infant.Mortality~Fertility ,data=swiss)
#기존 회귀예측 모델과의 유의차 검증
anova(model, model_simple)
```

## 실행결과

Analysis of Variance Table

```
Model 1: Infant.Mortality ~ Fertility + Agriculture + Examination + Education +
  Catholic
Model 2: Infant.Mortality ~ Fertility
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     41 295.07
2     45 322.54 -4   -27.472 0.9543 0.4427
```

두 모델의 설명력 유의차 여부에 대해 ANOVA로 분석한 결과  
P-value가 0.4427로  
두 모델간의 설명력에는 차이가 없다고 볼 수 있다.

# Prediction - Regression

R에서 생성한 회귀분석 모델링을 이용하여 별도의 데이터를 대상으로 예측하는 방법은 하기와 같다.

## 코드

```
> new_Fertility<-rnorm(10, mean=mean(swiss$Fertility), sd=sd(swiss$Fertility))  
> new_Fertility<-as.data.frame(new_Fertility)  
> colnames(new_Fertility)<-c("Fertility")  
> predict(model_simple, new_Fertility, interval="prediction")
```

## 실행결과

	fit	lwr	upr
1	19.58662	14.13237	25.04088
2	16.93327	11.13820	22.72834
3	19.83726	14.38756	25.28696
4	22.22750	16.57628	27.87871
5	19.94182	14.49256	25.39108
6	16.64577	10.78392	22.50762
7	19.92786	14.47859	25.37713
8	20.37420	14.91760	25.83080
9	21.70586	16.13545	27.27628
10	17.49798	11.81816	23.17780

새로운 10개의 Fertility 변수에 대한  
예측 값은 fit 컬럼에,  
95% 신뢰구간은  
lwr/up에서 확인할 수 있다.

# Prediction - Instance based learning(KNN)

비선형 예측/분류 모델링에서 자주 사용하는 방법으로 knn이 있으며 사용예제는 하기와 같다.

(knn이란? 대상 개체에 근접한 k개의 개체를 이용하여 분류/예측 문제를 해결하는 모델링 방법이다.)

## 코드

```
Library(kknn)
## iris 데이터셋을 불러오기
data(iris)
## 데이터를 훈련+테스트셋으로 구분하기
idxs <- sample(1:nrow(iris),as.integer(0.7*nrow(iris)))
trainIris <- iris[idxs,]
testIris <- iris[-idxs,]
## knn모델링(k=3, 정규화 적용하지 않음)
nn3 <- kknn(Species ~ .,trainIris,testIris,k=1) #
# 혼동행렬(confusion matrix)생성
table(testIris[, 'Species'],nn3$fitted.values)
```

	setosa	versicolor	virginica
setosa	14	0	0
versicolor	0	17	1
virginica	0	0	13

## 실행결과

```
## knn모델링(k=5, 정규화 적용함)
nn5 <- kknn(Species ~ .,trainIris,testIris,k=5)
## 혼동행렬(confusion matrix)생성
table(testIris[, 'Species'],nn5$fitted.values)
```

	setosa	versicolor	virginica
setosa	14	0	0
versicolor	0	17	1
virginica	0	0	13



비선형 예측/분류 모델링에서 자주 사용하는 방법으로 ANN이 있으며 사용예제는 하기와 같다.

(ANN이란? Artificial Neural Network의 약어로 신경망 모델을 이용하여 분류/예측 문제의 해결모델링이다.)

## 코드 및 실행결과

```
> # nnet 패키지를 불러온다.
> library(nnet)
> # 입력데이터를 설정한다.
> input<-matrix(c(0,0,1,1,0,1,0,1),ncol=2)
> input
      [,1] [,2]
[1,]  0  0
[2,]  0  1
[3,]  1  0
[4,]  1  1

> # 출력데이터를 설정한다.
> output<-matrix(c(0,1,1,0))
> output
      [,1]
[1,] 0
[2,] 1
[3,] 1
[4,] 0
```

nnet함수에 대한 Parameter

- maxit : max iteration

- weight decay parameter : overfitting을 피하기 위해 사용함

## 코드 및 실행결과

```
> # 신경망 모델링을 생성한다.
> # 최대 iteration회수는 100으로 설정한다.
> ann<-nnet(input,output,maxit=100, size=2, decay=0.001) # weights: 9
initial value 1.181533
iter 10 value 1.000386
iter 20 value 1.000191
iter 30 value 1.000127
iter 40 value 1.000007
final value 1.000000 converged
> # 생성된 신경망의 구조를 본다.
> ann
a 2-2-1 network with 9 weights
options were- decay=0.001
> #생성된 모델링에 input값을 입력하여 예측한 결과를 본다.
> result<-predict(ann, input)
> result

      [,1]
[1,] 0.4999999
[2,] 0.4999999
[3,] 0.4999999
[4,] 0.4999999
```

학습모델링을 개발하는 방법은 크게 ERM과 SRM으로 구분할 수 있으며, 상세 내용은 하기와 같다.

## □ ERM : Empirical Risk Minimization

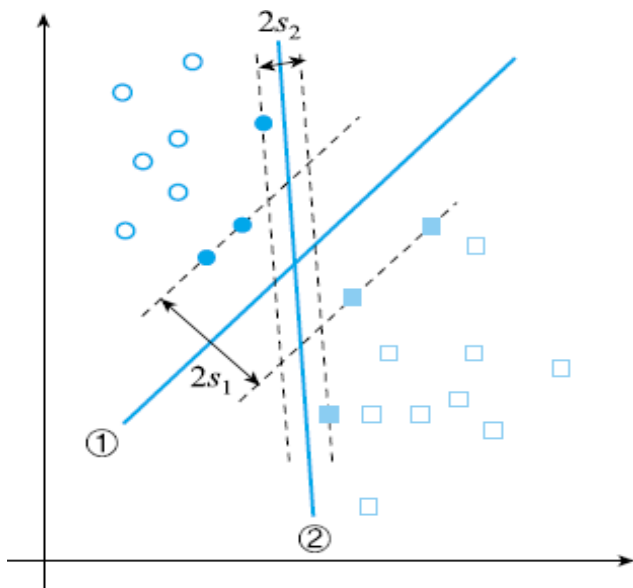
- 모델링 : 분류 오류율을 최소화하는 모델을 개발
- 특징
  - 관찰과 경험을 통하여 수집할 수 있는 데이터가 현실을 100%반영한다고 보기 어려움
  - Training Data의 분류/예측 오류 최소화에만 중점을 둠 → 일반화 능력 떨어짐  
(Training Data에는 적합도가 높으나 미학습된 신규 Data에 대한 예측능력은 낮음)
- 예 : 회귀모델링, 신경망

## □ SRM : Structural Risk Minimization

- 모델링 : 개별 부류 사이에 존재하는 공간을 최대화하는 모델을 개발
- 특징
  - 미학습된 신규 Data에 대한 예측능력이 상대적으로 뛰어남
- 예 : SVM(Support Vector Machine)

SVM의 분석은 아래와 같은 프로세스에 근거하여 수행할 수 있다.

## 여백과 Support Vector 정의



- 서포트 벡터(Support Vector)
  - 직선에서 가장 가까운 샘플
- 여백(Margin)
  - 직선에서 서포트 벡터까지 거리 \* 2

## 최적 분류 초평면(Hyperplane) 계산

### ▪ 결정 초평면(Hyperplane) 정의

- 초평면 :  $WX + b$
- 방향 :  $W$
- 위치 :  $b$

### ▪ 최적 초평면 계산

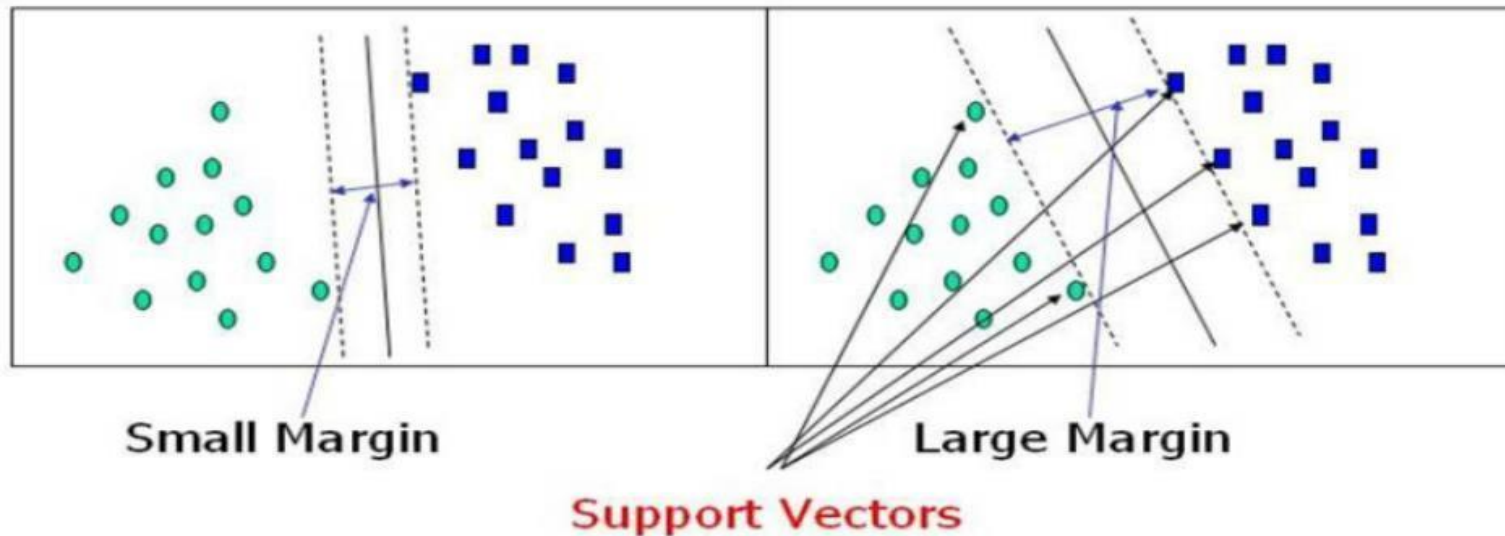
- 목적함수 : 여백최대화
- 제한조건 : 가능해가 존재하는 영역
- 제한조건을 고려한 목적함수의 최적화 문제해결

# Prediction - SVM

SVM이란 Support Vector을 이용하여 비선형 예측/분류문제를 해결하는 방법이다.

## □ Support Vector

- 우리가 사각형과 원 샘플을 구별하고 싶다면 어떤 식으로 나뉘야 하는가?
- 만약 선을 그어 그 사이를 나눈다면 어떤 선이어야 할 것인가?



샘플로만 보면 두가지 방법 모두 큰 차이가 없어 보인다.

하지만 small margin와 large margin의 경우 정확도와 일반화 측면에서 다음과 같은 차이가 있다.

	Small margin	Large margin
정확도	高	高
일반화	低	高

따라서 최적모델링을 개발하기 위해서는 샘플사이의 Margin을 최대한 만드는 구분선을 구해야 하며 이때 경계Line상에 있는 샘플을 Support Vectors라고 한다.

# Prediction - SVM

비선형 예측/분류 모델링에서 자주 사용하는 방법으로 SVM이 있으며 사용예제는 하기와 같다.

(SVM은 support vector machine의 약어로 support vector을 이용하여 분류/예측 문제를 해결하는 모델링이다.)

## 코드

```
> library(kernlab)
> data('spam')
> summary(spam)
> table(spam$type)
> index<-1:nrow(spam)
> testindex<-sample(index,trunc(length(index)/3))
> testset<-spam[testindex,]
> trainingset<-spam[-testindex,]

> model<-svm(type~.,data=trainingset,
+ method='C-classification',
+ kernel='radial',
+ cost=10,
+ gamma=0.1)
> summary(model)
```

## 실행결과

```
Call:
svm(formula = type ~ ., data = trainingset, method = "C-classification",
     kernel = "radial", cost = 10, gamma = 0.1)

Parameters:
  SVM-Type:  C-classification
 SVM-kernel: radial
      cost:  10
    gamma:  0.1

Number of Support Vectors: 1537
( 642 895 )

Number of classes: 2

Levels:
nonspam spam

> pred<-predict(model, testset)
> table(pred,testset$type)

  pred      nonspam spam
nonspam    882   102
spam       37   512

> (882+512)/(882+102+37+512)
[1] 0.9093281
```

# R Cheat sheet - Machine Learning Package

## Supervised & Unsupervised Learning

	ALGORITHM	DESCRIPTION	R PACKAGE::FUNCTION	SAMPLE CODE
SUPERVISED LEARNING	<b>NBC</b> Naïve Bayes classifier	A classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature	e1071::naiveBayes	naiveBayes(class ~ ., data = x)
	<b>KNN</b> k-Nearest Neighbours	A non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression	class::knn	knn(train, test, cl, k = 1, l = 0, prob = FALSE, use.all = TRUE)
	<b>REG</b> Linear Regression	Model the linear relationship between a scalar dependant variable Y and one or more explanatory variables (or independent variables) denoted X	stats::lm	lm(dist ~ speed, data=cars)
	<b>LREG</b> Logistic Regression	Used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables.	stats::glm	glm(Y ~ ., family = binomial (link = 'logit'), data = X)
	<b>TM</b> Tree-Based Models	The idea is to consecutively divide (branch) the training dataset based on the input features until an assignment criterion with respect to the target variable into a "data bucket" (leaf) is reached	rpart::rpart	rpart(Kyphosis ~ Age + Number + Start, data = kyphosis)
	<b>ANN</b> Artificial Neural Network	Neural networks are built from units called perceptrons. Perceptrons have one or more inputs, an activation function and an output. An ANN model is built up by combining perceptrons in structured layers.	neuralnet::neuralnet	neuralnet(f,data=train_,hidden=c(5,3),linear.output=T)
UNSUPERVISED LEARNING	<b>SVM</b> Support Vector Machine	A data classification method that separates data using hyperplanes	e1071::svm	svm(formula, data = NULL, ..., subset, na.action = na.omit, scale = TRUE)
	<b>PCA</b> Principal Component Analysis	A procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.	stats::prcomp stats::princomp FactoMineR::PCA ade4::dudi.pca amap::acp	<b>stats</b> : prcomp(formula, data = NULL, subset, na.action, ...) <b>stats</b> : princomp(formula, data = NULL, subset, na.action, ...) <b>FactoMineR</b> : PCA(decathlon, quanti.sup = 11:12, quali.sup=13) <b>ade4</b> : dudi.pca(deug\$tab, center = deug\$cent, scale = FALSE, scan = FALSE) <b>amap</b> : acp(lubisch)
	<b>kMC</b> k-Mean Clustering	Aims at partitioning n observations into k clusters in which each observation belongs to the cluster with the nearest mean	stats::kmeans	kmeans(x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"), trace=FALSE)
	<b>HCL</b> Hierarchical Clustering	An approach which builds a hierarchy from the bottom-up, and doesn't require the number of clusters to be specified beforehand.	stats::hclust	hclust(d, method = "complete", members = NULL)



# R Cheat sheet - Machine Learning Package

## Meta-Algorithm, Time Series & Model Validation

	ALGORITHM	DESCRIPTION	R PACKAGE::FUNCTION	SAMPLE CODE
META ALGORITHM	<b>REGU</b> Regularisation L1 (Lasso) L2 (Ridge)	Regularization adds a penalty on the different parameters of a model to reduce the freedom of the model. Hence, the model will be less likely to fit the noise of the training data and will improve the generalization abilities of the model	glmnet::glmnet	<b>L1</b> : glmnet(myMatrixA, myMatrixB, family = "gaussian", alpha = 1) <b>L2</b> : glmnet(myMatrixA, myMatrixB, family = "gaussian", alpha = 0)
	<b>BOO</b> Boosting	A process of iteratively refining, e.g. by reweighting, of estimated regression and classification functions (though it has primarily been applied to the latter), in order to improve predictive ability.	<b>Parametric model</b> - mboost::glmboost	glmboost(Yen ~ ., data = curr1[trnidxs,])
	<b>BAG</b> Bagging	Bagging is a way to increase the power of a predictive statistical model by taking multiple random samples (with replacement) of the training data set, and using each of them to construct a separate model and separate predictions for the original test set	<b>All models</b> : foreach <b>Tree models</b> : ipred::bagging	<b>foreach</b> : d <- data.frame(x=1:10, y=rnorm(10)) s <- foreach(d=iter(d, by='row'), .combine=rbind) %dopar% identical(s, d) <b>ipred</b> : bagging(formula, data, subset, na.action=na.rpart, \dots)
	<b>PRU</b> Pruning	Pruning is a technique that reduces the size of decision tree by removing sections of the tree that provide little power to classify instances. Pruning reduces the complexity of the final classifier and hence improves predictive accuracy by reducing overfitting	rpart::prune	prune(x, cp = 0.1)
	<b>RFO</b> Random Forrest	An ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression)	randomForest::randomForest	randomForest(X ~ ., data = Y, subset = mySub)
TIME SERIES	<b>STS</b> Lead-lag analysis, Auto-correlation, Spectral analysis, Time series clustering, Seasonality, Trend....	Random sampling of observations for training and testing a model can be an issue when faced with a times dimension. Random sampling may either destroy serial correlation properties in the data which we would like to exploit	stats xts forecast spectral TTR ....	<b>Auto-correlation</b> : acf(x, lag.max = NULL, type = c("correlation", "covariance", "partial")) <b>Spectral Analysis</b> : spec.pgram(myTs, spans = NULL) <b>Seasonal Decomposition of Time Series</b> - stl(x, s.window = 7, t.window = 50, t.jump = 1) .....
MODEL VALIDATION	<b>PM</b> Performance metrics	Depends on the problem: • <b>Regression</b> : squared errors, outliers, error rate... • <b>Classification</b> : Accuracy, precision, recall, F-score...	<b>Regression</b> -stats::outlierTest, stats:: qqPlot .... <b>Classification</b> -ROCR:: <b>Tree</b> : caret:: confusionMatrix	<b>Regresion</b> : fit <- lm(Y~X,data=myData) outlierTest(fit) qqPlot(fit, main="QQ Plot")
	<b>BVT</b> Biias-Variance Tradeoff	• Simple models with few parameters are easier to compute but may lead to poorer fits ( <b>high bias</b> ). • Complex models may provide more accurate fits but may over-fit the data ( <b>high variance</b> )	Tailored to the analysis	Tailored to the analysis
	<b>CV</b> Cross validation	Cross validation compares the test performances of different model realisations with different sets or values of parameters	caret::createDataPartition caret::createFolds	createDataPartition(classes, p = 0.8, list = FALSE)
	<b>LC</b> Learning Curves	Learning curves plot a model's training and test errors, or the chosen performance metric, depending on the training set size	caret:: learing_curve_dat	learing_curve_dat(dat, outcome = NULL, proportion = (1:10)/10, test_prop = 0, verbose = TRUE, ...)