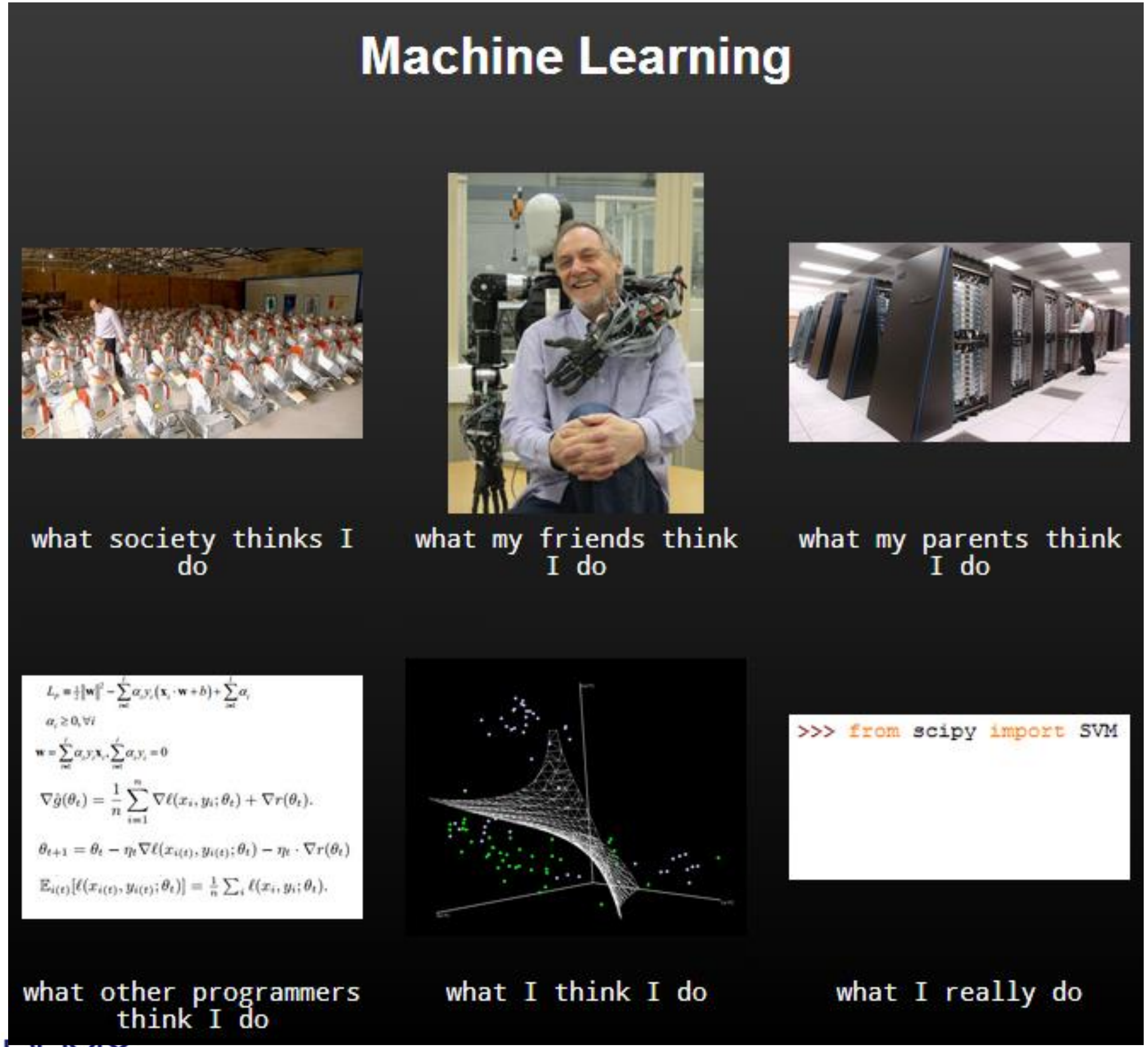
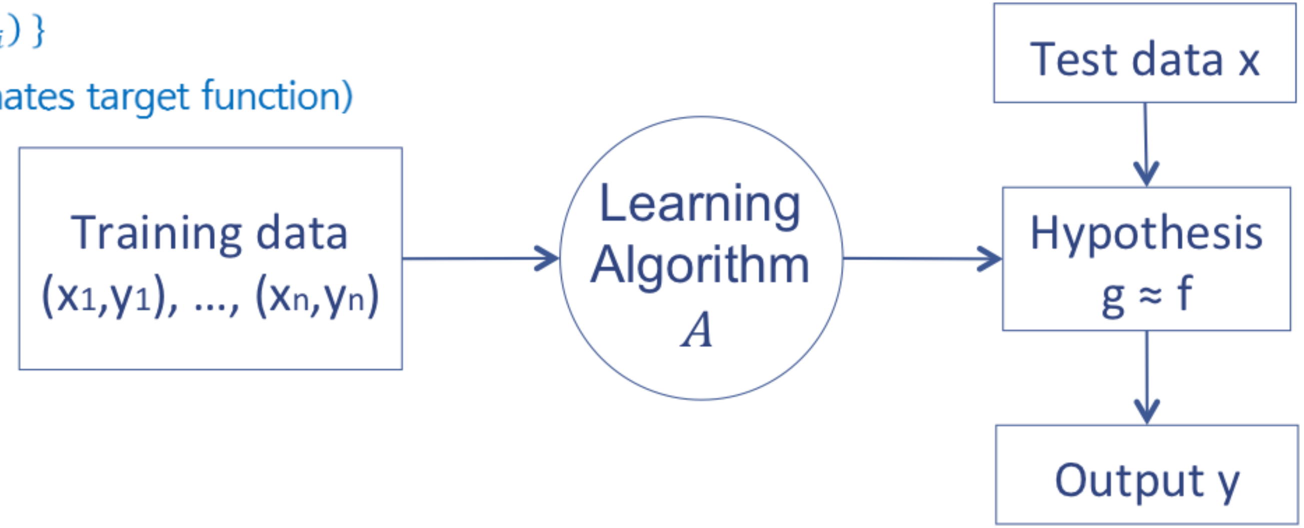
Machine Learning with R

Machine Learning

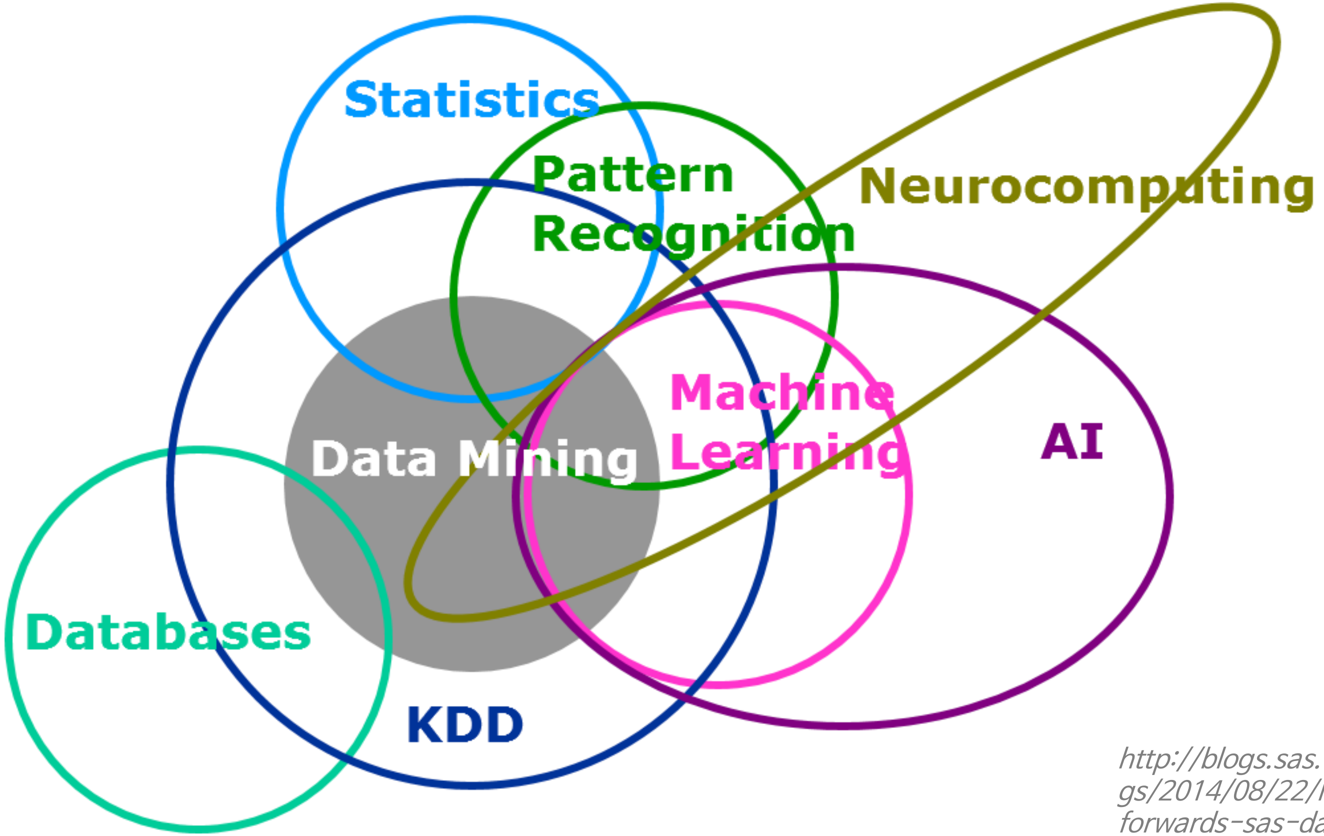


http://sentdex.com/sentiment-analysisbig-data-and-python-tutorials-algorithmic-trading/machine-learning-algorithmic-trading-automated-trading/

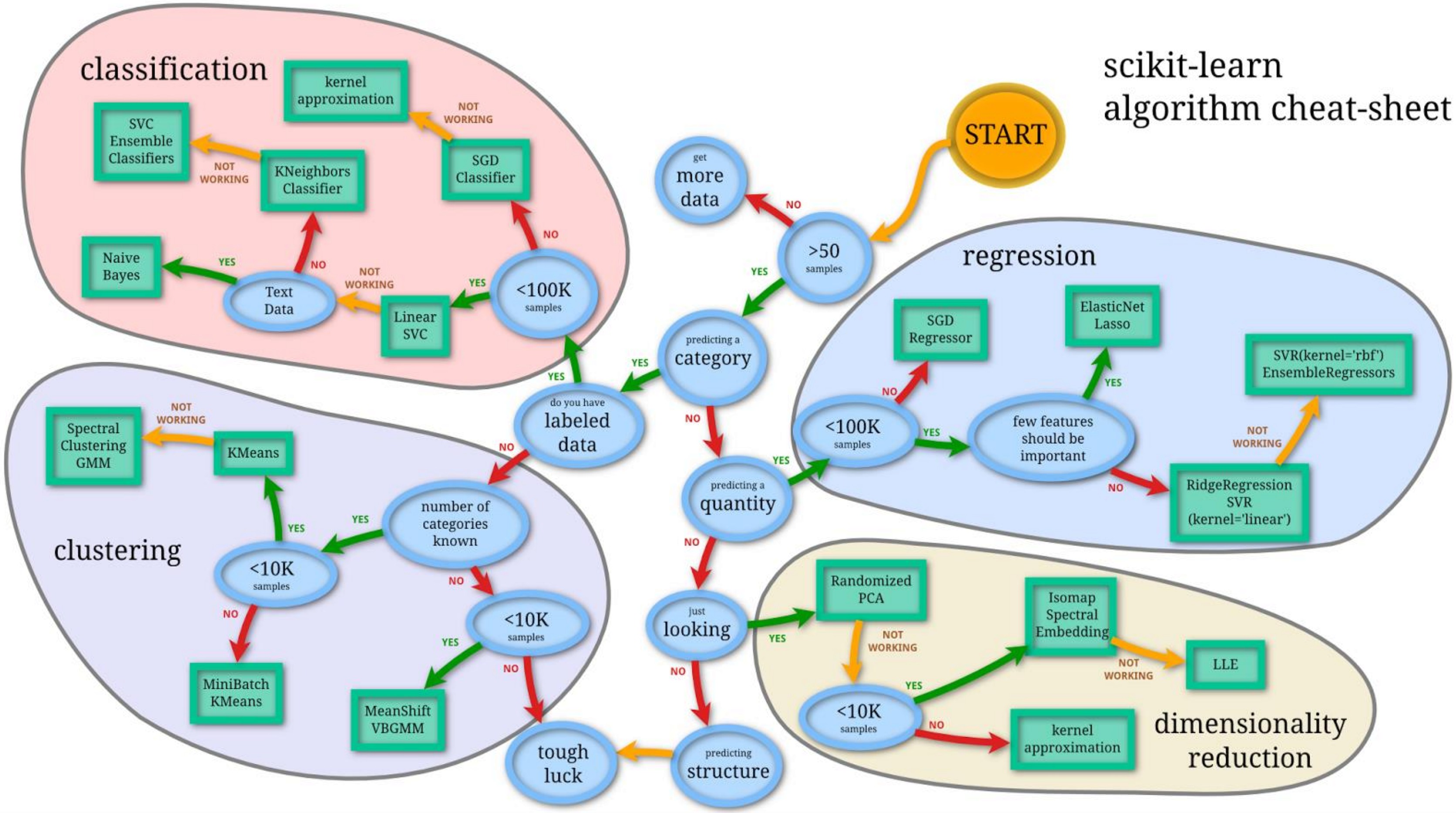
- 컴퓨터에게 배울 수 있는 능력, 즉 코드로 정의하지 않은 동작을 실행하는 능력에 대한 연구 분야(Arthur Samuel, 1959)
- Machine Learning : 사람이 직접 명시적으로 Logic을 지시하지 않아도 데이터를 통해 컴퓨터가 학습을 하고 그것을 사용해 컴퓨터가 자동으로 문제를 해결하도록 하는 것
- 주어진 데이터 X = $(x_1, x_2, x_3, \dots, x_n)$와 각 데이터에 대응하는 실제 현상 Y = $(y_1, y_2, y_3, \dots, y_n)$ 에 대한 관계 function f 를 찾는 과정. 이때, 정확한 함수 f를 찾기 위해 데이터에 대한 가정을 하고, 그 가정에 따라 주어진 데이터를 최대한 잘 설명할 수 있는 함수 f' 를 찾는다. (f' : Hypothesis)
    - Set of possible instance(domain) : X, Output : Y, Unknown target function : $f : X \to Y$
    - Set of hypothesis function space : $\mathrm{H} \in \{ h | h : X \to Y \}$
    - Input : Training example $\{ (x_i, y_i) \}$
    - Output : $\mathrm{h} \in H$ (best approximates target function)



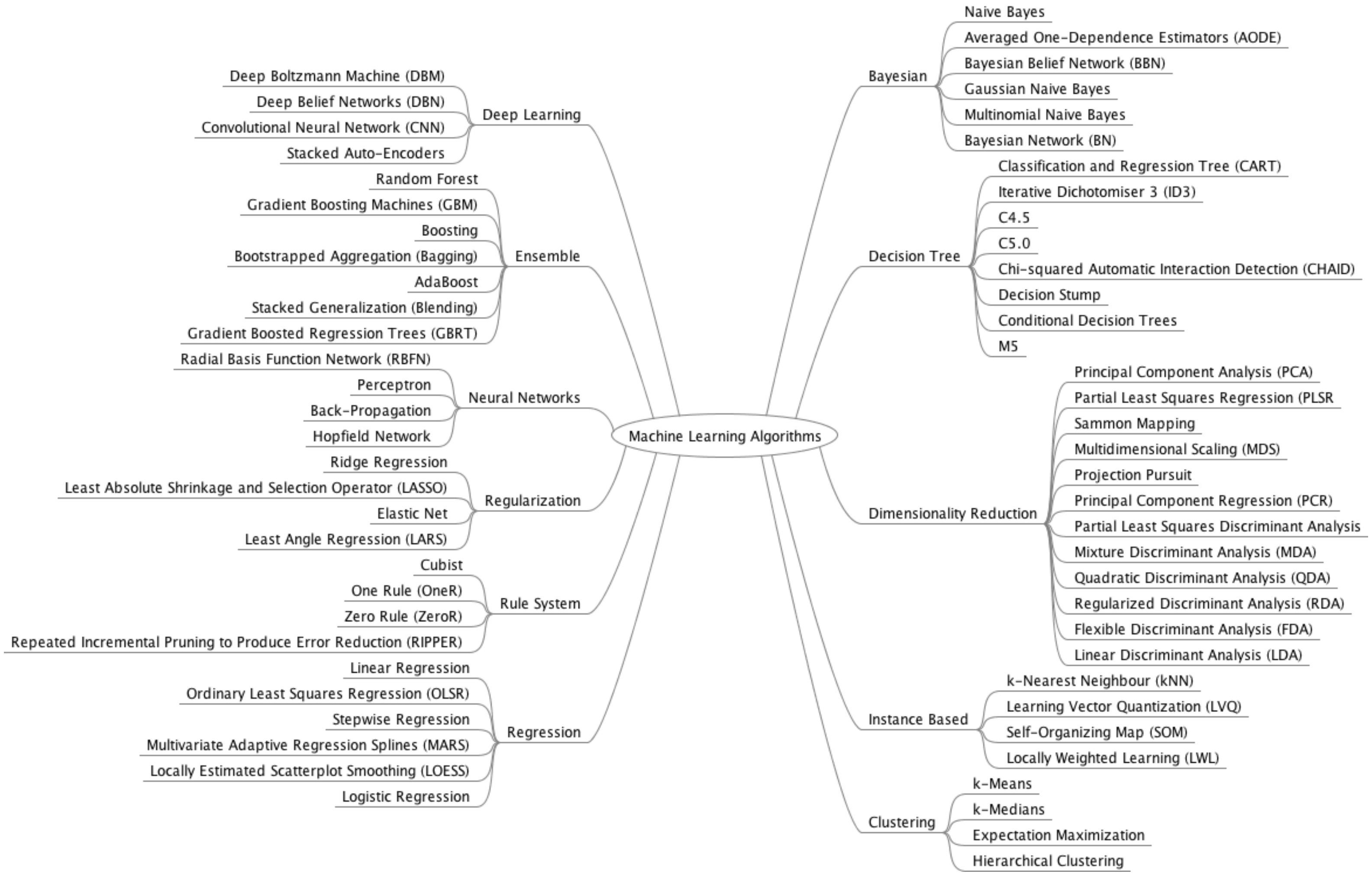
- Data Mining : 데이터의 미처 몰랐던 속성을 발견하는 것
- Deep Learning : 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화를 시도하는 기계학습 알고리즘의 집합

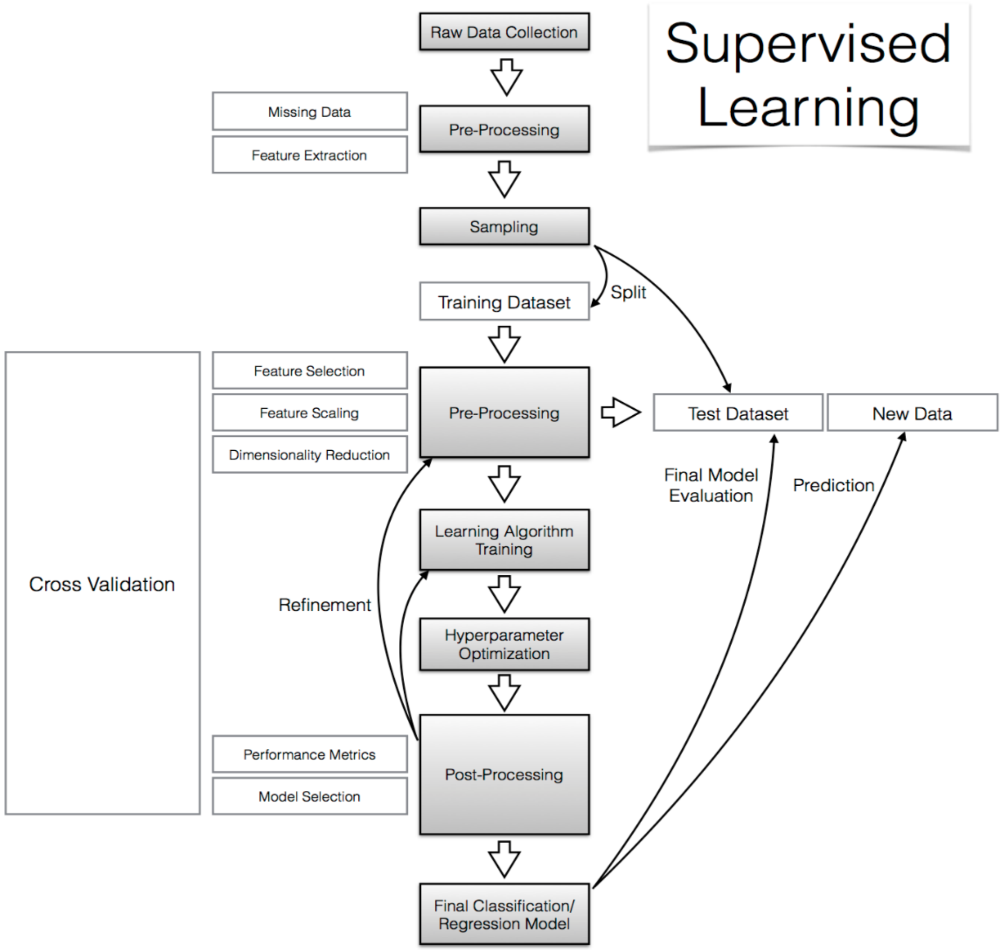


http://blogs.sas.com/content/subconsciousmusings/2014/08/22/looking-backwards-looking-forwards-sas-data-mining-and-machine-learning/



http://radimrehurek.com/data_science_python/

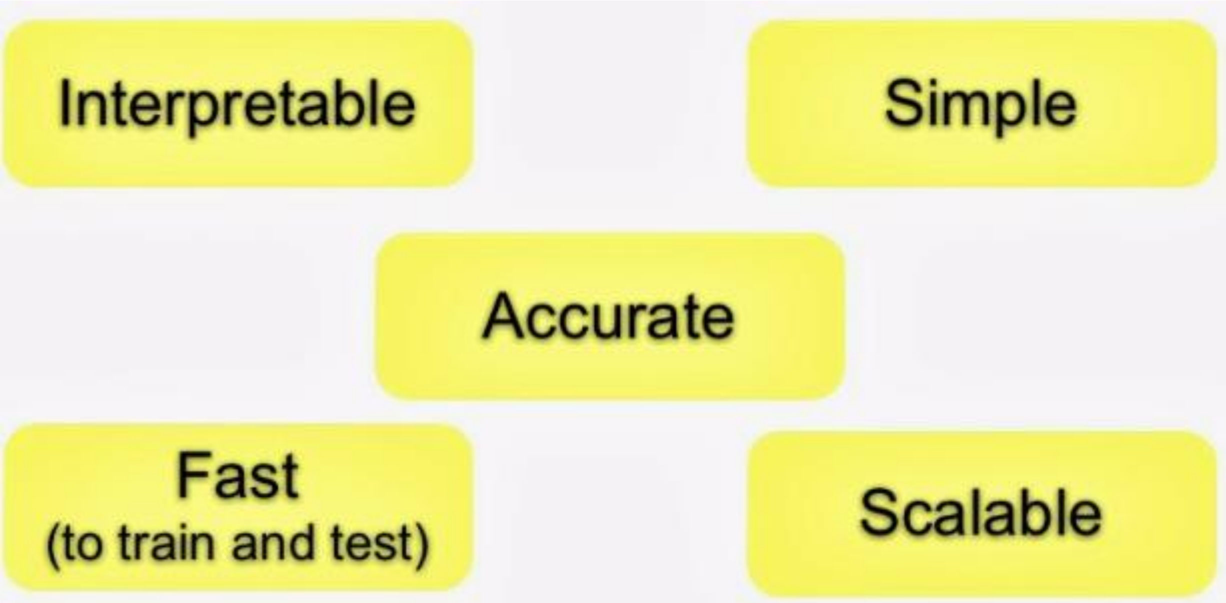

Machine Learning Algorithms
Deep Learning
Deep Boltzmann Machine (DBM)
Deep Belief Networks (DBN)
Convolutional Neural Network (CNN)
Stacked Auto-Encoders
Ensemble
Random Forest
Gradient Boosting Machines (GBM)
Boosting
Bootstrapped Aggregation (Bagging)
AdaBoost
Stacked Generalization (Blending)
Gradient Boosted Regression Trees (GBRT)
Neural Networks
Radial Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Regularization
Ridge Regression
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Rule System
Cubist
One Rule (OneR)
Zero Rule (ZeroR)
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
Regression
Linear Regression
Ordinary Least Squares Regression (OLSR)
Stepwise Regression
Multivariate Adaptive Regression Splines (MARS)
Locally Estimated Scatterplot Smoothing (LOESS)
Logistic Regression
Bayesian
Naive Bayes
Averaged One-Dependence Estimators (AODE)
Bayesian Belief Network (BBN)
Gaussian Naive Bayes
Multinomial Naive Bayes
Bayesian Network (BN)
Decision Tree
Classification and Regression Tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5
C5.0
Chi-squared Automatic Interaction Detection (CHAID)
Decision Stump
Conditional Decision Trees
M5
Dimensionality Reduction
Principal Component Analysis (PCA)
Partial Least Squares Regression (PLSR
Sammon Mapping
Multidimensional Scaling (MDS)
Projection Pursuit
Principal Component Regression (PCR)
Partial Least Squares Discriminant Analysis
Mixture Discriminant Analysis (MDA)
Quadratic Discriminant Analysis (QDA)
Regularized Discriminant Analysis (RDA)
Flexible Discriminant Analysis (FDA)
Linear Discriminant Analysis (LDA)
Instance Based
k-Nearest Neighbour (kNN)
Learning Vector Quantization (LVQ)
Self-Organizing Map (SOM)
Locally Weighted Learning (LWL)
Clustering
k-Means
k-Medians
Expectation Maximization
Hierarchical Clustering



Sebastian Raschka 2014

This work is licensed under a Creative Commons Attribution 4.0 International License.

http://sebastianraschka.com/Articles/2014_intro_supervised_learning.html question > input data > features > algorithm > parameter > evaluation



R Programming

❑ Data Structure

	Homogeneous	Hetrogeneous
1d	Atomic Vector	List
2d	Matrix	Data frame
nd	Array	

❑ Vector

intV = c(1,2,3);intV

[1] 1 2 3

charV = c(1, "a", 3); charV

[1] "1" "a" "3" > doubleV = c(1, 2, 3.0);

doubleV = c(1, 2, 3.5); doubleV

[1] 1.0 2.0 3.5

booleanV = c(T, F, TRUE); booleanV

[1] TRUE FALSE TRUE

as.numeric(booleanV)

[1] 1 0 1

attr(booleanV, "desc") = "This is boolean Vector"

booleanV

[1] TRUE FALSE TRUE

attr(,"desc")

[1] "This is boolean Vector"

str(booleanV)

atomic [1:3] TRUE FALSE TRUE

- attr(*, "desc")= chr "This is boolean Vector"

BIGDATA ACADEMY 빅데이터아카데미

❑ List

```
x = list(1:3, "a", c(TRUE, FALSE, TRUE), c(2.3, 5.9));x
[[1]]
[1] 1 2 3
[[2]]
[1] "a"
[[3]]
[1]  TRUE FALSE  TRUE
[[4]]
[1] 2.3 5.9
str(x)
List of 4
 $ : int [1:3] 1 2 3
 $ : chr "a"
 $ : logi [1:3] TRUE FALSE TRUE
 $ : num [1:2] 2.3 5.9
```

❑ Factor

```
x = factor(c("a", "b", "b", "a"));x
[1] a b b a
Levels: a b
class(x)
[1] "factor"
levels(x)
[1] "a" "b“
sex_char = c("m", "m", "m")
sex_factor = factor(sex_char, levels=c("m","f"))
table(sex_char)
sex_char
m
3
table(sex_factor)
sex_factor
m f
3 0
```

❑ Matrix & Array

```
mat = matrix(1:6, ncol = 3, nrow = 2);mat
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
arr = array(1:12, c(2, 3, 2));arr
, , 1
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
, , 2
     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12
```

```
length(mat);length(arr)
[1] 6
[1] 12
nrow(mat);nrow(arr)
[1] 2
[1] 2
ncol(mat);ncol(arr)
[1] 3
[1] 3
```

```
rownames(mat) = c("A", "B"); colnames(mat) = c("a", "b", "c"); mat
  a b c
A 1 3 5
B 2 4 6
dimnames(arr) = list(c("A", "B"), c("a", "b", "c"), c("one", "two")); arr
, , one

  a b c
A 1 3 5
B 2 4 6

, , two

  a  b  c
A 7  9 11
B 8 10 12
```

Data Frame

```
df = data.frame(x=1:3, y=c("a", "b", "c"), stringsAsFactors=FALSE);df
  x y
1 1 a
2 2 b
3 3 c
str(df)
'data.frame':  3 obs. of  2 variables:
 $ x: int  1 2 3
 $ y: chr  "a" "b" "c"
class(df)
[1] "data.frame"
is.data.frame(df)
[1] TRUE
```

```
cbind(df, data.frame(z=3:1))
  x y z
1 1 a 3
2 2 b 2
3 3 c 1
rbind(df, data.frame(x=10,y="z"))
   x y
1  1 a
2  2 b
3  3 c
4 10 z
```

```
data.frame(x = 1:3, y = list(1:2, 1:3, 1:4))
Error in data.frame(1:2, 1:3, 1:4, check.names = FALSE,
stringsAsFactors = TRUE) :
  arguments imply differing number of rows: 2, 3, 4
df = data.frame(x = 1:3)
df$y = list(1:2, 1:3, 1:4)
df
  x          y
1 1       1, 2
2 2    1, 2, 3
3 3 1, 2, 3, 4
df = data.frame(x = 1:3, y = I(list(1:2, 1:3, 1:4)))
df
  x          y
1 1       1, 2
2 2    1, 2, 3
3 3 1, 2, 3, 4
```

❑ Subset

```
a = matrix(1:9, nrow = 3); colnames(a) = c("A", "B", "C");a
     A B C
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
a[1:2,]
     A B C
[1,] 1 4 7
[2,] 2 5 8
a[c(T, F, T), c("B", "A")]
     B A
[1,] 4 1
[2,] 6 3
a[0, -2]
     A C
```

```
a = outer(1:5, 1:5, FUN = "paste", sep = ",");a
     [,1]  [,2]  [,3]  [,4]  [,5]
[1,] "1,1" "1,2" "1,3" "1,4" "1,5"
[2,] "2,1" "2,2" "2,3" "2,4" "2,5"
[3,] "3,1" "3,2" "3,3" "3,4" "3,5"
[4,] "4,1" "4,2" "4,3" "4,4" "4,5"
[5,] "5,1" "5,2" "5,3" "5,4" "5,5"
select = matrix(ncol = 2, byrow = TRUE, c(1,1,3,1,2,4));select
     [,1] [,2]
[1,]    1    1
[2,]    3    1
[3,]    2    4
a[select]
[1] "1,1" "1,2" "3,4"
```

Subset

```
df = data.frame(x = 1:3, y = 3:1, z = letters[1:3]);df
  x y z
1 1 3 a
2 2 2 b
3 3 1 c
df[df$x == 2, ]
  x y z
2 2 2 b
df[c("x", "z")]
  x z
1 1 a
2 2 b
3 3 c
```

```
df[,c("x", "z")]
  x z
1 1 a
2 2 b
3 3 c

str(df["x"])
'data.frame':  3 obs. of  1 variable:
 $ x: int  1 2 3
str(df[,"x"])
 int [1:3] 1 2 3
```

❑ Subset

	Simplifying	Preserving
Vector	x[[1]]	x[1]
List	x[[1]]	x[1]
Factor	x[1:4, drop =T]	x[1:4]
Array	x[1,] / x[, 1]	x[1, , drop = F] / x[, 1, drop = F]
Data frame	x[, 1] / x[[1]]	x[, 1, drop = F] / x[1]

```
a = list(a=1, b=2);a
$a
[1] 1
$b
[1] 2
```

```
a[1]
$a
[1] 1
a[[1]]
[1] 1
```

```
a["a"]
$a
[1] 1
a[["a"]]
[1] 1
```

❑ Out of bound index

operator	index	Atomic	List
[	oob	NA	list(NULL)
[	NA_real_	NA	list(NULL)
[	NULL	x[0]	list(NULL)
[[	oob	Error	Error
[[	NA_real_	Error	NULL
[[	NULL	Error	Error

❑ Assignment

```
x = 1:5;x
[1] 1 2 3 4 5
x[c(2,4)] = c(9,23);x
[1]  1  9  3 23  5
x[-1] = 99;x
[1]  1 99 99 99 99
```

```
df = data.frame(a = c(1, 10, NA));df
   a
1  1
2 10
3 NA
```

```
df$a[df$a < 5] = 0;df
   a
1  0
2 10
3 NA
> df$a
[1]  0 10 NA
```

Function

- Variable Scope - Dynamic loopup

```
f = function() x
f()
x = 15
f()
x = 20
f()
```

where? when?

```
f = function() {
  i = 10
  x
  cat(paste0(i,",",x))

}
codetools::findGlobals(f)
```

external dependencies of function

- Function call with argument

```
f <- function(abcdef, bcde1, bcde2) {
  list(a = abcdef, b1 = bcde1, b2 = bcde2)
}
str(f(1, 2, 3))
str(f(2, 3, abcdef = 1))
str(f(2, 3, a = 1))
str(f(1, 3, b = 1))
```

- Function call with list argument

```
mean(1:10, na.rm = TRUE)
args = list(1:10, na.rm = TRUE)
mean(args)
do.call(mean, args)
```

- Default argument

```
f <- function(a = 1, b = a * 2) {
  c(a, b)
}

f()
f(3)
f(3,5)
```

- Lazy evaluation

```
f <- function(x) {
  10
}
f()

f <- function(x) {
  force(x)
  10
}
f()
```

Function

- Replcement function

```
second <- function(x,
value) {
  x[2] <- value
  x
}
x = 1:10
second(x) = 5L

`second<-`<- function(x,
value) {
  x[2] <- value
  x
}
x = 1:10
second(x) = 5L;x
```

- on.exit

```
in_dir <- function(dir, code) {
  old <- setwd(dir)  # return old working dir
  on.exit(setwd(old))

  force(code)
}
getwd()
in_dir("/", getwd())
getwd()
```

Functional programming

- Imperative Programming : mutable variables, assignments, control structure(if-then-else, loop, break, continue, return) – C++, Java
- Logic Programming : formal logic – Prolog, Answer set programming(ASP)
- Functional Programming
    - restricted sense : not use imperative programming paradigm
    - wider sense : use function, functions can be values that are produces, consumed, composed.
    - function can be defined anywhere, including side other functions
    - like any other value, they can be passed as parameters to functions and returned as results
    - as for other values, there exists a set operators to compose functions

1959	1975-77	1978	1986	1990	1999	2000	2003	2005	2007
Lisp	ML, FP, Scheme	Smalltalk	Standard ML	Haskell, Erlang	XSLT	OCaml	Scala, XQuery	F#	Clojure

❑ Functional programming

```java
public class Factorial {

    public static long imperativeFactorial(int n){
        assert n > 0 : "n should be greater than 0 ";
        long result = 1;
        for(int i=2;i<=n;i++){
            result *= i;
        }
        return result;
    }

    public static long declarativeFactorial(int n){
        assert n > 0 : "n should be greater than 0 ";
        if(n==1)
            return 1;
        else
            return n * declarativeFactorial(n-1);
    }

}
```

Functional programming

- anonymous function

```
lapply(mtcars, function(x) length(unique(x)))
Filter(function(x) !is.numeric(x), mtcars)
integrate(function(x) sin(x) ^ 2, 0, pi)
```

- closures

```
power = function(exponent) {
  function(x) {
    x ^ exponent
  }
}
square <- power(2)
square(2)
```

- Mutable state

```
new_counter <- function() {
  i <- 0
  function() {
    i <<- i + 1
    i
  }
}
one = new_counter()
one()
one()
```

```
i <- 0
new_counter2 <- function() {
  i <<- i + 1
  i
}
new_counter3 <- function() {
  i <- 0
  function() {
    i <- i + 1
    i
  }
}
one2 = new_counter2()
one2();one2()
one3 = new_counter3()
one3();one3()
```

Functional programming

- Lazy evaluation & closure

```
factory = function (K) {
       function (x) print(K + x)
}
funcs<-list()
for(i in 1:5)
   funcs[[i]]<-factory({cat("evaluating K:",i,"\n"); i})
funcs[[1]](10)

factory = function (K) {
  force(K)
  function (x) print(K + x)
}
funcs<-list()
for(i in 1:5)
  funcs[[i]]<-factory({cat("evaluating K:",i,"\n"); i})
funcs[[1]](10)
```

Functional programming

- List of functions

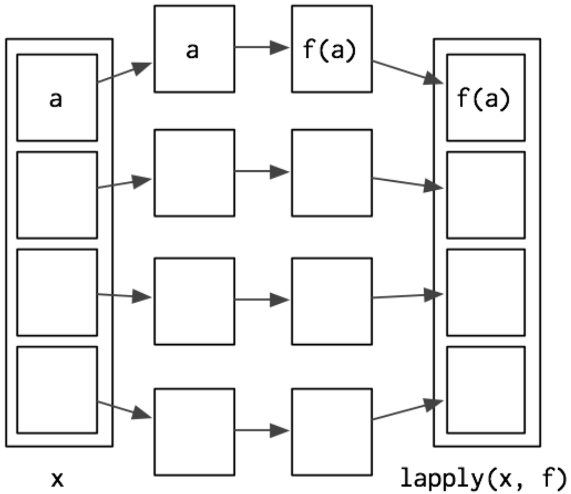
```
compute_mean <- list(
  base = function(x) mean(x),
  sum = function(x) sum(x) / length(x),
  manual = function(x) {
    total <- 0
    n <- length(x)
    for (i in seq_along(x)) {
      total <- total + x[i] / n
    }
    total
  }
)
```

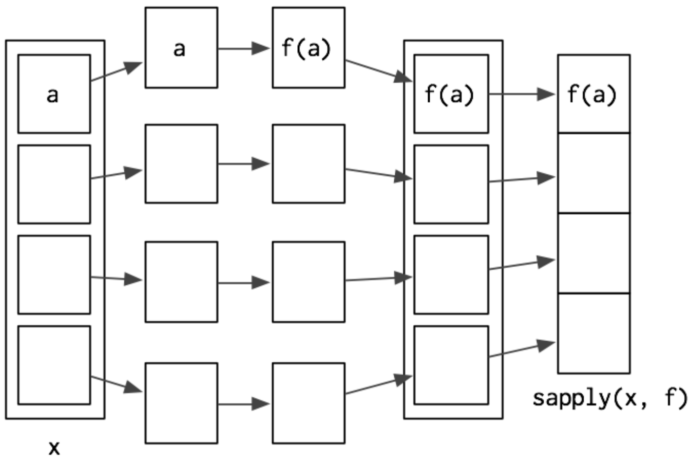
```
compute_mean$base(x)
compute_mean[[2]](x)
compute_mean[["manual"]](x)

lapply(compute_mean, function(f) f(x))
```

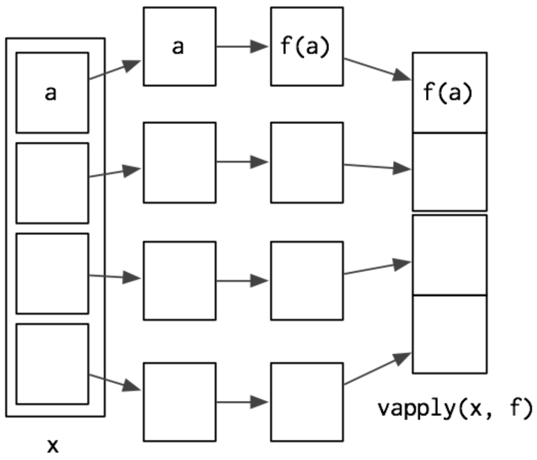
ListOfFunctions.R

❑ Functional programming

- lapply functions



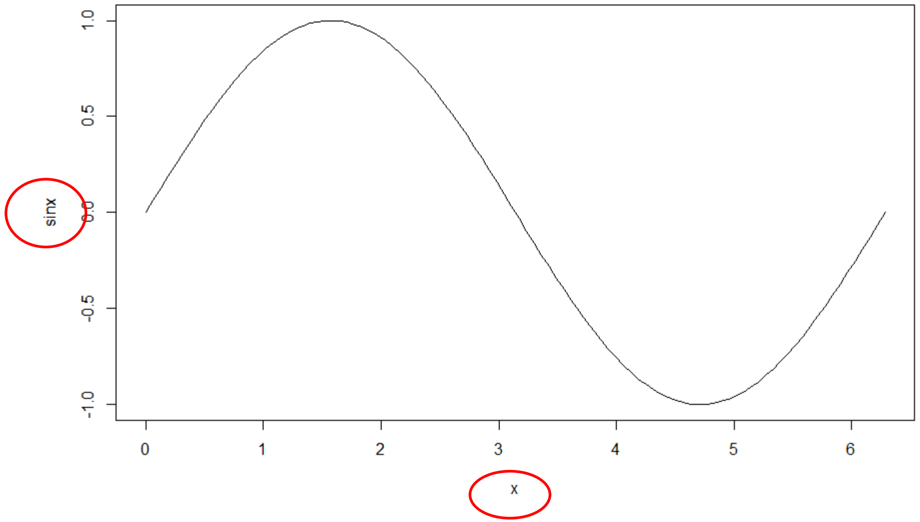




lapply.R

❑ Non-standard evaluation

```
x = seq(0, 2 * pi, length = 100)
sinx = sin(x)
plot(x, sinx, type = "l")
```



❖ Capturing expression

- substitute
- deparse : char vector
- library(ggplot2) / library(“ggplot2”)

BIGDATA ACADEMY 빅 데 이 터 아 카 데 미

Performance

- 연산 속도 측정

```
library(microbenchmark)

x <- runif(100)
microbenchmark(
  sqrt(x),
  x ^ 0.5
)
```

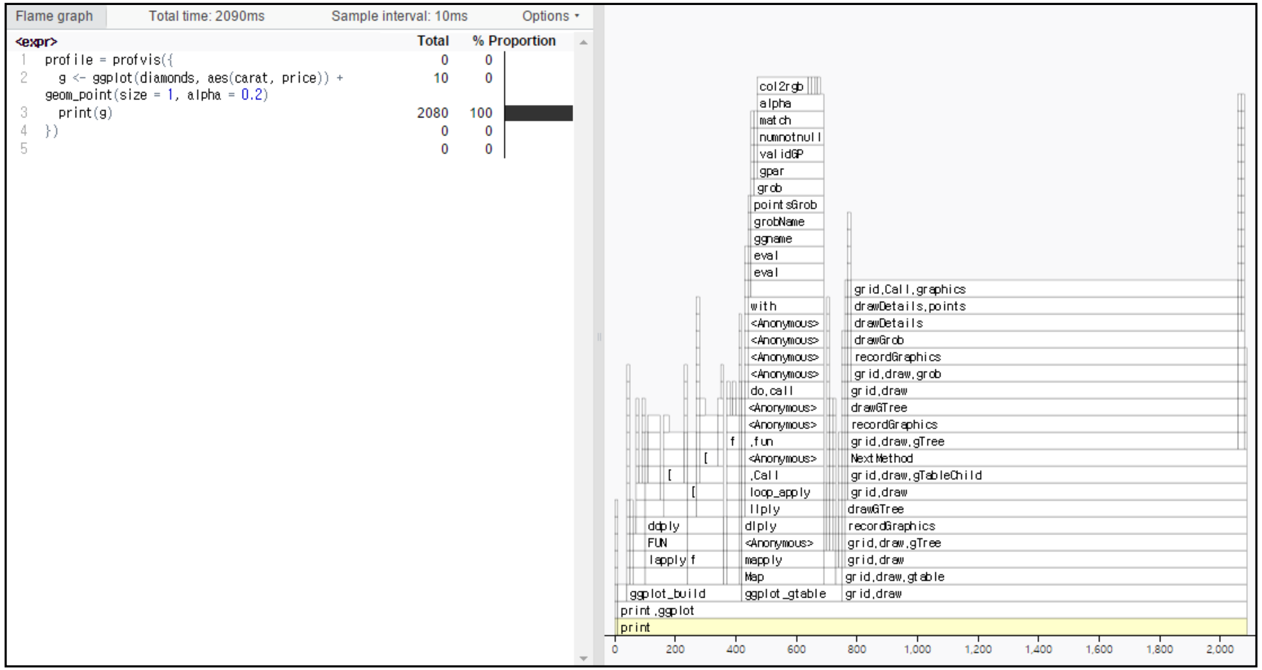
100번 수행한 시간에 대한 통계

- Lazy evaluation

```
f0 <- function() NULL
f1 <- function(a = 1) NULL
f2 <- function(a = 1, b = 1) NULL
f3 <- function(a = 1, b = 2, c = 3) NULL
f4 <- function(a = 1, b = 2, c = 4, d = 4) NULL
f5 <- function(a = 1, b = 2, c = 4, d = 4, e = 5) NULL
microbenchmark(f0(), f1(), f2(), f3(), f4(), f5(), times = 50)

Unit: nanoseconds
 expr min  lq   mean median  uq  max neval cld
 f0()   0   0  28.94    0.0   0  963    50  a
 f1()   0   0  86.76    0.0   1  962    50  a
 f2()   0   0 356.30    0.5 481 9625    50  ab
 f3()   0   0 250.36    1.0 481  963    50  ab
 f4()   0   0 298.70    1.0 482 1925    50  ab
 f5()   0 481 577.66  482.0 962 1925    50   b
```

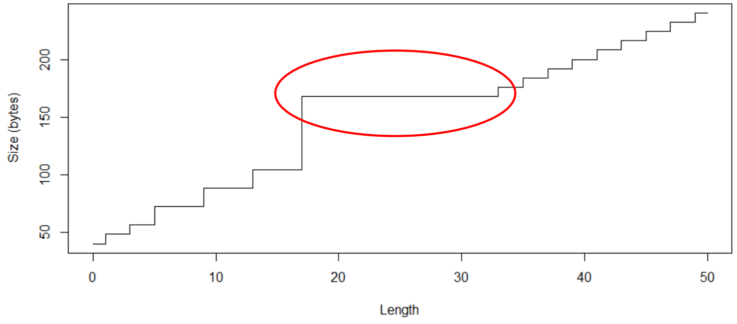
Code profiling

```
devtools::install_github("rstudio/profvis")
library(profvis)
library(ggplot2)

profile = profvis({
  g <- ggplot(diamonds, aes(carat, price)) +
geom_point(size = 1, alpha = 0.2)
  print(g)
})
profile
```



❑ Memory

```
library(pryr)
object_size(1:10)
sizes = sapply(0:50, function(n) object_size(seq_len(n)))
plot(0:50, sizes, xlab = "Length", ylab = "Size (bytes)", type = "s")
```




```
mem_used()
92.8 MB
mem_change(v <- list(1:1e8, 1:1e8, 1:1e8))
1.2 GB
mem_used()
1.29 GB
rm(v)
mem_used()
93.1 MB
```

- Memory profiling

```
devtools::install_github("hadley/lineprof")
library(lineprof)
profile = lineprof(f())
shine(profile)
```

❑ Data read

- file

```
df= read.table("http://www.ats.ucla.edu/stat/data/test.txt", header = T)
is.data.frame(df)
head(df)
?read.table

table.fixed = read.fwf("http://www.ats.ucla.edu/stat/data/test_fixed.txt", width = c(8, 1, 3, 1, 1, 1))
is.data.frame(table.fixed)
head(table.fixed)
```

- RDBMS

```
library(RJDBC)
drv = JDBC("com.mysql.jdbc.Driver",
        "/etc/jdbc/mysql-connector-java-3.1.14-bin.jar",
        identifier.quote="`")
conn = dbConnect(drv, "jdbc:mysql://localhost/test", "user", "pwd")
df = dbReadTable(conn, "iris")
df = dbGetQuery(conn,  "SELECT * FROM iris”)
```

❑ Data read

- HIVE

```
options( java.parameters = "-Xmx2g" )
library(rJava)
library(RJDBC)

cp = c("/usr/hdp/current/hive-client/lib/hive-jdbc.jar",  "/usr/hdp/current/hadoop-client/hadoop-ommon.jar")
.jinit(classpath=cp)
drv = JDBC("org.apache.hive.jdbc.HiveDriver", "/usr/hdp/current/hive-client/lib/hive-jdbc.jar",  identifier.quote="`")
conn = dbConnect(drv, "jdbc:hive2://servername:10000/demo", "user", "password")
df <- dbGetQuery(conn, "show databases")
```

- hdfs file (Spark)

```
Sys.setenv(SPARK_HOME=”/home/shige/bin/spark”)
.libPaths(c(file.path(Sys.getenv(“SPARK_HOME”), “R”, “lib”), .libPaths()))
library(SparkR)
sc = sparkR.init(master = "local[*]", sparkEnvir = list(spark.driver.memory="2g"))
sqlContext = sparkRSQL.init(sc)
df = read.df(sqlContext, “hdfs://namenode:port/xxx/yyy.parquet", "parquet")
```

Data Manipulation

❑ tidyr & dplyr package

```
library(tidyr)
library(dplyr)
```

❖ tidyr

- gather()
- spread()
- separate()
- unite()

❖ dplyr

- select()
- filter()
- group_by()
- summarise()
- arrange()
- join()
- mutate()

❖ %>% 연산자

```
a <- filter(data, variable == numeric_value)
b <- summarise(a, Total = sum(variable))
c <- arrange(b, desc(Total))
```

```
arrange(
        summarize(
           filter(data, variable == numeric_value),
           Total = sum(variable)
        ),
     desc(Total)
  )
```

```
data %>%
        filter(variable == “value”) %>%
        summarise(Total = sum(variable)) %>%
        arrange(desc(Total))
```

KODB 한국데이터베이스진흥원
BIGDATA ACADEMY 빅데이터아카데미

❑ data example

```
library(nycflights13)

str(flights)
str(weather)
str(planes)
str(airports)
```

```
str(flights)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':      336776 obs. of  16 variables:
 $ year     : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
 $ month    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ day      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ dep_time : int  517 533 542 544 554 554 555 557 557 558 ...
 $ dep_delay: num  2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
 $ arr_time : int  830 850 923 1004 812 740 913 709 838 753 ...
 $ arr_delay: num  11 20 33 -18 -25 12 19 -14 -8 8 ...
 $ carrier  : chr  "UA" "UA" "AA" "B6" ...
 $ tailnum  : chr  "N14228" "N24211" "N619AA" "N804JB" ...
 $ flight   : int  1545 1714 1141 725 461 1696 507 5708 79 301 ...
 $ origin   : chr  "EWR" "LGA" "JFK" "JFK" ...
 $ dest     : chr  "IAH" "IAH" "MIA" "BQN" ...
 $ air_time : num  227 227 160 183 116 150 158 53 140 138 ...
 $ distance : num  1400 1416 1089 1576 762 ...
 $ hour     : num  5 5 5 5 5 5 5 5 5 5 ...
 $ minute   : num  17 33 42 44 54 54 55 57 57 58 ...
```

❑ gather()

```
flight_delay = flights[c("tailnum","arr_delay", "dep_delay")]
flight_delay = flight_delay[sample(nrow(flight_delay), 5), ]
flight_delay
delay_gather = flight_delay %>% gather(delay, time, arr_delay:dep_delay)
delay_gather
```

❖ flight_delay

	tailnum	arr_delay	dep_delay
1	N434UA	-29	-4
2	N520JB	9	26
3	N538UA	19	-2
4	N934XJ	1	10
5	N744P	-12	-7

❖ delay_gather

	tailnum	delay	time
1	N434UA	arr_delay	-29
2	N520JB	arr_delay	9
3	N538UA	arr_delay	19
4	N934XJ	arr_delay	1
5	N744P	arr_delay	-12
6	N434UA	dep_delay	-4
7	N520JB	dep_delay	26
8	N538UA	dep_delay	-2
9	N934XJ	dep_delay	10
10	N744P	dep_delay	-7

❑ spread()

```
head(delay_gather,10)
flight_return <- delay_gather %>% spread(delay, time)
head(flight_return)
```

❖ delay_gather

	tailnum	delay	time
1	N434UA	arr_delay	-29
2	N520JB	arr_delay	9
3	N538UA	arr_delay	19
4	N934XJ	arr_delay	1
5	N744P	arr_delay	-12
6	N434UA	dep_delay	-4
7	N520JB	dep_delay	26
8	N538UA	dep_delay	-2
9	N934XJ	dep_delay	10
10	N744P	dep_delay	-7

❖ flight_return

	tailnum	arr_delay	dep_delay
1	N434UA	-29	-4
2	N520JB	9	26
3	N538UA	19	-2
4	N934XJ	1	10
5	N744P	-12	-7

seperate()

```
head(airport)
name_seperate <- airports %>% separate(name, c("prefix", "suffix"))
head(name_seperate)
```

❖ airport

	faa	name	lat	lon	alt	tz	dst
1	04G	Lansdowne Airport	41.13047	-80.61958	1044	-5	A
2	06A	Moton Field Municipal Airport	32.46057	-85.68003	264	-5	A
3	06C	Schaumburg Regional	41.98934	-88.10124	801	-6	A
4	06N	Randall Airport	41.43191	-74.39156	523	-5	A
5	09J	Jekyll Island Airport	31.07447	-81.42778	11	-4	A
6	0A9	Elizabethton Municipal Airport	36.37122	-82.17342	1593	-4	A

❖ name_seperate

	faa	prefix	suffix	lat	lon	alt	tz	dst
1	04G	Lansdowne	Airport	41.13047	-80.61958	1044	-5	A
2	06A	Moton	Field	32.46057	-85.68003	264	-5	A
3	06C	Schaumburg	Regional	41.98934	-88.10124	801	-6	A
4	06N	Randall	Airport	41.43191	-74.39156	523	-5	A
5	09J	Jekyll	Island	31.07447	-81.42778	11	-4	A
6	0A9	Elizabethton	Municipal	36.37122	-82.17342	1593	-4	A

❑ unite()

```
weather_part = weather[c("Date","Location", "MinTemp", "MaxTemp","Rainfall")]
head(weather_part)
temp_unite <- weather_part %>% unite(Temp, MinTemp, MaxTemp, sep = "/")
head(temp_unite)
```

❖ weather_part

	Date	Location	MinTemp	MaxTemp	Rainfall
1	2007-11-01	Canberra	8.0	24.3	0.0
2	2007-11-02	Canberra	14.0	26.9	3.6
3	2007-11-03	Canberra	13.7	23.4	3.6
4	2007-11-04	Canberra	13.3	15.5	39.8
5	2007-11-05	Canberra	7.6	16.1	2.8
6	2007-11-06	Canberra	6.2	16.9	0.0

	Date	Location	Temp	Rainfall
1	2007-11-01	Canberra	8/24.3	0.0
2	2007-11-02	Canberra	14/26.9	3.6
3	2007-11-03	Canberra	13.7/23.4	3.6
4	2007-11-04	Canberra	13.3/15.5	39.8
5	2007-11-05	Canberra	7.6/16.1	2.8
6	2007-11-06	Canberra	6.2/16.9	0.0

❖ temp_unite

KODB 한국데이터베이스진흥원
BIGDATA ACADEMY 빅데이터아카데미

❑ select()

```
head(planes)
planes_part = planes %>% select(tailnum, year, model:speed)
head(planes_part)
planes %>% select(starts_with("t"))
planes %>% select(-manufacturer, -speed)
```

❖ planes

	tailnum	year	type	manufacturer	model	engines	seats	speed	engine
1	N10156	2004	Fixed wing multi engine	EMBRAER	EMB-145XR	2	55	NA	Turbo-fan
2	N102UW	1998	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NA	Turbo-fan
3	N103US	1999	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NA	Turbo-fan
4	N104UW	1999	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NA	Turbo-fan
5	N10575	2002	Fixed wing multi engine	EMBRAER	EMB-145LR	2	55	NA	Turbo-fan
6	N105UW	1999	Fixed wing multi engine	AIRBUS INDUSTRIE	A320-214	2	182	NA	Turbo-fan

	tailnum	year	model	engines	seats	speed
1	N10156	2004	EMB-145XR	2	55	NA
2	N102UW	1998	A320-214	2	182	NA
3	N103US	1999	A320-214	2	182	NA
4	N104UW	1999	A320-214	2	182	NA
5	N10575	2002	EMB-145LR	2	55	NA
6	N105UW	1999	A320-214	2	182	NA

❖ planes_part

❑ filter()

```
summary(planes)
planes_2004 = planes %>% filter(year=='2004', engines > 2)
head(planes_2004)
```

<	Less than	!=	Not equal to
>	Greater than	%in%	Group membership
==	Equal to	is.na	is NA
<=	Less than or equal to	!is.na	is not NA
>=	Greater than or equal to	&,|,!	Boolean operators

❖ planes

```
    tailnum            year                      engines
Length:3322       Min.   :1956       ...       Min.   :1.000      ...
Class :character  1st Qu.:1997       ...       1st Qu.:2.000      ...
Mode  :character  Median :2001       ...       Median :2.000      ...
                  Mean   :2000       ...       Mean   :1.995      ...
                  3rd Qu.:2005       ...       3rd Qu.:2.000      ...
                  Max.   :2013       ...       Max.   :4.000      ...
                  NA's   :70
```

❖ planes_2004

```
  tailnum year                    type manufacturer     model engines seats speed    engine
1  N854NW 2004 Fixed wing multi engine       AIRBUS A330-223       3   379    NA Turbo-fan
2  N856NW 2004 Fixed wing multi engine       AIRBUS A330-223       3   379    NA Turbo-fan
```

❑ summarise() & group_by()

```
flights %>% summarise(dep_delay_mean=mean(dep_delay),arr_delay_mean=mean(arr_delay))
head(flights);summary(flights)
flights_complete = flights %>% filter(!is.na(dep_delay), !is.na(arr_delay))
summary(flights_complete)
flights_complete %>% summarise(dep_delay_mean=mean(dep_delay),arr_delay_mean=mean(arr_delay))
flights_groupby_summarise = flights_complete %>% group_by(month) %>%
                                    summarise(dep_delay_mean=mean(dep_delay),arr_delay_mean=mean(arr_delay))
flights_groupby_summarise
```

❖ 첫번째 summarise 결과

Source: local data frame [1 x 2]

	dep_delay_mean	arr_delay_mean
1	NA	NA

❖ 두번째 summarise 결과

Source: local data frame [1 x 2]

	dep_delay_mean	arr_delay_mean
1	12.55516	6.895377

Source: local data frame [12 x 3]

	month	dep_delay_mean	arr_delay_mean
1	1	9.985491	6.1299720
2	2	10.760239	5.6130194
3	3	13.164289	5.8075765
4	4	13.849187	11.1760630
5	5	12.891709	3.5215088
6	6	20.725614	16.4813296
7	7	21.522179	16.7113067
8	8	12.570524	6.0406524
9	9	6.630285	-4.0183636
10	10	6.233175	-0.1670627
11	11	5.420340	0.4613474
12	12	16.482161	14.8703553

❑ arrange()

```
flights_groupby_summarise_arrange = flights_groupby_summarise %>% arrange(dep_delay_mean)
flights_groupby_summarise_arrange
flights_groupby_summarise_arrange = flights_groupby_summarise %>% arrange(desc(arr_delay_mean))
flights_groupby_summarise_arrange
```

Source: local data frame [12 x 3]

	month	dep_delay_mean	arr_delay_mean
1	11	5.420340	0.4613474
2	10	6.233175	-0.1670627
3	9	6.630285	-4.0183636
4	1	9.985491	6.1299720
5	2	10.760239	5.6130194
6	8	12.570524	6.0406524
7	5	12.891709	3.5215088
8	3	13.164289	5.8075765
9	4	13.849187	11.1760630
10	12	16.482161	14.8703553
11	6	20.725614	16.4813296
12	7	21.522179	16.7113067

Source: local data frame [12 x 3]

	month	dep_delay_mean	arr_delay_mean
1	7	21.522179	16.7113067
2	6	20.725614	16.4813296
3	12	16.482161	14.8703553
4	4	13.849187	11.1760630
5	1	9.985491	6.1299720
6	8	12.570524	6.0406524
7	3	13.164289	5.8075765
8	2	10.760239	5.6130194
9	5	12.891709	3.5215088
10	11	5.420340	0.4613474
11	10	6.233175	-0.1670627
12	9	6.630285	-4.0183636

join()

```
flights_dest_group = flights %>%  group_by(dest) %>%  filter(!is.na(arr_delay)) %>%
  summarise(arr_delay = mean(arr_delay),  n = n()  ) %>%  arrange(desc(arr_delay))
location = airports %>%  select(dest = faa, name, lat, lon)
flights_join = flights_dest_group %>% left_join(location)
flights_join = flights_dest_group %>% left_join(location, by='dest')
```

```
  dest arr_delay   n
1  CAE  41.76415 106
2  TUL  33.65986 294
3  OKC  30.61905 315
4  JAC  28.09524  21
5  TYS  24.06920 578
6  MSN  20.19604 556
```

```
  dest                           name      lat       lon
1  04G              Lansdowne Airport 41.13047 -80.61958
2  06A  Moton Field Municipal Airport 32.46057 -85.68003
3  06C            Schaumburg Regional 41.98934 -88.10124
4  06N                Randall Airport 41.43191 -74.39156
5  09J          Jekyll Island Airport 31.07447 -81.42778
6  0A9 Elizabethton Municipal Airport 36.37122 -82.17342
```

```
  dest arr_delay   n                   name      lat        lon
1  CAE  41.76415 106  Columbia Metropolitan 33.93883  -81.11953
2  TUL  33.65986 294              Tulsa Intl 36.19839  -95.88811
3  OKC  30.61905 315       Will Rogers World 35.39309  -97.60073
4  JAC  28.09524  21    Jackson Hole Airport 43.60733 -110.73775
5  TYS  24.06920 578          Mc Ghee Tyson 35.81097  -83.99403
6  MSN  20.19604 556 Dane Co Rgnl Truax Fld 43.13986  -89.33751
```

❑ join()

❖ Superheroes

name	alignment	gender	publisher
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Hellboy	good	male	Dark Horse Comics

❖ Publishers

publisher	founded
DC	1934
Marvel	1939
Image	1992

❖ Superheroes %>% inner_join(Publishers, by=publisher)

name	alignment	gender	publisher	founded
Magneto	bad	male	Marvel	1939
Storm	good	female	Marvel	1939
Mystique	bad	female	Marvel	1939
Batman	good	male	DC	1934
Joker	bad	male	DC	1934
Catwoman	bad	female	DC	1934

❑ join()

❖ Superheroes %>% semi_join(Publishers, by=publisher)

name	alignment	gender	publisher
Batman	good	male	DC
Joker	bad	male	DC
Catwoman	bad	female	DC
Magneto	bad	male	Marvel
Storm	good	female	Marvel
Mystique	bad	female	Marvel

❖ Superheroes %>% left_join(Publishers, by=publisher)

name	alignment	gender	publisher	founded
Magneto	bad	male	Marvel	1939
Storm	good	female	Marvel	1939
Mystique	bad	female	Marvel	1939
Batman	good	male	DC	1934
Joker	bad	male	DC	1934
Catwoman	bad	female	DC	1934
Hellboy	good	male	Dark Horse Comics	NA

❖ Superheroes %>% anti_join(Publishers, by=publisher)

name	alignment	gender	publisher
Hellboy	good	male	Dark Horse Comics

❑ mutate()

```
flights_mutate = flights %>%  select(year, month, day, tailnum, hour, minute) %>%  mutate(time = hour + minute / 60)
flights_mutate_summarise = flights %>% mutate(time = hour + minute / 60) %>%
                                                       group_by(time) %>%  summarise(arr_delay = mean(arr_delay, na.rm = TRUE), n = n())
```

❖ flights_mutate

	year	month	day	tailnum	hour	minute	time
1	2013	1	1	N14228	5	17	5.283333
2	2013	1	1	N24211	5	33	5.550000
3	2013	1	1	N619AA	5	42	5.700000
4	2013	1	1	N804JB	5	44	5.733333
5	2013	1	1	N668DN	5	54	5.900000
6	2013	1	1	N39463	5	54	5.900000

❖ flights_mutate_summarise

	time	arr_delay	n
1	0.01666667	75.96000	25
2	0.03333333	90.00000	35
3	0.05000000	65.46154	26
4	0.06666667	60.50000	26
5	0.08333333	74.50000	21
6	0.10000000	91.90909	22

❑ data.table package

```
library(data.table)
df = copy(flights)
dt = setDT(df)
```

- large data set
- fast
- clean code

- ❖ select columns
- ❖ select rows
- ❖ group by
- ❖ add, remove fields
- ❖ join
- ❖ fread

DataTable.R

Visualization

ggplot2 package

- data
- aesthetic mapping
- geometric object
- statistical transformations
- scales
- coordinate system
- position adjustments
- faceting

http://docs.ggplot2.org/current/

- ggplot2에서 지원하지 않는 기능
  - 3 차원 그래프 : rgl package
  - 그래프 이론 형태의 그래프(node/edges layout) : igraph package
  - 대화형 그래프 : ggvis package

- ggplot2 구조

```
ggplot(data = <default data set>,
       aes(x = <default x axis variable>,
           y = <default y axis variable>,
           ... <other default aesthetic mappings>),
       ... <other plot defaults>) +

       geom_<geom type>(aes(size = <size variable for this geom>,
                   ... <other aesthetic mappings>),
             data = <data for this point geom>,
             stat = <statistic string or function>,
             position = <position string or function>,
             color = <"fixed color specification">,
             <other arguments, possibly passed to the _stat_ function) +

  scale_<aesthetic>_<type>(name = <"scale label">,
               breaks = <where to put tick marks>,
               labels = <labels for tick marks>,
               ... <other options for the scale>) +

  theme(plot.background = element_rect(fill = "gray"),
        ... <other theme elements>)
```

❑ ggplot2 package

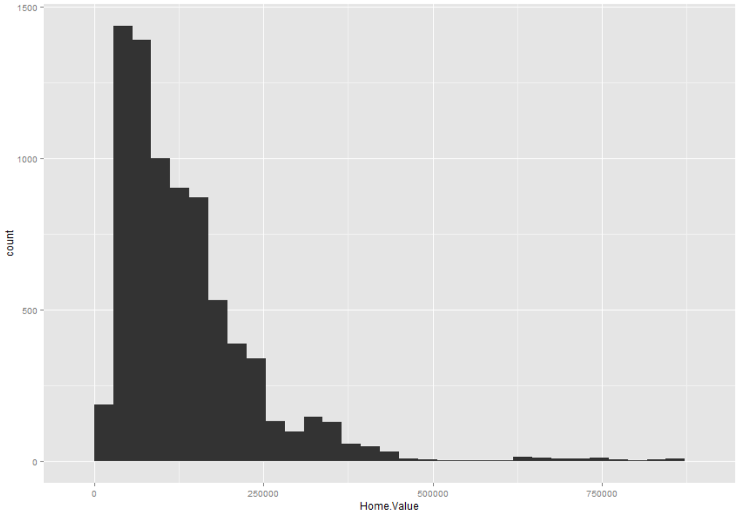
❖ sample data

```
> housing = read.csv("landdata-states.csv")
> str(housing)
'data.frame':	7803 obs. of  9 variables:
 $ State           : Factor w/ 51 levels "AK","AL","AR",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ region          : Factor w/ 4 levels "Midwest","N. East",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ Date            : int  20101 20102 20093 20094 20074 20081 20082 20083 20084 20091 ...
 $ Home.Value      : int  224952 225511 225820 224994 234590 233714 232999 232164 231039 229395 ...
 $ Structure.Cost  : int  160599 160252 163791 161787 155400 157458 160092 162704 164739 165424 ...
 $ Land.Value      : int  64352 65259 62029 63207 79190 76256 72906 69460 66299 63971 ...
 $ Land.Share..Pct.: num  28.6 28.9 27.5 28.1 33.8 32.6 31.3 29.9 28.7 27.9 ...
 $ Home.Price.Index: num  1.48 1.48 1.49 1.48 1.54 ...
 $ Land.Price.Index: num  1.55 1.58 1.49 1.52 1.88 ...
```

❖ histogram

```
> ggplot(housing, aes(x=Home.Value)) + geom_histogram()
stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

```
> ggplot(housing, aes(x=Home.Value)) + geom_histogram(bins=100)
> ggplot(housing, aes(x=Home.Value)) + geom_histogram(binwidth = 4000)
```



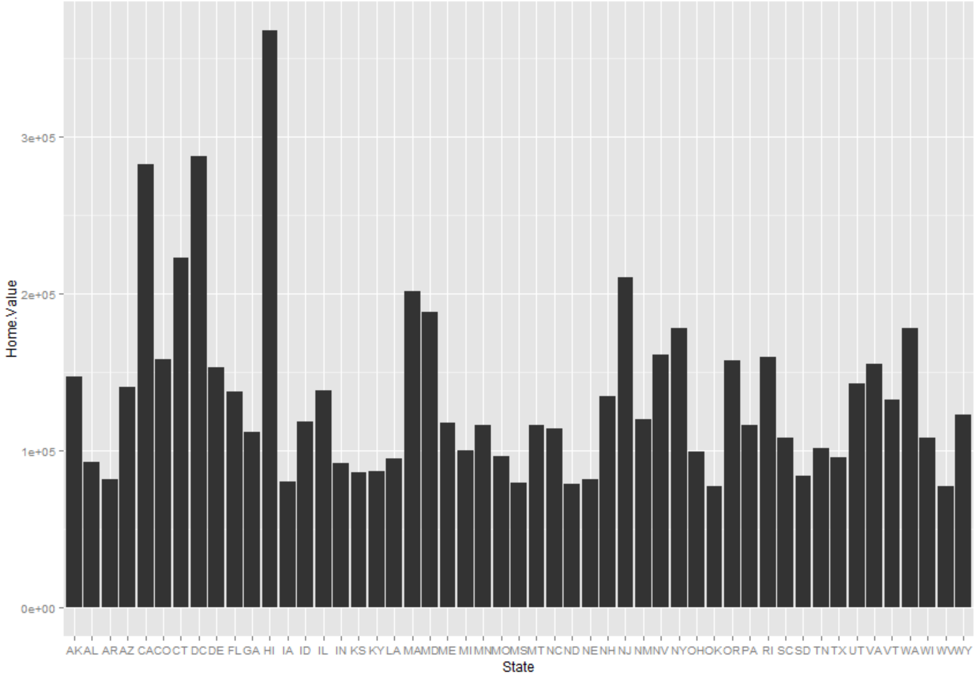
ggplot2 package

- Statistical transformation

```
housing.sum = aggregate(housing["Home.Value"], housing["State"], FUN=mean)
head(housing.sum, 10)

ggplot(housing.sum, aes(x=State, y=Home.Value)) + geom_bar()
ggplot(housing.sum, aes(x=State, y=Home.Value)) + geom_bar(stat="identity")
```

- housing.sum

	State	Home.Value
1	AK	147385.14
2	AL	92545.22
3	AR	82076.84
4	AZ	140755.59
5	CA	282808.08
6	CO	158175.99
7	CT	223063.08
8	DC	287552.56
9	DE	152905.53
10	FL	137842.59

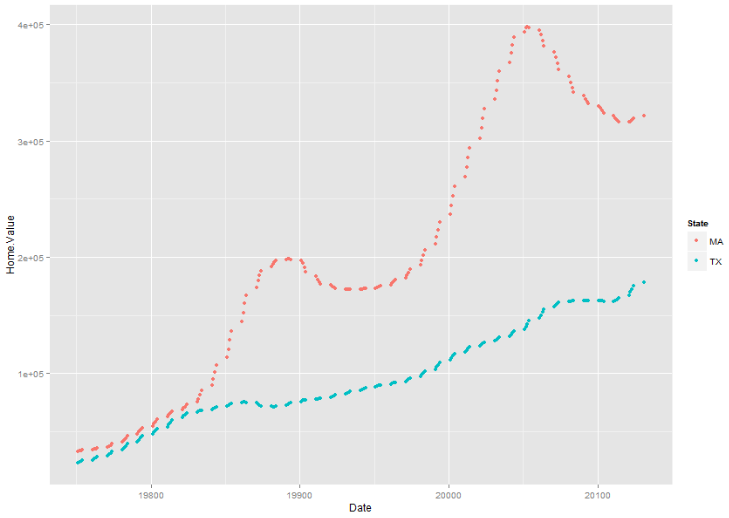


BIGDATA ACADEMY 빅데이터 아카데미

❑ ggplot2 package

❖ scatter plot

```
ggplot(subset(housing, State %in% c("MA", "TX")), aes(x=Date, y=Home.Value, color=State))+geom_point()
```



❑ ggplot2 package

❖ Aesthetics

- position(on the x, y axes)
- color(outside color)
- fill(inside color)
- shape(of point)
- linetype
- size

❖ Geometic objects

- geom_point : scatter plot, dot plot
- geom_line : time series, trend line
- geom_boxplot : box plots

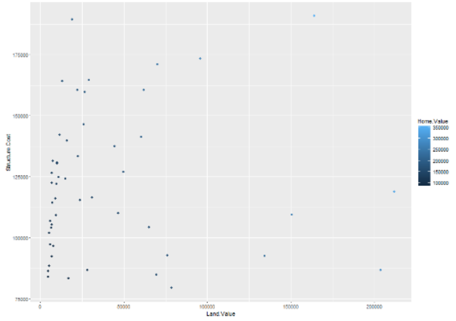
```
help.search("geom_", package = "ggplot2")
```

http://docs.ggplot2.org/current/

❑ ggplot2 package

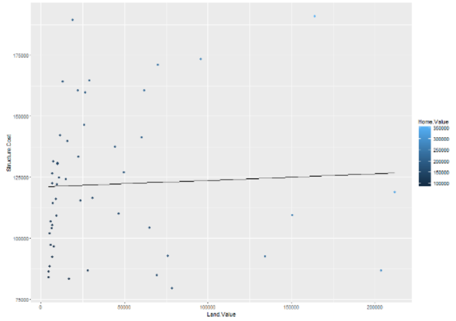
❖ Scatter plot

```
hp2001Q1 = subset(housing, Date == 20011)
p1 = ggplot(hp2001Q1, aes(y = Structure.Cost, x = Land.Value))
(p2 = p1 + geom_point(aes(color = Home.Value)))
```



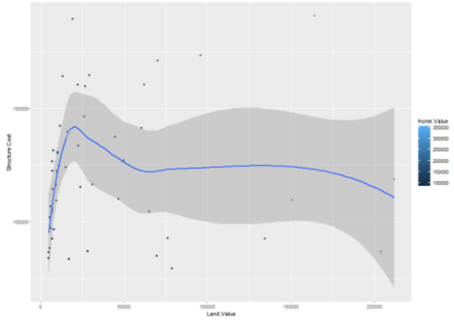
❖ Prediction line

```
hp2001Q1$pred.SC <- predict(lm(Structure.Cost ~ Land.Value, data = hp2001Q1))
(p3 = p2 + geom_line(aes(y=hp2001Q1$pred.SC)))
```



❖ Smoothers

```
(p4 = p2 + geom_smooth(method=loess))
```



❑ ggplot2 package

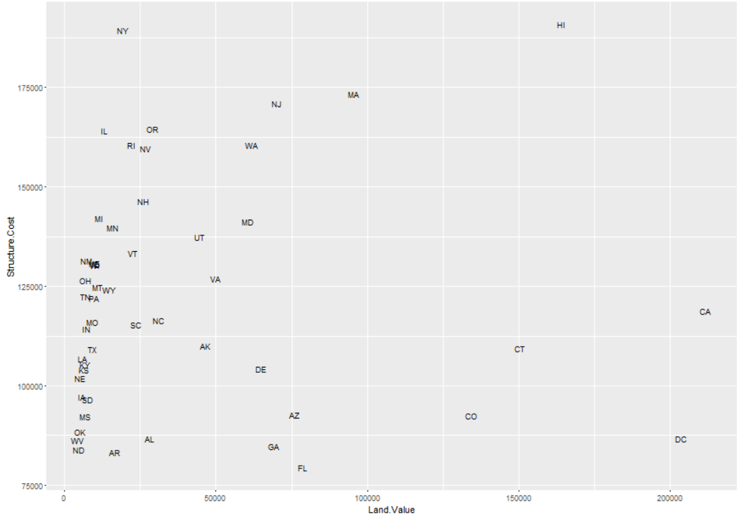
❖ Text

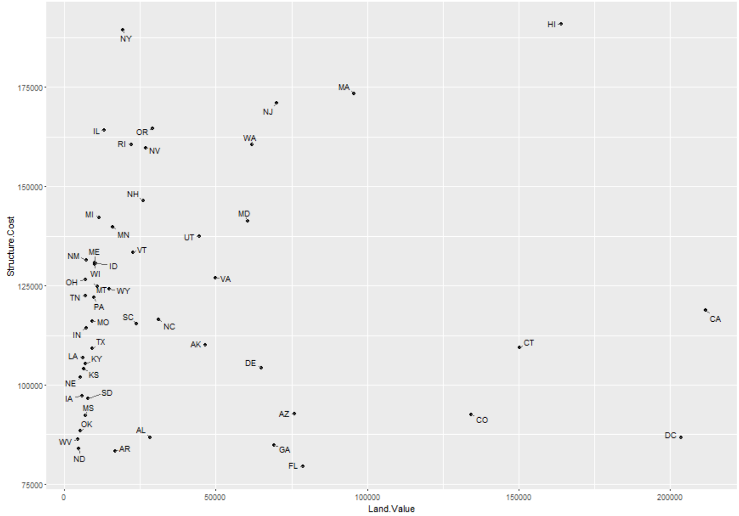
(p5 = p1 + geom_text(aes(label=State), size=3))

❖ 겹침 방지

library(ggrepel)

(p6 = p1 + geom_point() + geom_text_repel(aes(label=State), size = 3))





❑ ggplot2 package

❖ Asethetic mapping vs. Assignment

```
p1 + geom_point(aes())

p1 + geom_point(aes(size=100, color="red"))

p1 + geom_point(aes(), size=2, color="red")                # 고정 값은 ase() 밖에서 설정

p1 + geom_point(aes(color=Home.Value, shape = region))    # ase() 안에서는 field로 설정
```

<< 실습 >>

실습 데이터 : EconomistData.csv

1. x 축은 CPI, y축은 HDI로 scatter plot
2. 1번 plot 점의 색깔은 파란색으로
3. 점의 색깔을 Region 별로 다르게
4. Region에 의한 CPI boxplot
5. box plot 과 scatter plot overlay

<< 실습 >>

6. 1번 plot에 lm method를 이용하여 smoothing line 추가
7. 1번 plot에 기본 method를 이용하여 smoothing line 추가

❑ ggplot2 package

❖Scale : 데이터와 aesthetics 간의 mapping 조정

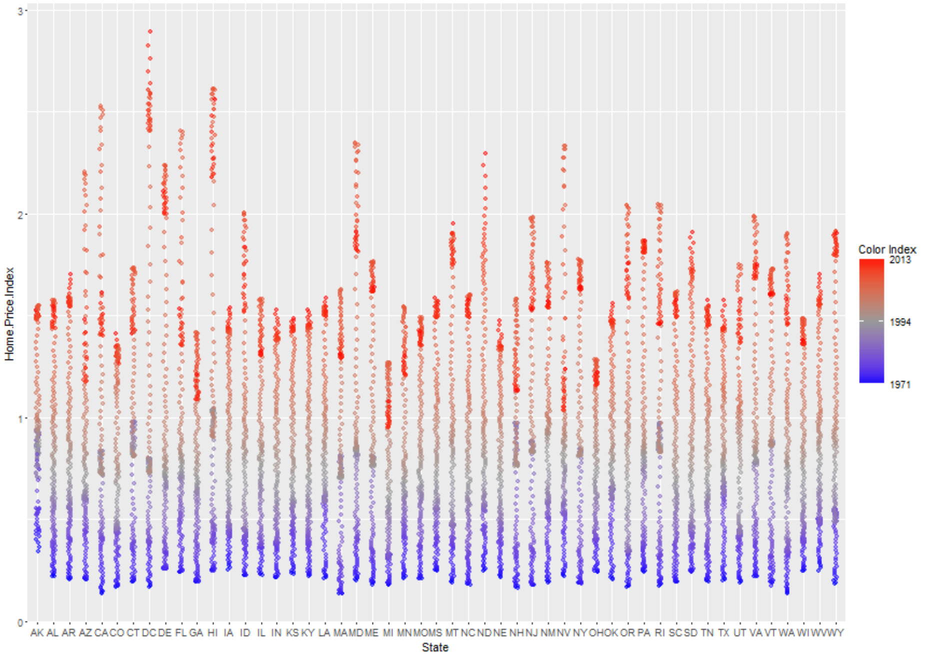
scale_<aestheitc>_<type>

position, color, fill, size, shape, line type

```
(p3 = ggplot(housing, aes(x = State, y = Home.Price.Index)))
(p4 = p3 + geom_point(aes(color=Date), alpha=0.5, size=1.5,
                      position=position_jitter(width=0.25, height=0)))

(p4 + scale_x_discrete(name="State Abbreviation") +
      scale_color_continuous(name="Color Index",
                breaks = c(19751, 19941, 20131),
                labels = c(1971, 1994, 2013),
                low = "blue", high = "red"))

(p4 + scale_color_gradient2(name="Color Index",
                breaks = c(19751, 19941, 20131),
                labels = c(1971, 1994, 2013),
                low = "blue",
                high = "red",
                mid = "gray60",
                midpoint = 19941))
```

help.search(“scale_”, package = "ggplot2")

http://docs.ggplot2.org/current/

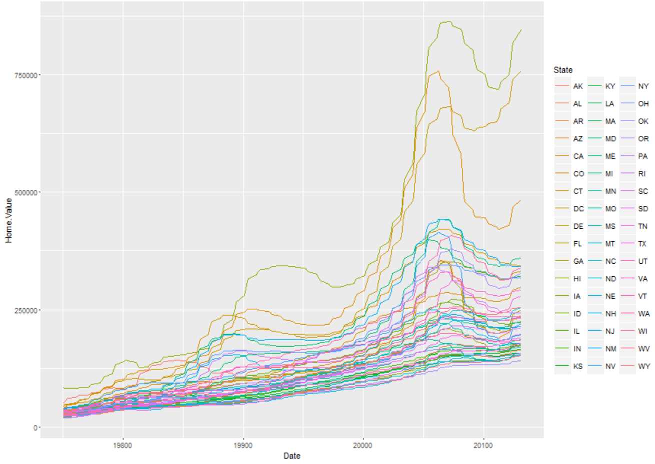


❑ ggplot2 package

❖Faceting : 데이터셋을 일부를 다른 panel에 표시

facet_wrap() : 1차원

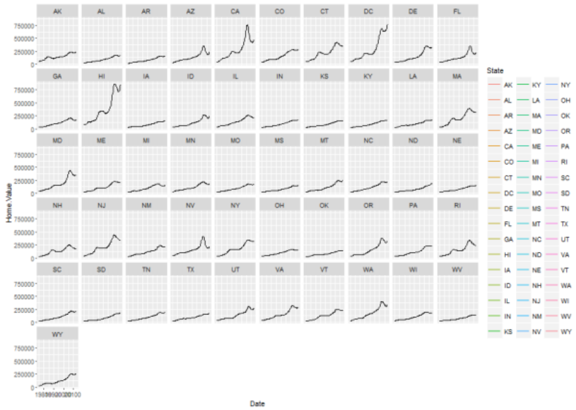
facet_grid() : 2차원

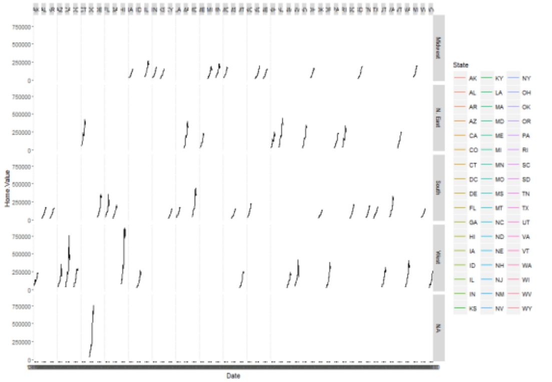


(p5 = ggplot(housing, aes(x = Date, y = Home.Value)) +
geom_line(aes(color = State)))

(p5 <- p5 + geom_line() + facet_wrap(~State, ncol = 10))

(p5 + geom_line() + facet_grid(region~State))





ggplot2 package

❖Theme : 데이터 plot 이외의 다른 요소 설정(축 레이블, 배경, 범례 등)

```
p5 + theme_linedraw()
p5 + theme_light()

# theme 재정의
p5 + theme_minimal()
p5 + theme_minimal()+ theme(text = element_text(color = "turquoise"))


theme_new = theme_bw() +
  theme(plot.background = element_rect(size = 1, color = "blue", fill = "black"),
        text=element_text(size = 12, family = "Arial", color = "ivory"),
        axis.text.y = element_text(colour = "purple"),
        axis.text.x = element_text(colour = "red"),
        panel.background = element_rect(fill = "pink"),
        strip.background = element_rect(fill = "orange"))

p5 + theme_new
```

❑ ggplot2 package

❖ 두개의 변수로 plot 그리기

```
housing.byyear = aggregate(cbind(Home.Value, Land.Value) ~ Date,
                           data = housing, mean)
head(housing.byyear)

ggplot(housing.byyear, aes(x=Date)) +
      geom_line(aes(y=Home.Value), color="red") +
      geom_line(aes(y=Land.Value), color="blue")
```

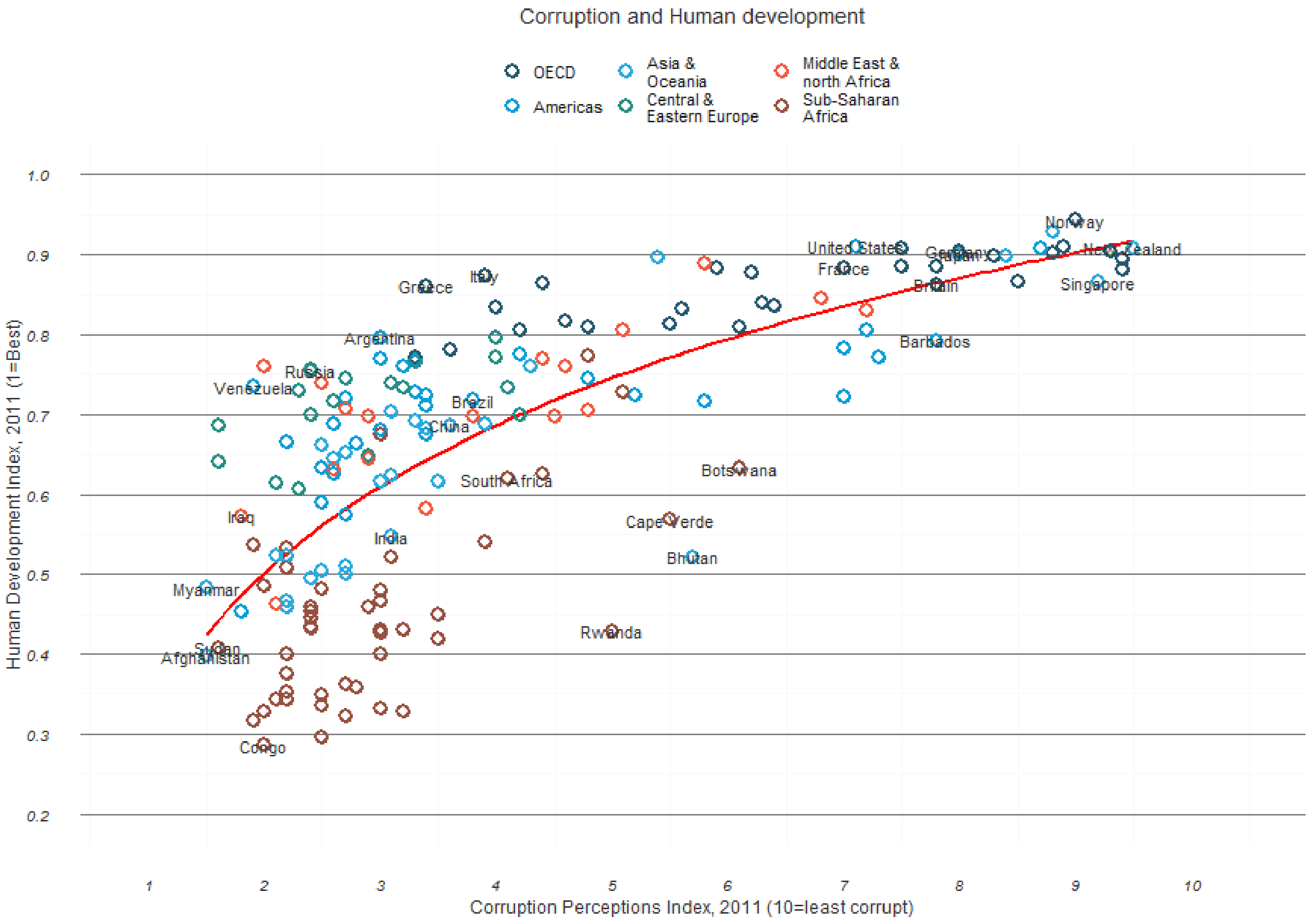
```
library(tidyr)
home.land.byyear = gather(housing.byyear, value = "value",
                          key = "type",Home.Value, Land.Value)
head(home.land.byyear)

ggplot(home.land.byyear, aes(x=Date, y=value, color=type)) +  geom_line()
```

❑ ggplot2 package

❖ 실습

- 데이터 : EconomistData.csv

1. scatter plot
2. trend line
3. open point
4. labeling
5. 겹침 방지
6. 범례 변경
7. scale 설정 : x, y, color
8. theme 설정

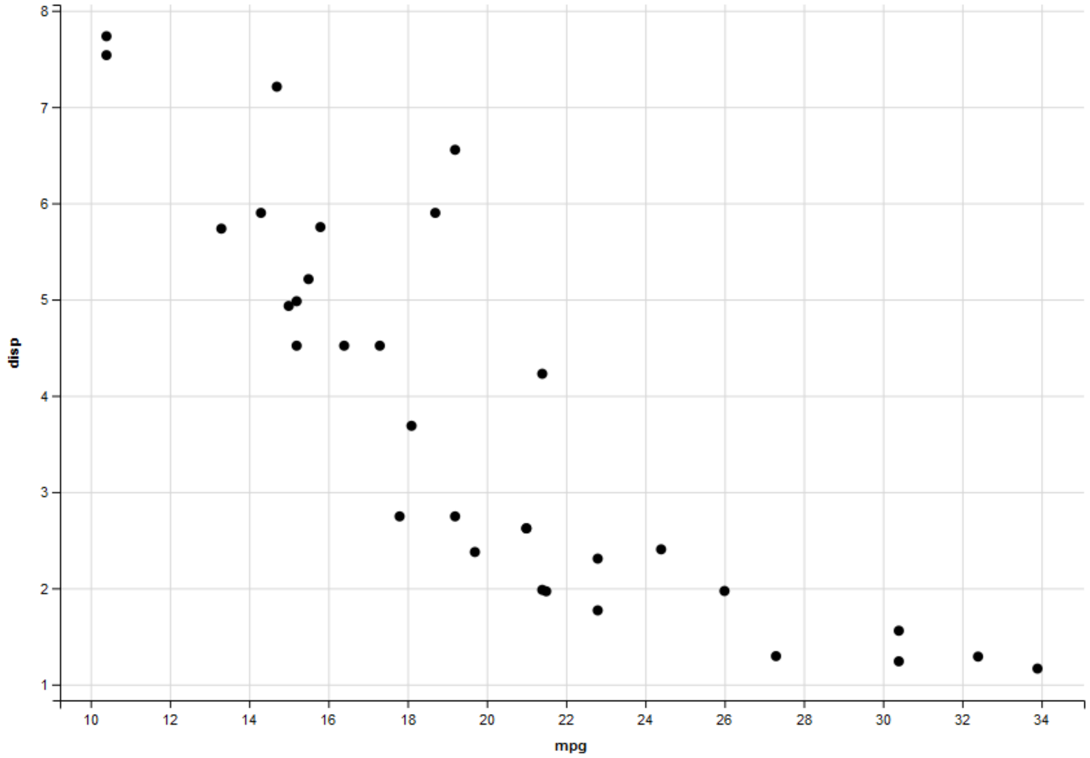


❑ ggvis package

```
library(ggvis)
library(dplyr)
library(shiny)

mtcars %>%  ggvis(x = ~mpg, y = ~disp) %>%
            mutate(disp = disp / 61.0237) %>%
            layer_points()
```

ggvis.R

https://rstudio-pubs-static.s3.amazonaws.com/1704_8f4e918c76cc447fac11113df250e02b.html



Machine Learning with R

❑ caret package

- Caret(Classification and regression training)
  - data splitting
  - pre-processing
  - feature selection
  - model tuning using resampling
  - variable importance estimation

- Machine learning algorithm
  - Linear discriminant analysis
  - Regression
  - Naïve Bayes
  - Support vector machines
  - Classification, regression trees
  - Random forests, Boosting

https://topepo.github.io/caret/modelList.html

❑ EDA(Exploratory data analysis)

```
#install.packages("caret", dependencies = c("Depends", "Suggests"))
library(caret)
library(kernlab)
data(spam)
data = spam

dim(data)
str(data)
sapply(data, class)
summary(data)

head(data,10)
levels(data$type)

percentage <- prop.table(table(data$type)) * 100
cbind(freq=table(data$type), percentage=percentage)
```

	freq	percentage
nonspam	2788	60.59552
spam	1813	39.40448

❑ EDA(Exploratory data analysis)

```
features <- data[,1:57]
target <- data[,58]

par(mfrow=c(1,4))

for(i in 1:57) {
  boxplot(features[,i], main=names(data)[i])
}

plot(target)

partFeatures = data[,1:4]
featurePlot(x=partFeatures, y=target, plot="ellipse")
featurePlot(x=partFeatures, y=target, plot="box")
scales = list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=partFeatures, y=target, plot="density", scales=scales)
```

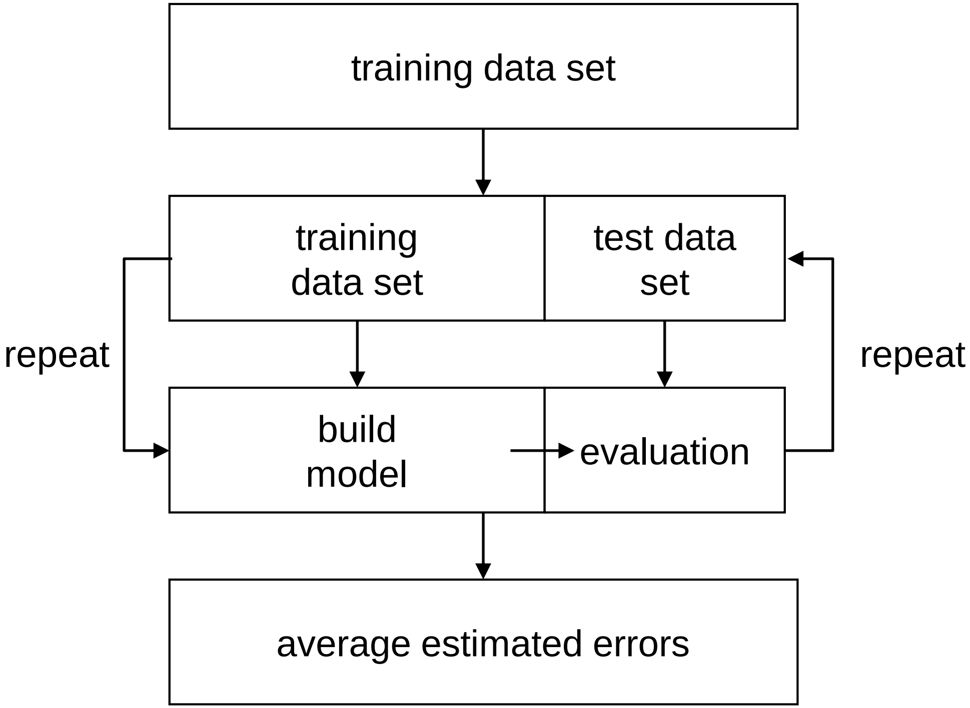
```
x <- matrix(rnorm(50*5),ncol=5)
y <- factor(rep(c("A", "B"),  25))

# classification
featurePlot(x, y, "ellipse")
featurePlot(x, y, "strip", jitter = TRUE)
featurePlot(x, y, "box")
featurePlot(x, y, "pairs")

# regression
pairs, scatter
```

❑ Data slicing

```
sampling = createDataPartition(y=data$type, p=0.75, list=F)
sampling

trainData = data[sampling,]
testData = data[-sampling,]
dim(trainData);dim(testData)

# folding
set.seed(1234)
training = createFolds(y=data$type, k=10, list=T, returnTrain=T)
set.seed(1234)
testing = createFolds(y=data$type, k=10, list=T, returnTrain=F)

sapply(training, length)
sapply(testing, length)
training[[1]][1:10]
testing[[1]][1:10]
```

```
# resampling
set.seed(1234)
resampling = createResample(y=data$type,
                            times=10, list=T)
sapply(resampling, length)
resampling[[1]][1:10]

# time slices
set.seed(1234)
tme = 1:1000
timeslicing = createTimeSlices(tme,
                  initialWindow=20, horizon=10)
names(timeslicing)
sapply(timeslicing$train, length)
sapply(timeslicing$test, length)
timeslicing$train[[1]]
timeslicing$test[[1]]
```

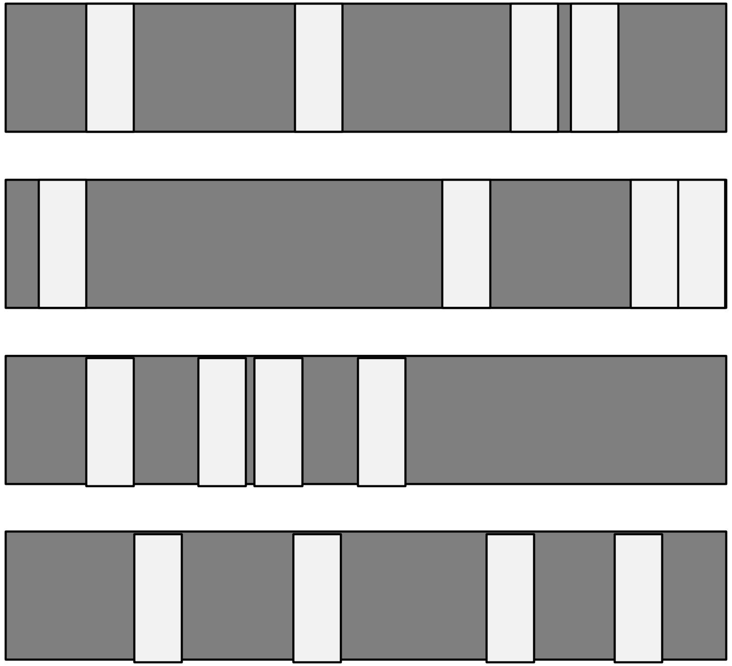
❑ Cross Validation



- 모델을 생성할 변수 선택
- 사용할 알고리즘 선택
- 알고리즘에 적용할 파라메터 선택
- 알고리즘간 비교

❑ Cross Validation

❖ Random subsampling(replace=false)



❖ bootstrap(replace=true)

❑ Cross Validation

❖ Leave one out

❖ k-fold

Testing

Training

❑ Training

```
model = train(type~., data=trainData, method="glm")
args(train.default)
```

```
function (x, y, method = "rf", preProcess = NULL, ...,
    weights = NULL,
    metric = ifelse(is.factor(y), "Accuracy", "RMSE"),
    maximize = ifelse(metric %in% c("RMSE", "logLoss"),
                              FALSE, TRUE),
    trControl = trainControl(),
    tuneGrid = NULL, tuneLength = 3)
NULL
```

```
Generalized Linear Model

3451 samples
  57 predictor
   2 classes: 'nonspam', 'spam'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 3451, 3451, 3451, 3451, 3451,
3451, ...
Resampling results

  Accuracy   Kappa      Accuracy SD  Kappa SD
  0.9199084  0.8318034  0.01199406   0.02314435
```

BIGDATA ACADEMY
빅 데 이 터 아 카 데 미

❑ Training

```
control <- trainControl(method = 'repeatedcv',
                        repeats = 5,
                        number = 5,
                        classProbs = T)
model = train(type~., data=trainData, method="glm“,
              trControl = control)
```

❖ method : resampling

- boot - bootstrapping
- boot632 - bootstrapping with adjustment
- cv - cross validation
- repeatedcv - repeated cross validation
- LOOCV - leave one out cross validation

❖ repeats

- subsampling을 반복 하는 회수
- 숫자가 커지면 수행속도 느려짐

❖ number

- boot / cross validation
- 사용할 subsampling 개수

❖ classProbs

- 결과값을 확률로 나타낼 것인가

  분류를 할 것인가

❑ preprocessing

```
ggplot(trainData, aes(x=capitalAve)) + geom_histogram()
mean(trainData$capitalAve)
sd(trainData$capitalAve)    # 값의 편차가 너무 큼

# standarization
capitalAveS = (trainData$capitalAve - mean(trainData$capitalAve)) / sd (trainData$capitalAve)
mean(capitalAveS)
sd(capitalAveS)   # 1

preObj = preProcess(trainData[,-58], method=c("center","scale"))
predict(preObj, trainData[,-58])
capitalAveS = predict(preObj, trainData[,-58])$capitalAve
mean(capitalAveS)
sd(capitalAveS)
par(mfrow=c(1,2))
hist(capitalAveS); qqnorm(capitalAveS)

model = train(type~., data=trainData, method="glm",
              trControl = control, preProcess=c("center","scale"))
```

❑ preprocessing

```
# box-cox transform
preObj = preProcess(trainData[,-58], method=c("BoxCox"))
capitalAveS = predict(preObj, trainData[,-58])$capitalAve
mean(capitalAveS)
sd(capitalAveS)
hist(capitalAveS); qqnorm(capitalAveS)

# missing value(Imputing data)
set.seed(1234)
trainData$capAve = trainData$capitalAve
summary(trainData$capAve)
selectNA = rbinom(dim(trainData)[1], size=1, prob=0.05) == 1
trainData$capAve[selectNA] = NA
summary(trainData$capAve)

preObj = preProcess(trainData[,-58], method=c("knnImpute"))
capAve = predict(preObj, trainData[,-58])$capAve
summary(capAve)
```

❑ preprocessing

method	설명	method	설명
scale	data / sd(data)	zv	zero variance
center	data - mean(data)	nzv	near zero variance
range	값을 [0,1] 사이의 값으로 scaling(normalization)	knnImpute	knn을 이용해서 NA 근처의 값들을 가중평균해서 값을 채움
Box-Cox	치우친 데이터를 정규분포화 (positive value)	bagImpute	Bagged tree model을 통해 NA 값을 예측하여 채움
YeoJohnson	치우친 데이터를 정규분포화 (0, negative 도 가능)	medianImpute	중앙값으로 NA 값을을 채움
expoTrans	지수함수를 이용한 정규분포화(positive, negative)		
pca	주성분 분석		
ica	독립성분분석, n.comp 지정해야 함		

❑ Evaluation

```
modelPredict =  predict(model, testData)
confusionMatrix(modelPredict, testData$type)
```

```
Confusion Matrix and Statistics

          Reference
Prediction nonspam spam
   nonspam     653   38
   spam         44  415

               Accuracy : 0.9287
                 95% CI : (0.9123, 0.9429)
    No Information Rate : 0.6061
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.851
 Mcnemar's Test P-Value : 0.5808

            Sensitivity : 0.9369
            Specificity : 0.9161
         Pos Pred Value : 0.9450
         Neg Pred Value : 0.9041
             Prevalence : 0.6061
         Detection Rate : 0.5678
   Detection Prevalence : 0.6009
      Balanced Accuracy : 0.9265

       'Positive' Class : nonspam
```

❑ Model evaluation matrix - Classification

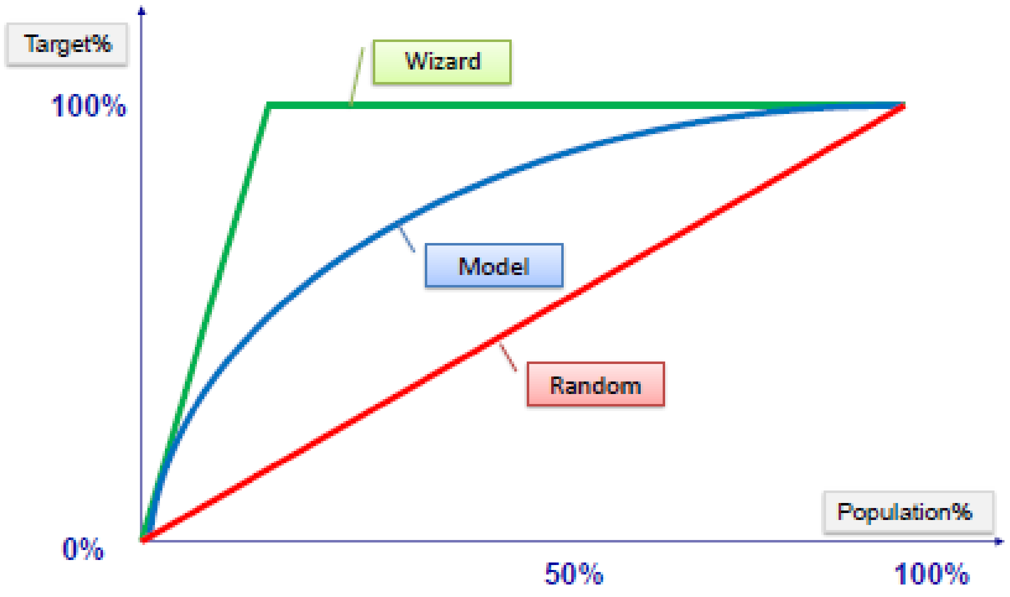
❖ Confusion matrix

<table>
<tr><td colspan="2" rowspan="2">Confusion Matrix</td><td colspan="2">Real</td><td colspan="2" rowspan="2"></td></tr>
<tr><td>Positive</td><td>Negative</td></tr>
<tr><td rowspan="2">Predict</td><td>Positive</td><td>a</td><td>b</td><td>Positive Predictive Value
(precision)</td><td>a/(a+b)</td></tr>
<tr><td>Negative</td><td>c</td><td>d</td><td>Negative Predictive Value</td><td>d/(c+d)</td></tr>
<tr><td colspan="2" rowspan="2"></td><td>Sensitivity
(recall)</td><td>Specificity</td><td colspan="2" rowspan="2">Accuracy
(a+d) / (a+b+c+d)</td></tr>
<tr><td>a/(a+c)</td><td>d/(b+d)</td></tr>
</table>

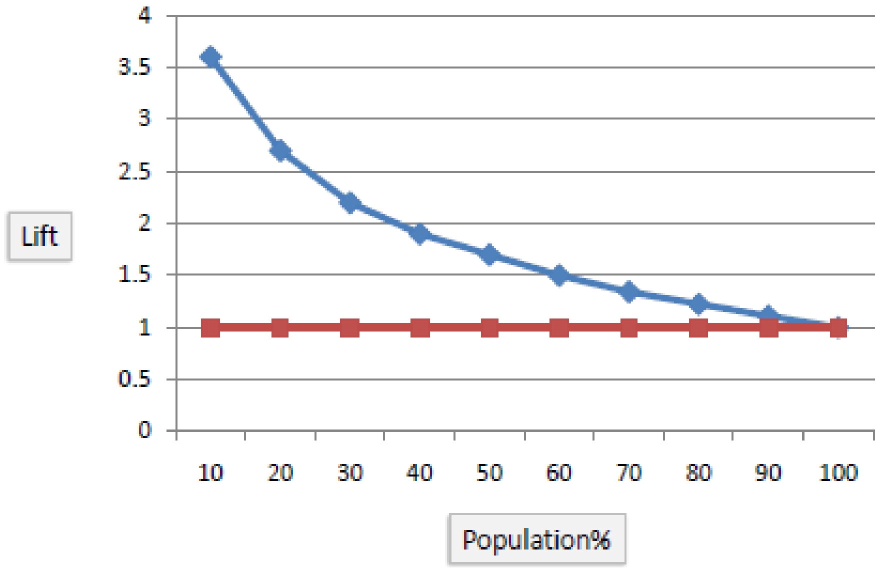
- F = 2 * (precision * recall) / (precision + recall)

❑ Model evaluation matrix - Classification

❖ Gain Chart

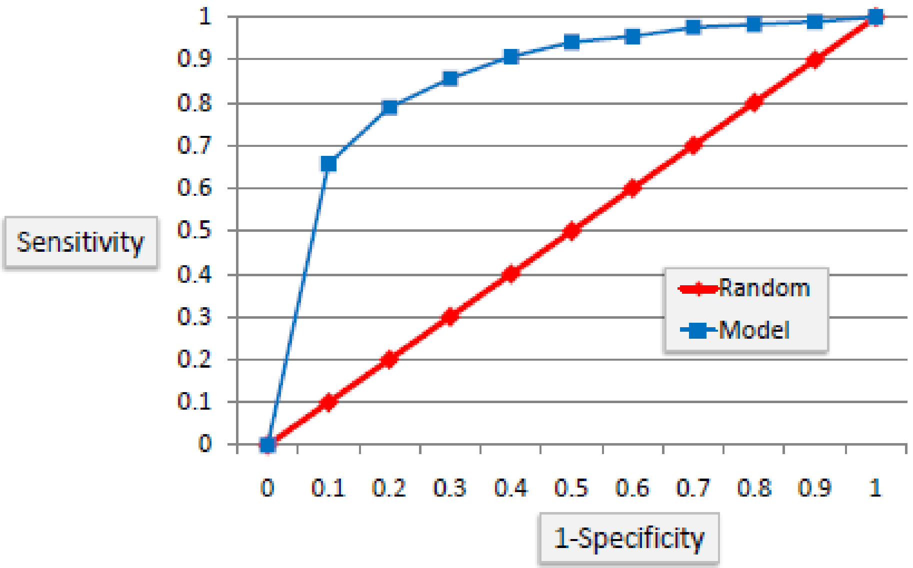


❖ Lift Chart

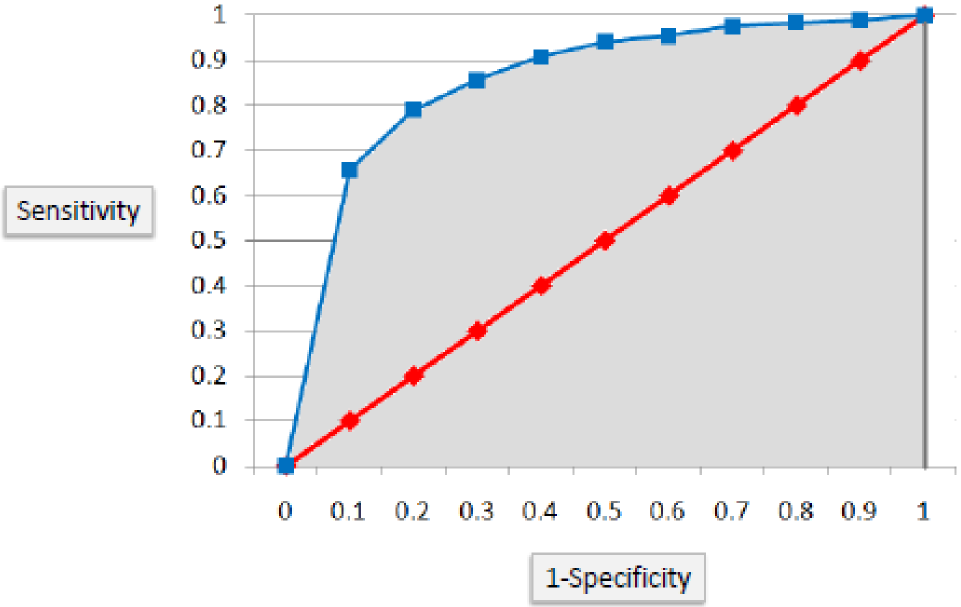


❑ Model evaluation matrix - Classification

❖ ROC Curve



❖ Area under ROC Curve



Model evaluation matrix - Regression

- RMSE - Root Mean Squared Error

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}}$$

- RSE - Relative Squared Error

$$RSE = \frac{\sum_{i=1}^{n}(p_i - a_i)^2}{\sum_{i=1}^{n}(\bar{a} - a_i)^2}$$

- MAE - Mean Absolute Error

$$MAE = \frac{\sum_{i=1}^{n}|p_i - a_i|}{n}$$

- RAE - Relative Absoulte Error

$$RAE = \frac{\sum_{i=1}^{n}|p_i - a_i|}{\sum_{i=1}^{n}|\bar{a} - a_i|}$$

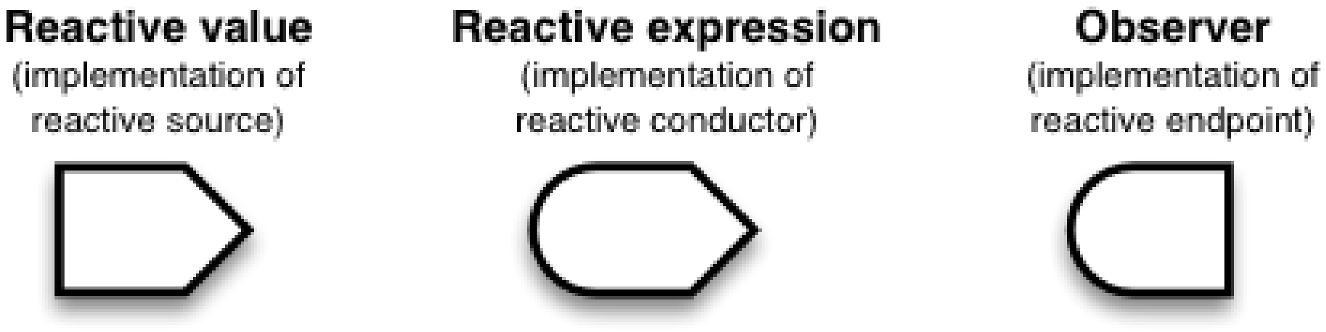
a : real value
p : predict value
ā : average

R Shiny

❑ Reactive programming

❖ Reactive programming : Asynchronous and Event based data streaming을 처리하는 Message driven architecture

Responsive	Scalable	Event-Driven	Resillient



Reactive source **Reactive conductors** **Reactive endpoints**

❑ Shiny

```
library(shiny)

runExample("01_hello") # a histogram
runExample("02_text") # tables and data frames
runExample("03_reactivity") # a reactive expression
runExample("04_mpg") # global variables
runExample("05_sliders") # slider bars
runExample("06_tabsets") # tabbed panels
runExample("07_widgets") # help text and submit buttons
runExample("08_html") # Shiny app built from HTML
runExample("09_upload") # file upload wizard
runExample("10_download") # file download wizard
runExample("11_timer") # an automated timer

# http://shiny.rstudio.com/gallery/
```

❑ 소스 구조

```
FirstShinyApp
  ├ ui.R
  └ server.R
```

ui.R

```r
library(shiny)

shinyUI(fluidPage(
  titlePanel("Shiny Text"),
  sidebarLayout(
    sidebarPanel(
      selectInput("dataset", "Choose a dataset:",
                  choices = c("rock", "pressure", "cars")),

      numericInput("obs", "Number of observations to
view:", 10)
    ),

mainPanel(
      verbatimTextOutput("summary"),

      tableOutput("view")
    )
  )
))
```

server.R

```r
library(shiny)
library(datasets)

shinyServer(function(input, output) {

  datasetInput <- reactive({
    switch(input$dataset,
           "rock" = rock,
           "pressure" = pressure,
           "cars" = cars)
  })

  output$summary <- renderPrint({
    dataset <- datasetInput()
    summary(dataset)
  })

  output$view <- renderTable({
    head(datasetInput(), n = input$obs)
  })
})
```

❑ 소스 구조

FirstShinyApp.R

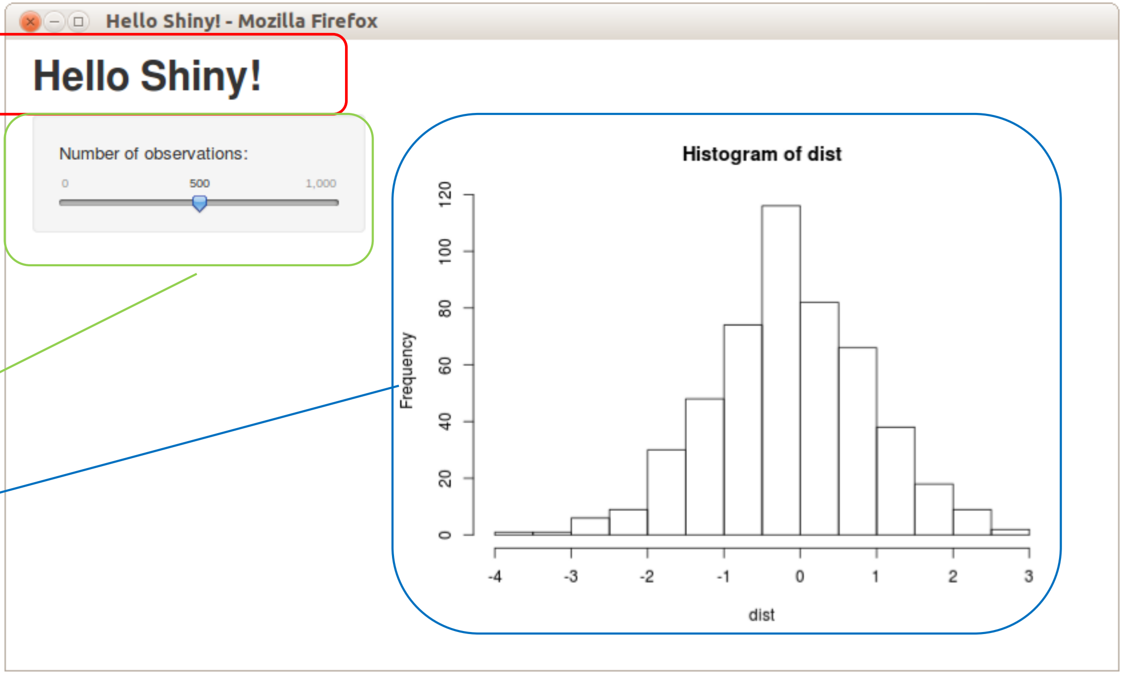
```
library(shiny)

runApp(
    list(
        ui = fluidPage(
            ……
            ……
        ),
        server = function(input, output) {
            ……
            ……
        })
    )
)
```

```
runApp("first")

runApp("first", display.mode = "showcase")
```

UI Layout

- simple layout

```
library(shiny)

shinyUI(fluidPage(

  titlePanel("Hello Shiny!"),

  sidebarLayout(

   sidebarPanel(
     sliderInput("obs", "Number of observations:",
                 min = 1, max = 1000, value = 500)
   ),

   mainPanel(
     plotOutput("distPlot")
   )
  )
))
```



BIGDATA ACADEMY 빅데이터 아카데미

UI Layout

- Grid layout
  - 12 column 으로 구성

```
shinyUI(fluidPage(

 titlePanel("Diamonds Explorer!"),

 fluidRow(
  column(3,
        h4("Diamonds Explorer"),
        sliderInput('sampleSize', 'Sample Size',
                min=1, max=nrow(dataset), value=min(1000, nrow(dataset)),
                step=500, round=0),
        br(),
        checkboxInput('jitter', 'Jitter'),
        checkboxInput('smooth', 'Smooth')
  ),
  column(4, offset = 1,
        selectInput('x', 'X', names(dataset)),
        selectInput('y', 'Y', names(dataset), names(dataset)[[2]]),
        selectInput('color', 'Color', c('None', names(dataset)))
  ),
  column(4,
        selectInput('facet_row', 'Facet Row', c(None='.', names(dataset))),
        selectInput('facet_col', 'Facet Column', c(None='.', names(dataset)))
  )
 )
))
```

UI Layout

- Segment layout
  - tabsetPanel()
  - navlistPanel()

tabsetPanel(type = "tabs", position="below",

// "above", "below", "left", "right"

```
shinyUI(fluidPage(

  titlePanel("Tabsets"),

  sidebarLayout(
    sidebarPanel(
      radioButtons("dist", "Distribution type:",
                   c("Normal" = "norm",          "Uniform" = "unif",
                     "Log-normal" = "lnorm",      "Exponential" = "exp")),
      br(),

      sliderInput("n",
                  "Number of observations:",
                  value = 500,              min = 1,              max = 1000)
    ),

    mainPanel(
      tabsetPanel(type = "tabs",
                  tabPanel("Plot", plotOutput("plot")),
                  tabPanel("Summary", verbatimTextOutput("summary")),
                  tabPanel("Table", tableOutput("table"))
      )
    )
  )
))
```

KODB 한국데이터베이스진흥원

❑ UI Layout

❖ Segment layout

- tabsetPanel()
- navlistPanel()

```
shinyUI(fluidPage(

  titlePanel("Application Title"),

  navlistPanel(
    "Header A",
    tabPanel("Component 1", "Component 1"),
    tabPanel("Component 2", "Component 2"),
    "Header B",
    tabPanel("Component 3", "Component 3"),
    tabPanel("Component 4", "Component 5"),
    "-----",
    tabPanel("Component 5", "Component 5")
  )
))

# Navbar pages
shinyUI(navbarPage("My Application", header="header", footer="footer",
                   tabPanel("Component 1", "Component 1"),
                   tabPanel("Component 2", "Component 2"),
                   navbarMenu("Component 3",
                              tabPanel("Sub-Component A", "Sub-Component A"),
                              tabPanel("Sub-Component B", "Sub-Component B"))
))
```

❑ HTML contents

method	html tag
p	<p>
h1	<h1>
h2	<h2>
h3	<h3>
h4	<h4>
h5	<h5>
h6	<h6>
a	<a>
br	
div	<div>
span	<span>
pre	<pre>

method	html tag
code	<code>
img	<img>
strong	<strong>
em	<em>
HTML	

```
shinyUI(fluidPage(
  titlePanel("My Shiny App"),
  sidebarLayout(
    sidebarPanel(),
    mainPanel(
      h1("First level title"),
      HTML("<br>"),
      h2("Second level title"),
      HTML("<br>"),
      h3("Third level title")
    )
  )
))
```

❑ Control widget

runApp(“widget")

function	widget
actionButton	Action Button
checkboxGroupInput	A group of check boxes
checkboxInput	A single check box
dateInput	A calendar to aid date selection
dateRangeInput	A pair of calendars for selecting a date range
fileInput	A file upload control wizard
helpText	Help text that can be added to an input form
numericInput	A field to enter numbers
radioButtons	A set of radio buttons
selectInput	A box with choices to select from
sliderInput	A slider bar
submitButton	A submit button
textInput	A field to enter text

❑ Control widget

❖ action widget

- actionButton("<inputId>", "<label>")
- actionLink("<inputId>", "<label>")

▪ Command 실행

```
library(shiny)

ui <- fluidPage(
  tags$head(tags$script(
    "Shiny.addCustomMessageHandler('testmessage',
    function(message) {
    alert(JSON.stringify(message));
    }
  );")),
  actionButton("do", "Click Me")
  )

server <- function(input, output, session) {
  observeEvent(input$do, {
    session$sendCustomMessage(type = 'testmessage',
                              message = 'Thank you for clicking')
  })
}

shinyApp(ui, server)
```

❑ Control widget

- Reactive data

```
library(shiny)
ui <- fluidPage(
  actionButton("go", "Go"),
  numericInput("n", "n", 50),
  plotOutput("plot")
)
server <- function(input, output) {

  randomVals <- eventReactive(input$go, {
    runif(input$n)
  })

  output$plot <- renderPlot({
    hist(randomVals())
  })
}
shinyApp(ui, server)
```

- 여러 개의 actionButton 사용

```
library(shiny)
ui <- fluidPage(
  actionButton("runif", "Uniform"),
  actionButton("rnorm", "Normal"),
  hr(),
  plotOutput("plot")
)
server <- function(input, output){
  v <- reactiveValues(data = NULL)
  observeEvent(input$runif, {
    v$data <- runif(100)
  })
  observeEvent(input$rnorm, {
    v$data <- rnorm(100)
  })
  output$plot <- renderPlot({
    if (is.null(v$data)) return()
    hist(v$data)
  })
}
shinyApp(ui, server)
```

❑ Reactive Data Stream

output function	create	renderer
htmlOutput	raw HTML	renderPrint
imageOutput	image	renderImage
plotOutput	plot	renderPlot
tableOutput	table	renderTable
textOutput	text	renderText
uiOutput	raw HTML	renderUI
verbatimTextOutput	text	renderPrint

```
shinyServer(
  function(input, output) {
    output$text1 <- renderText({
      paste("You have selected", input$var)
    })
  }
)
```

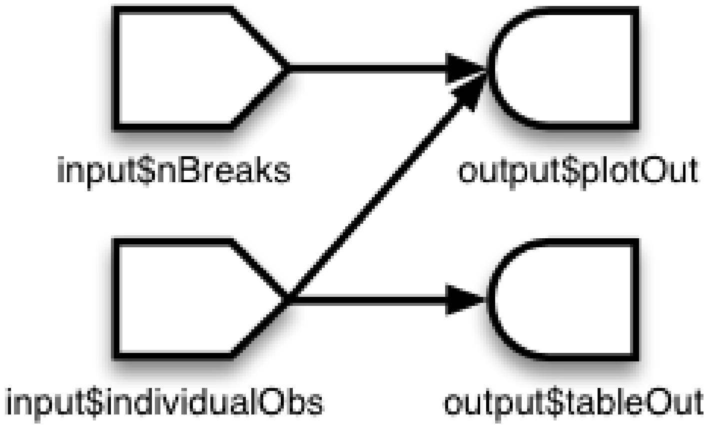
```
shinyUI(fluidPage(
  titlePanel("censusVis"),

  sidebarLayout(
    sidebarPanel(
      selectInput("var",
      label = "Choose a variable to display",
      choices = c("Percent White",
                  "Percent Black",
                  "Percent Hispanic",
                  "Percent Asian"),
      selected = "Percent White")
  ),

    mainPanel(
      textOutput("text1")
    )
  )
))
```

❑ Reactive Data Stream

```
library(shiny)
runApp(
  list(
    ui = pageWithSidebar(
      headerPanel("Display Text!!"),
      sidebarPanel(
        p("display text...")
      ),
      mainPanel(
        verbatimTextOutput("text1"),
        verbatimTextOutput("text3"),
        htmlOutput("text2")
      )
    ),
    server = function(input, output){

      output$text1 <- renderText({
        "hello world!!"
      })

output$text2 <- renderUI({
        "hello world!!"
      })

      c = list("a", "b", "c")
      output$text3 <- renderPrint({
        c
      })

    }
  )
)
```

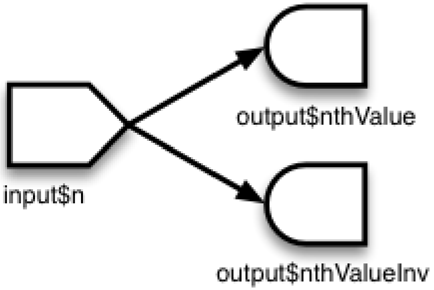
❑ Reactive Data Stream

```
shinyServer(function(input, output) {
  output$plotOut <- renderPlot({
    hist(faithful$eruptions, breaks = as.numeric(input$nBreaks))
    if (input$individualObs)
      rug(faithful$eruptions)
  })

  output$tableOut <- renderTable({
    if (input$individualObs)
      faithful
    else
      NULL
  })
})
```

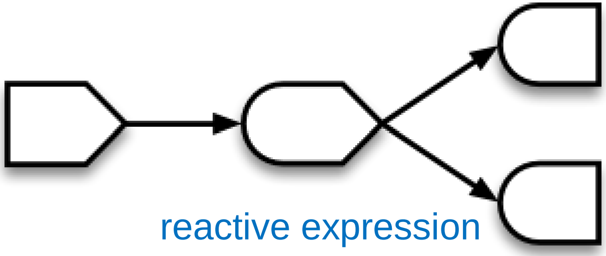


❑ Reactive Data Stream

```
fib <- function(n) ifelse(n<3, 1, fib(n-1)+fib(n-2))

shinyServer(function(input, output) {
  output$nthValue    <- renderText({ fib(as.numeric(input$n)) })
  output$nthValueInv <- renderText({ 1 / fib(as.numeric(input$n)) })
})
```




```
fib <- function(n) ifelse(n<3, 1, fib(n-1)+fib(n-2))

shinyServer(function(input, output) {
  currentFib         <- reactive({ fib(as.numeric(input$n)) })

  output$nthValue    <- renderText({ currentFib() })
  output$nthValueInv <- renderText({ 1 / currentFib() })
})
```



reactive expression
(reactive conductor)

❑ Reactive Data Stream

❖ isolation

```
library(shiny)

ui <- pageWithSidebar(
  headerPanel("Click the button"),
  sidebarPanel(
    sliderInput("obs", "Number of observations:",
                min = 0, max = 1000, value = 500),
    actionButton("run","run")
  ),
  mainPanel(
    plotOutput("distPlot")
  )
)

server <- function(input, output) {
  output$distPlot <- renderPlot({
    input$run
    dist <- isolate(rnorm(input$obs))
    hist(dist)
  })
}

shinyApp(ui, server)
```

❑ HTML UI

```
names(tags)
  [1] "a"          "abbr"       "address"    "area"       "article"    "aside"      "audio"
  [8] "b"          "base"       "bdi"        "bdo"        "blockquote" "body"       "br"
 [15] "button"     "canvas"     "caption"    "cite"       "code"       "col"        "colgroup"
 [22] "command"    "data"       "datalist"   "dd"         "del"        "details"    "dfn"
 [29] "div"        "dl"         "dt"         "em"         "embed"      "eventsource" "fieldset"
 [36] "figcaption" "figure"     "footer"     "form"       "h1"         "h2"         "h3"
 [43] "h4"         "h5"         "h6"         "head"       "header"     "hgroup"     "hr"
 [50] "html"       "i"          "iframe"     "img"        "input"      "ins"        "kbd"
 [57] "keygen"     "label"      "legend"     "li"         "link"       "mark"       "map"
 [64] "menu"       "meta"       "meter"      "nav"        "noscript"   "object"     "ol"
 [71] "optgroup"   "option"     "output"     "p"          "param"      "pre"        "progress"
 [78] "q"          "ruby"       "rp"         "rt"         "s"          "samp"       "script"
 [85] "section"    "select"     "small"      "source"     "span"       "strong"     "style"
 [92] "sub"        "summary"    "sup"        "table"      "tbody"      "td"         "textarea"
 [99] "tfoot"      "th"         "thead"      "time"       "title"      "tr"         "track"
[106] "u"          "ul"         "var"        "video"      "wbr"
```

❑ HTML UI

```
library(shiny)
shinyUI(fluidPage(
  titlePanel("Hello Shiny!"),
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
                  "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30),
      tags$div(class="header", checked=NA,
               tags$p("Ready to take the Shiny tutorial? If so"),
               tags$a(href="http://shiny.rstudio.com/tutorial", "Click Here!")
      )
    ),
    mainPanel(
      plotOutput("distPlot")
    )
  )
))
```

❑ HTML UI

runExample("06_tabsets")

```
<application-dir>
   www
        index.html
   server.R
```

```html
<html>
  <head>
  <script src="shared/jquery.js" type="text/javascript"></script>
  <script src="shared/shiny.js" type="text/javascript"></script>
  <link rel="stylesheet" type="text/css" href="shared/shiny.css"/>
  </head>
  <body>
  <h1>HTML UI</h1>
  <p>
  <label>Distribution type:</label><br />
  <select name="dist">
  <option value="norm">Normal</option> <option value="unif">Uniform</option>
  <option value="lnorm">Log-normal</option> <option value="exp">Exponential</option>
  </select>
  </p>
  <p>
  <label>Number of observations:</label><br />
  <input type="number" name="n" value="500" min="1" max="1000" />
  </p>
  <pre id="summary" class="shiny-text-output"></pre>
  <div id="plot" class="shiny-plot-output"  style="width: 100%; height: 400px"></div>
  <div id="table" class="shiny-html-output"></div>
  </body>
  </html>
```

❑ Execution Flow

```r
library(shiny)

shinyUI(fluidPage(
  titlePanel("Reactive"),

  sidebarLayout(
    sidebarPanel(
      textInput("name", "Your Name")
    ),

    mainPanel(
      textOutput("text1")
    )
  )
))
```

```r
library(shiny)

print("Outside!")

shinyServer(function(input, output) {

  print("Inside!")
  output$text1 <- renderText({
    print("Inside render!")
    paste0("Hello ", input$name)
  })

})
```

❑ Session

```r
shinyServer(function(input, output, session) {

  # Return the components of the URL in a string:
  output$urlText <- renderText({
    paste(sep = "",
          "protocol: ", session$clientData$url_protocol, "\n",
          "hostname: ", session$clientData$url_hostname, "\n",
          "pathname: ", session$clientData$url_pathname, "\n",
          "port: ",     session$clientData$url_port,     "\n",
          "search: ",   session$clientData$url_search,   "\n"
    )
  })

  # Parse the GET query string
  output$queryText <- renderText({
    query <- parseQueryString(session$clientData$url_search)

    # Return a string with key-value pairs
    paste(names(query), query, sep = "=", collapse=", ")
  })
})
```

```r
shinyUI(bootstrapPage(
  h3("URL components"),
  verbatimTextOutput("urlText"),

  h3("Parsed query string"),
  verbatimTextOutput("queryText")
))
```

Run

- R Studio
  - runApp("<directory name>")
  - runUrl("<url>")
  - runGitHub("<your repository name>", "<your user name>")
  - runGist("<gist number>")

- command line
  - R -e "shiny::runApp('<path>')"

❑ Run

❖ Hosting

- http://www.shinyapp.io

```
install.packages('devtools')
devtools::install_github('rstudio/rsconnect')
rsconnect::setAccountInfo(name='raonbit', token=‘~~', secret=‘~~')
library(rsconnect)
rsconnect::deployApp('shiny/stockVis')
```

```
Preparing to deploy application...DONE
Uploading bundle for application: 89077...
Detecting system locale ... ko_KO
DONE
Deploying bundle: 396314 for application: 89077 ...
Waiting for task: 168849237
building: Parsing manifest
building: Fetching packages
building: Installing packages
building: Installing files
building: Pushing image: 389669
deploying: Starting instances
rollforward: Activating new instances
success: Stopping old instances
Application successfully deployed to https://raonbit.shinyapps.io/stockVis/
```

Linear algebra

❑ 기본개념

선형 대수 : 행렬 연산 , 벡터 연산, 미분, 적분 등의 선형 함수를 수를 대신하여 문자를 사용해 식을 전개하고 방정식을 푸는것

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}, \ \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \ (A:\ m\times n\text{행렬},\ \mathbf{x}:\text{열벡터})$$

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$$

$$AE=EA=A$$

$$E=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ E=\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ \cdots$$

$$(AB)^T=B^TA^T$$

$$(A+B)^T=A^T+B^T$$

$$det(A^T)=det(A)$$

$$\|\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{x} = \mathbf{x}\cdot\mathbf{x}$$

$$A=\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \rightarrow A^T=\begin{pmatrix} a & d & g \\ b & e & h \\ c & f & i \end{pmatrix}$$

$$A=\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \rightarrow A^T=\begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix}$$

❑ 기본개념

$$A = \begin{pmatrix} a_{11} a_{12} a_{13} \\ a_{21} a_{22} a_{23} \end{pmatrix} = \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix} = (\vec{b}_1\ \vec{b}_2)$$

$$AB = \begin{pmatrix} a_{11} a_{12} a_{13} \\ a_{21} a_{22} a_{23} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix} = \begin{pmatrix} \vec{a}_1 \\ \vec{a}_2 \end{pmatrix} (\vec{b}_1\ \vec{b}_2) = \begin{pmatrix} \vec{a}_1 \cdot \vec{b}_1 & \vec{a}_1 \cdot \vec{b}_2 \\ \vec{a}_2 \cdot \vec{b}_1 & \vec{a}_2 \cdot \vec{b}_2 \end{pmatrix}$$

$$\begin{bmatrix} 3\ 4\ 5 \\ 2\ 7\ 4 \end{bmatrix} \begin{bmatrix} 5\ 1 \\ 2\ 3 \\ 9\ 8 \end{bmatrix} = \begin{bmatrix} (3\ 4\ 5)\cdot(5\ 2\ 9) & (3\ 4\ 5)\cdot(1\ 3\ 8) \\ (2\ 7\ 4)\cdot(5\ 2\ 9) & (2\ 7\ 4)\cdot(1\ 3\ 8) \end{bmatrix} = \begin{bmatrix} 3x5 + 4x2 + 5x2 & 3x1 + 4x3 + 5x8 \\ 2x5 + 7x2 + 4x2 & 2x1 + 7x3 + 4x8 \end{bmatrix}$$

❑ 기본개념

$$tr(A)=a_{11}+a_{22}+\ldots+a_{nn}=\sum_{i=1}^{n} a_{ii}\ ,\ A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

$$A=\begin{pmatrix} a & b \\ c & d \end{pmatrix} \rightarrow tr(A)=a+d$$

$$A=\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \rightarrow tr(A)=a+e+i$$

$$tr(AB)=tr(BA)\neq tr(A)tr(B)$$

$$tr(ABC)=tr(BCA)=tr(CAB)\neq tr(ACB)$$

$$tr(P^{-1}AP)=tr(A)\quad (\because\ tr(AB)=tr(BA))$$

$$tr(A)=\sum_{i}\lambda_i \quad (\lambda_i:\ A\text{의 eigenvalue})$$

$$tr(A^k)=\sum_{i}\lambda_i^k \quad (\lambda_i:\ A\text{의 eigenvalue})$$

$$D=diag(a_1,a_2,\ldots,a_n)$$

$$D^k=diag(a_1^k,a_2^k,\ldots,a_n^k)$$

$$D^{-1}=diag\left(\frac{1}{a_1},\frac{1}{a_2},\ldots,\frac{1}{a_n}\right)$$

$$D^T=D$$

$$det(D)=a_1a_2\ldots a_n$$

$$diag(a_1,a_2)=\begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix},\ diag(a_1,a_2,a_3)=\begin{pmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{pmatrix},\ \ldots$$

KODB 한국데이터베이스진흥원

❑ 역행렬

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$D = [aei + bfg + cdh - ceg - bdi - afh]$$

$$A^{-1} = \frac{1}{D}\begin{bmatrix} ei - fh & -(bi - ch) & bf - ce \\ -(di - fg) & ai - cg & -(af - cd) \\ dh - eg & -(ah - bg) & ae - bd \end{bmatrix}$$

$$(AB)^{-1} = B^{-1}A^{-1}$$

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$$

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

$$AX = B$$

$$X = A^{-1}B$$

❑ 행렬식(determinant)

$$A=\begin{pmatrix} a & b \\ c & d \end{pmatrix} \rightarrow det(A)=ad-bc$$

$$A=\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \rightarrow det(A)=a(ei-fh)-b(di-fg)+c(dh-eg)$$

$det(AB)=det(A)det(B)$ (단, A,B는 동일크기의 정방행렬)

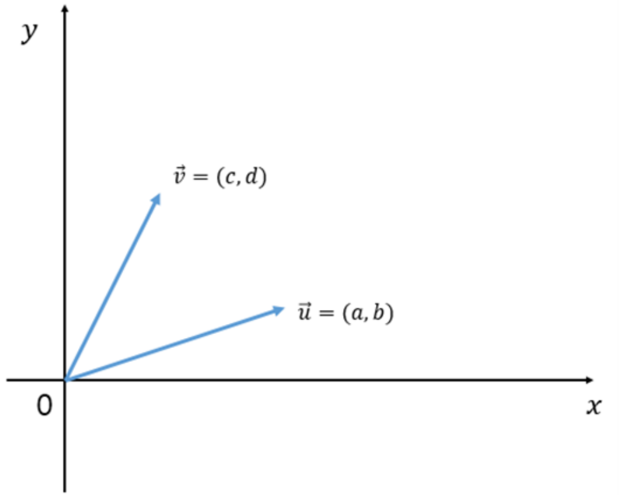
$$det(A^T)=det(A)$$

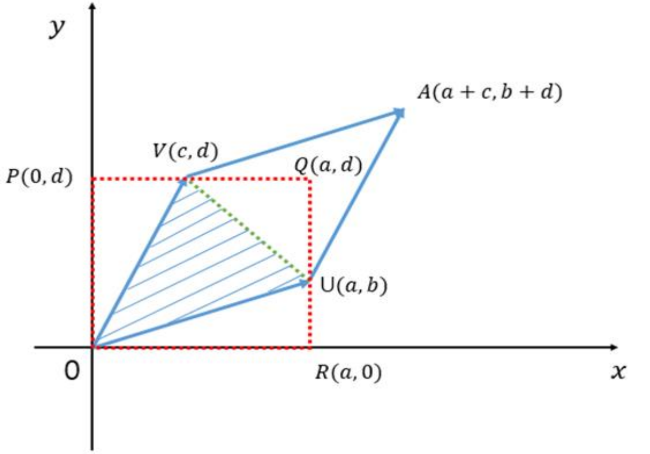
$$det(A^{-1})=\frac{1}{det(A)}$$

$$det(PAP^{-1})=det(A)$$

$det(cA)=c^n det(A)$ (단, A는 $n\times n$ 정방행렬)

$$det(A)=\sum_{j=1}^{n}(-1)^{i+j}a_{ij}det(S_{ij})$$





❑ 선형 변환(linear transformation)

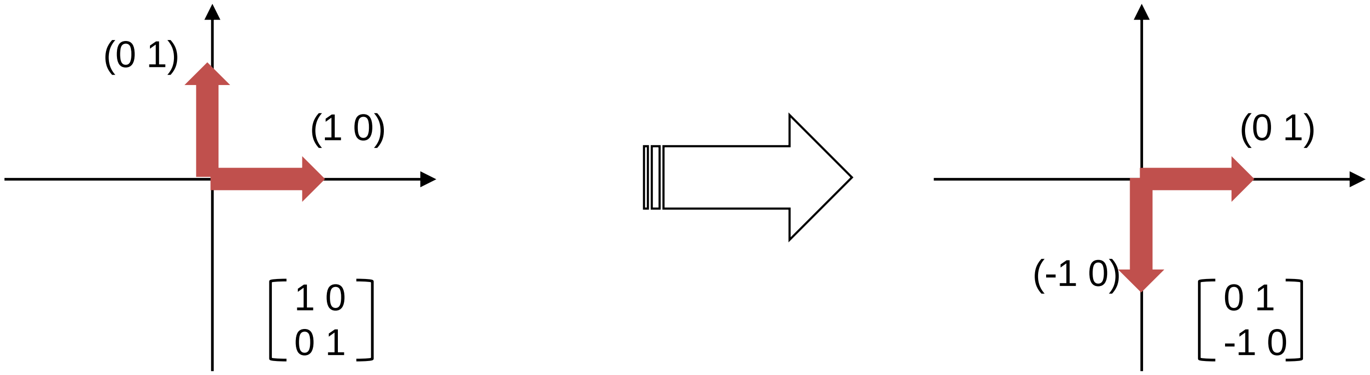
벡터 공간 V, W에 대하여 V에 속하는 임의의 두 벡터 x, y에 대해

$$f(x+y) = f(x) + f(y),$$

임의의 스칼라 a에 대해

$f(ax) = af(x)$ 를 만족하는 함수

$$f: V \;->W$$



basis vector

90도 회전

❑ 고유값(eigenvalue), 고유벡터(eigenvector)

- 선형 변환 A에 의한 변환 결과가 자기 자신의 상수배가 되는 0이 아닌 벡터(고유 벡터), 이 상수배 값(고유값)
- 고유벡터는 선형변환 A에 의해 방향은 보존되고 scale 만 변화되는 방향 벡터를 나타내고 고유값은 그 고유벡터의 변화되는 스케일 정도를 의미

$$A\mathbf{v}=\lambda\mathbf{v}$$

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}=\lambda\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

$A\mathbf{v}=\lambda\mathbf{v}$

$A\mathbf{v}-\lambda\mathbf{v}=\mathbf{0}$ $(\mathbf{0}$: 영행렬)

$(A-\lambda E)\mathbf{v}=\mathbf{0}$ $(E$: 단위행렬)

$$det(A-\lambda E)=0$$

$$A=\begin{bmatrix} 2 & 0 & -2 \\ 1 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$det(A-\lambda E)=det(\begin{bmatrix} 2 & 0 & -2 \\ 1 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}-\lambda\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix})$$

$$=\begin{vmatrix} 2-\lambda & 0 & -2 \\ 1 & 1-\lambda & -2 \\ 0 & 0 & 1-\lambda \end{vmatrix}$$

$$=(2-\lambda)((1-\lambda)(1-\lambda)-0)$$

$$=(2-\lambda)(1-\lambda)^2$$

고유값 : 1, 2

고유값 : 2

$$\begin{bmatrix} 2-\lambda & 0 & -2 \\ 1 & 1-\lambda & -2 \\ 0 & 0 & 1-\lambda \end{bmatrix}\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & -2 \\ 1 & -1 & -2 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$-2v_z=0,\ v_x-v_y-2v_z=0,\ -v_z=0$$

$$\therefore\ v_x=v_y,\ v_z=0$$

고유벡터 : (1, 1, 0)

BIGDATA ACADEMY 빅데이터 아카데미

❑ 대각화 분해(eigendecomposition)

$AP=P\Lambda$

$A=P\Lambda P^{-1}$

- P : 행렬 A의 고유벡터들을 열벡터로 하는 행렬
- Λ : 고유값들을 대각원소로 하는 대각행렬

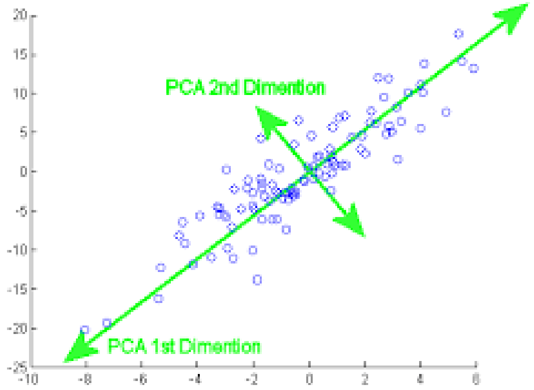
$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{bmatrix}^{-1}$$

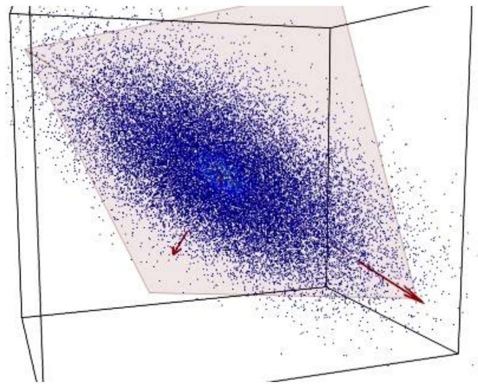
- 대칭행렬(symmetric matrix) : 정방행렬 중 대각원소를 중심으로 원소값들이 대칭되는 행렬
- 대칭행렬은 항상 고유값 대각화가 가능하며 직교행렬(orthogonal matrix)로 대각화가 가능
- 직교 벡터(orthogonal vector) : 두 벡터가 서로 수직(내적이 0)
- 정규직교 벡터(orthonomal vector) : 두 벡터가 단위 벡터이면서 서로 수직
- 직교행렬(orthogonal matrix) : 자신의 전치행렬을 역행렬로 가지는 정방행렬, 열벡터,행벡터들이 직교

$A^{-1}=A^T \qquad AA^T=E$

❑ PCA(주성분 분석)

- 분포된 데이터들의 주성분을 찾아주는 방법으로 통계 데이터 분석, 영상인식, 차원 감소, 노이즈 제거 등에서 활용

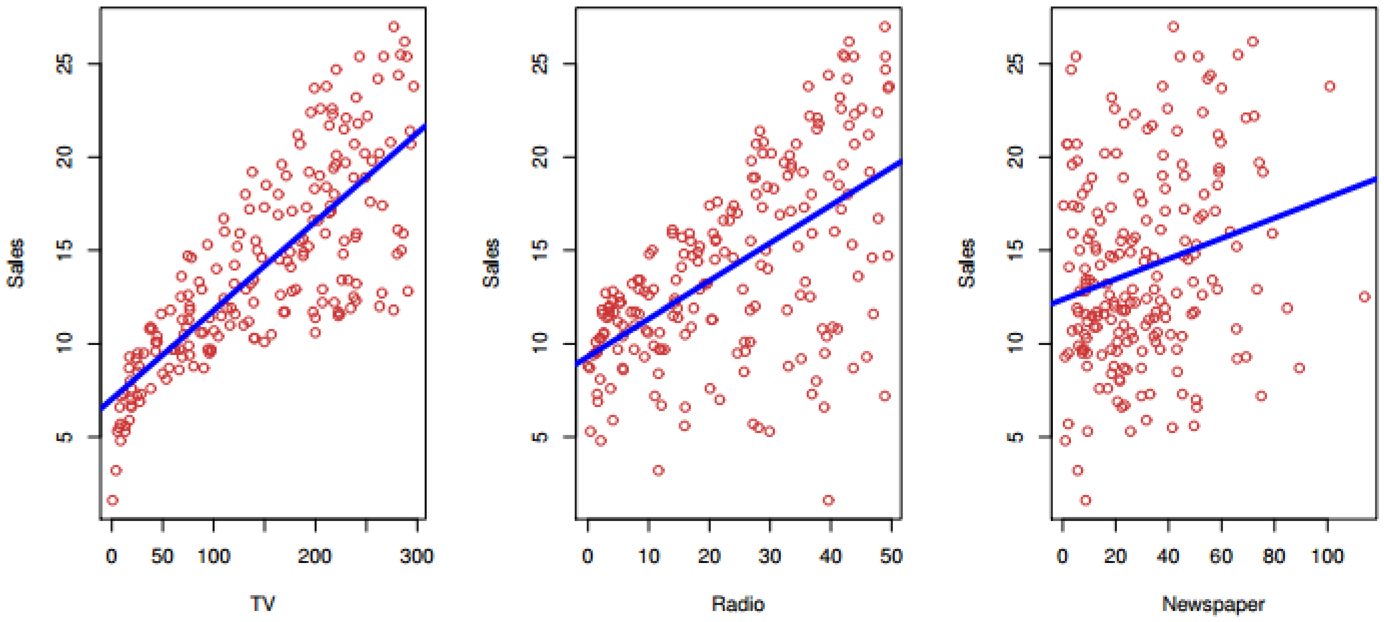




- 데이터의 분포가 그림과 같을 때, 이 데이터들의 분포 특성을 2개의 벡터로 설명
- 두개의 벡터의 방향과 크기를 알면 데이터의 분표 형태 파악 가능
- 첫번째 주성분 벡터 : 데이터들의 분산이 가장 큰 방향 벡터
- 두번째 주성분 벡터 : 첫번째 주성분 벡터와 수직이면서 그 다음으로 데이터들의 분산이 큰 방향 벡터

Linear Regression

❑ 선형 회귀



Sales = a + b*TV + c*Radio + d*Newspaper + e

❑ 선형 회귀

$$y \quad = \quad \beta_0 \quad + \quad \beta_1 x \quad + \quad \varepsilon$$

intercept slope error

coefficients
(parameters)

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

prediction value

$$e_i = y_i - \hat{y}_j$$

redidual

$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2$$

residual sum of squares

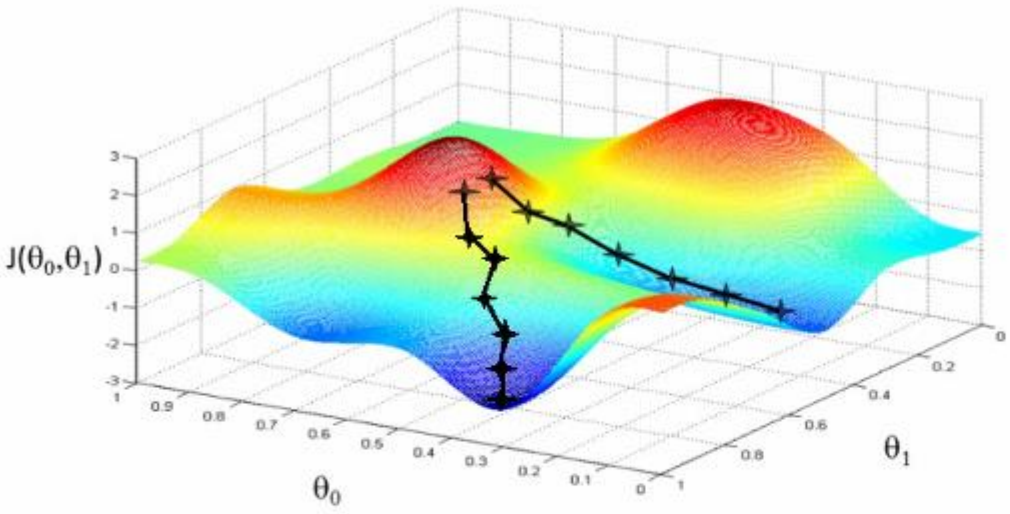
$$RSS = \left(y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1\right)^2 + \left(y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2\right)^2 + \ldots + \left(y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n\right)^2$$

❑ 선형 회귀

❖ RSS 최소화

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_{1\bar{x}}$$

$$\bar{y} = \frac{1}{n}\Sigma_{\mathrm{i}=1}^{n} y_i \qquad \bar{x} = \frac{1}{n}\Sigma_{\mathrm{i}=1}^{n} x_i$$



❑ 선형 회귀

```
library(caret)
data("faithful")

summary(faithful);head(faithful)

split = createDataPartition(y=faithful$eruptions, p=0.7, list=F)
trainData = faithful[split,]
testData = faithful[-split,]
dim(trainData);dim(testData)

model = lm(eruptions~waiting, data=trainData)
summary(model)

predict(model, testData)
coef(model)[1] + coef(model)[2] * testData[1,2]

(trainDataRMSE = sqrt(sum((model$fitted-trainData$eruptions))^2))
(testDataRMSE = sqrt(sum((predict(model,testData)-testData$eruptions))^2))
```

❑ 선형 회귀

```
Call:
lm(formula = eruptions ~ waiting, data = trainData)

Residuals:
     Min       1Q   Median       3Q      Max
-1.27825 -0.34440 -0.00492  0.34450  1.14211

Coefficients:
             Estimate   Std. Error  t value   Pr(>|t|)
(Intercept)  -1.971041   0.179954   -10.95   <2e-16 ***
waiting       0.077578   0.002513    30.86   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4671 on 190 degrees of freedom
Multiple R-squared:  0.8337,    Adjusted R-squared:  0.8328
F-statistic: 952.6 on 1 and 190 DF,  p-value: < 2.2e-16
```

- Std(Standard) Error : repeated sampling의 표준 오차

$$S_E = \sqrt{\frac{RSS}{n-2}}$$

- t value(t-statistic) : 모델에서 얻은 값과 귀무가설 값의 표준화된 차이

$$t = \frac{\hat{\beta}_1 - \beta_1}{s_E(\hat{\beta}_1)} \qquad , \beta_1 = 0$$

- Pr(>|t|) : p value, 유의 수준, 귀무가설이 옳다는 가정하에 모델에서 얻은 값 또는 그 이상의 값을 얻을 확률

❑ 선형 회귀

❖ R-squared : 회귀식이 얼마나 원래의 자료를 설명하는가

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}} \qquad \mathrm{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

❖ Adjusted R-squared : R-squared 값이 변수가 추가될때 마다 증가되는것을 방지, 새로 추가된 변수가 y의 예측에 기여하는 만큼만 증가하도록 조정

❖ F-statistic : 이 모델이 통계적으로 의미가 있는가? (F-statistic가 충분히 크면 통계적으로 유의미

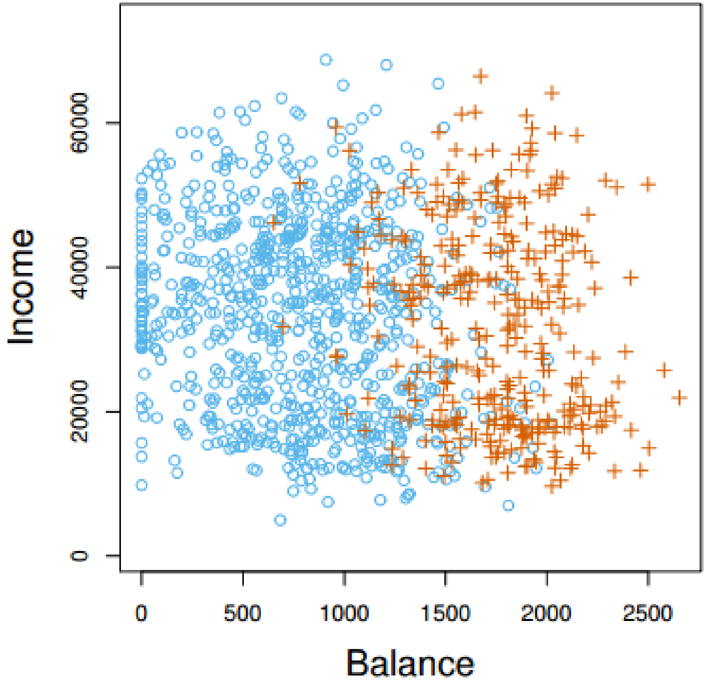
- 오차들의 크기가 작을 수록
- 표본의 수가 많을 수록
- 독립변수의 수가 적을 수록

❑ 선형 회귀

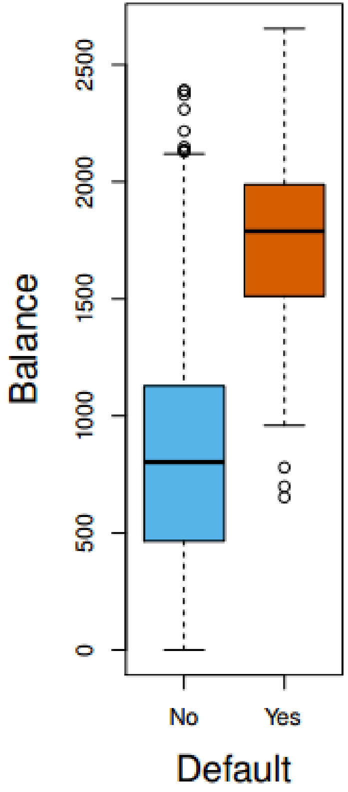

```
library(ISLR)
data(Wage)

summary(Wage);head(Wage)

split = createDataPartition(y=Wage$wage, p=0.7, list=F)
trainData = Wage[split,]
testData = Wage[-split,]
dim(trainData);dim(testData)

model = train(wage~ age+jobclass+education, mothod="lm", data=trainData)
#model = train(wage~ ., mothod="lm", data=trainData)
finalModel = model$finalModel

library(ggplot2)

pred = predict(model, testData)
qplot(wage, pred, colour=year, data=testData)
```

Classification

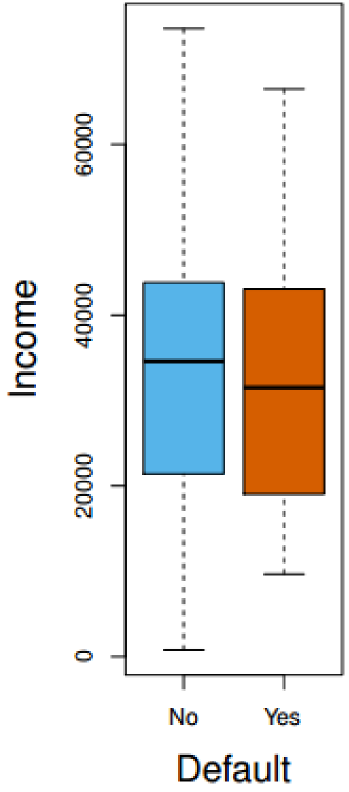
❑ 분류 분석



Income
60000
40000
20000
0
0
500
1000
1500
2000
2500
Balance



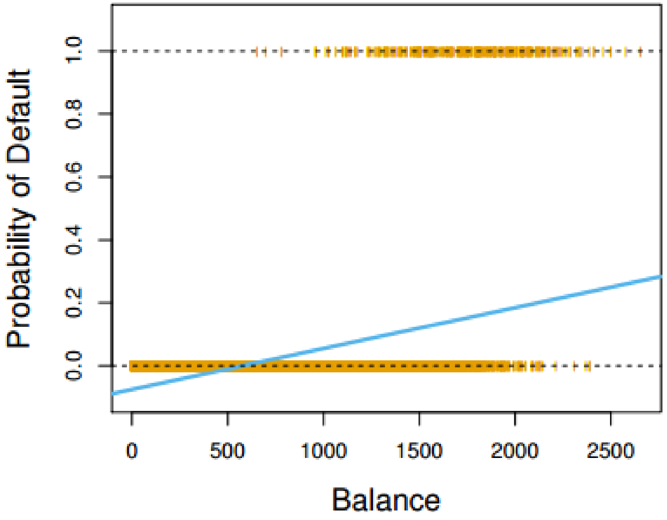
Balance
2500
2000
1500
1000
500
0
No
Yes
Default



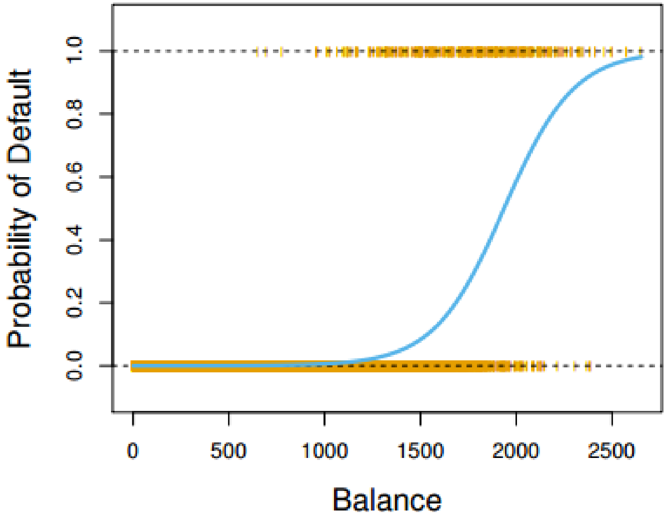
Income
60000
40000
20000
0
No
Yes
Default

❑ 분류 분석 - Logistic regression

❖ Linear regression



❖ Logistic regression



$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

❑ 분류 분석 - Naïve Bayesian classification

❖ 가정 : 주머니에 빨간공 60%, 파란공 40%, 빨간공의 20%는 깨졌고, 파란공의 30%는 깨졌다. 임의로 꺼낸 공이 깨졌을때 이 공이 빨간색일 확률은?

- P(깨진공/파란색) : likelihood
- P(파란색/깨진공) : posterior

❖ 깨진공이 파란색인지 빨간색인지 알아내는 방법

- Maximum A posterior : 깨진 공이 빨간색일 확률과 깨진 공이 파란색일 확률을 비교해서 더 확률 높은 쪽을 선택
- Maximum Likehood : 파란색이 깨진공일 확률과 빨간색이 깨진공일 확률을 계산해서 더 확률 높은 쪽을 선택

❖ 가정 : 빨간공 60개 파란공 40개, 빨간공 12개 깨짐, 파란공 12개 깨졌다. 임의로 꺼낸 공이 깨졌을때 이 공이 빨간색일 확률은?

❑ 분류 분석 - Naïve Bayesian classification

$$p(Y = k|X = x) = \frac{P(X = \chi|y = k)P(Y = k)}{p(X = x)}$$

❖ P(빨간색/깨진공) = P(깨진공/빨간색) * P(빨간색) / P(깨진공)

= P(깨진공/빨간색) * P(빨간색) / (P(깨진공/빨간색) * P(빨간색) + P(깨진공/파란색) * P(파란색))

= 0.2 * 0.6 / (0.2 * 0.6 + 0.3 * 0.4)

❖ 확장 : 주머니에 빨간공 35%, 파란공 55%, 흰공 : 10%, 빨간공의 15%는 깨졌고, 파란공의 20%는 깨졌고, 흰공의 35%는 깨짐 임의로 꺼낸 공이 깨졌을때 이 공이 빨간색일 확률은?

❖ P(빨간색/깨진공) = P(깨진공/빨간색) * P(빨간색) / P(깨진공)

= P(깨진공/빨간) * P(빨간색) / (P(깨진공/빨간색)* P(빨간색) + P(깨진공/파란색)* P(파란색) + P(깨진공/흰색) * P(흰색))

= 0.15 * 0.35 / (0.15*0.35 + 0.2*0.55 + 0.35*0.1)

❑ 분류 분석 - Naïve Bayesian classification

no	words	class
1	fun, couple, love, love	comedy
2	fast, furious, shoot	action
3	couple, fly, fast, fun, fun	comedy
4	furious, shoot, shoot, fun	action
5	fly, fast, shoot, love	action
6	fast, furious, fun	?

P(comedy/fast,furious,fun) = P(fast/comedy) * P(furious/comedy) * P(fun/comedy) * P(comedy)

= 1/9 * 0/9 * 3/9 * 2/5 = 0

P(action/fast,furious,fun) = P(fast/action) * P(furious/action) * P(fun/action) * P(action)

= 2/11 * 2/11 * 1/11 * 3/5 = 0.0018

Laplace smoothing

P(comedy/fast,furious,fun) = P(fast/comedy) * P(furious/comedy) * P(fun/comedy) * P(comedy)

= (1+1)/(9+7) * (0+1)/(9+7) * (3+1)/(9+7) * 2/5 = 0.00078

P(action/fast,furious,fun) = P(fast/action) * P(furious/action) * P(fun/action) * P(action)

= (2+1)/(11+7) * (2+1)/(11+7) * (1+1)/(11+7) * 3/5 = 0.0018

❑ 분류 분석 - Naïve Bayesian classification

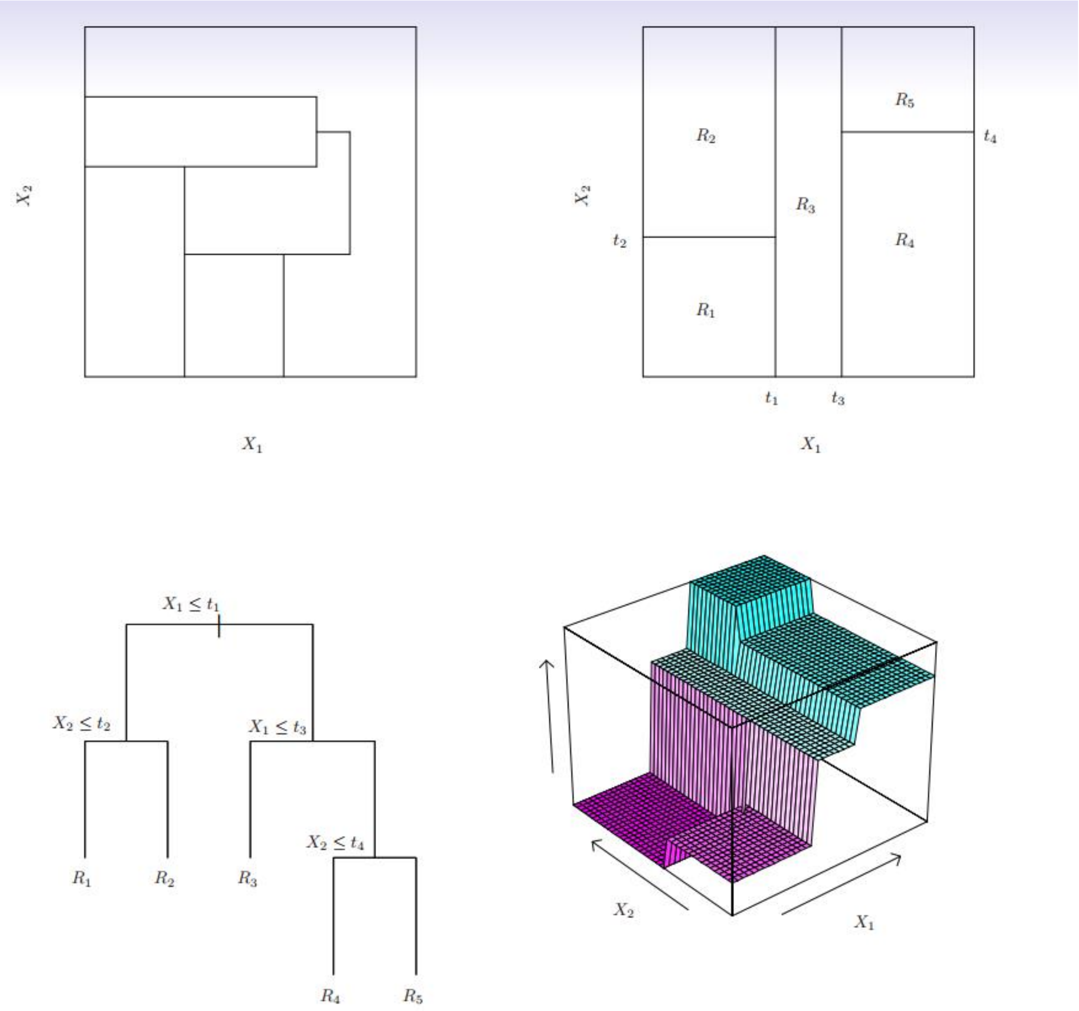
```
library (e1071)

data (HouseVotes84, package="mlbench")
summary(HouseVotes84); head(HouseVotes84)

model = naiveBayes(Class ~ ., data = HouseVotes84)
predict(model, HouseVotes84[1:20,-1])

# 확률
predict(model, HouseVotes84[1:20,-1], type = "raw")

pred = predict(model, HouseVotes84[,-1])
(table <- table(pred, HouseVotes84$Class))
confusionMatrix(table)

## Laplace smoothing:
model <- naiveBayes(Class ~ ., data = HouseVotes84, laplace = 3)
pred <- predict(model, HouseVotes84[,-1])
table <- table(pred, HouseVotes84$Class)
confusionMatrix(table)
```

```
library(ROCR)
HouseVotes84$republican =
        factor(1*(HouseVotes84$Class == 'republican'))
pred = predict(model, HouseVotes84[,-1], type = 'raw')
pred = pred[,2]
plot(performance(prediction(pred,
HouseVotes84$republican), 'tpr', 'fpr'))
```

KODB 한국데이터베이스진흥원
BIGDATA ACADEMY 빅데이터아카데미

Tree-based model

❑ Tree-based model



BIGDATA ACADEMY 빅데이터 아카데미

❑ Tree-based model

❖ Basic algorithm

i. 전체 데이터를 포함하는 root node 생성
ii. 만일 샘플들이 모두 같은 클래스이면 node는 leaf가 되고 해당 클래스로 label 부여
iii. 그렇지 않으면 information gain이 높은 속성 선택
iv. 선택된 속성으로 branch를 만들고 하위 node 생성
v. 각 노드에 대하여 ii 부터 반복

➢ 정지 조건 : 해당 node에 속한는 데이터들이 모두 같은 클래스를 가지거나 상위 node에서 모든 속성을 사용

❖ Information gain : 특정 속성을 기준으로 데이터를 구분할 때 감소되는 entropy의 양

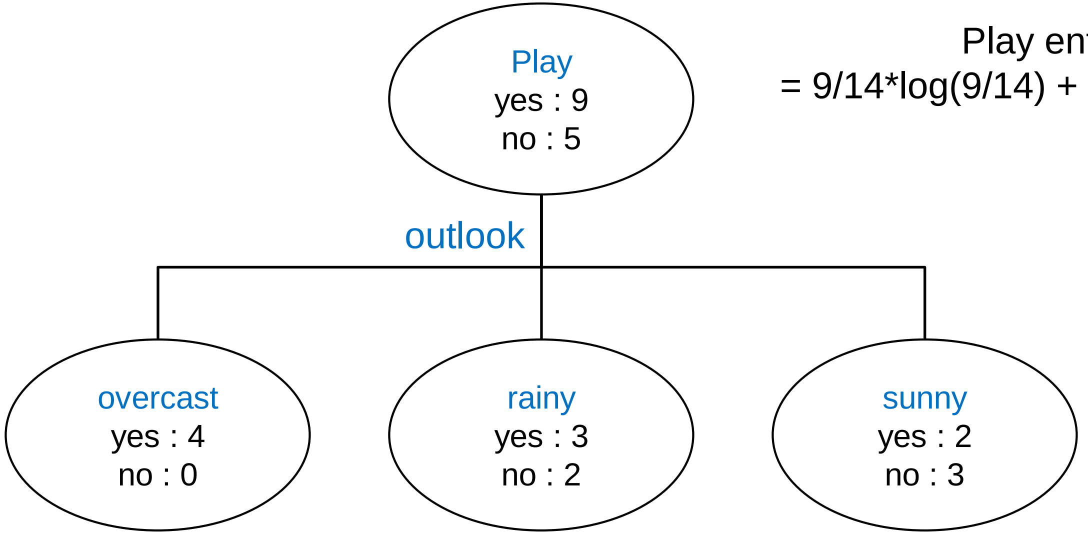
$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (\ S_v = \{s \in S | A(s) = v\}\ )$$

❖ Entropy(무질서도) $H(p) = -\sum_{x \in X} p(x) \log p(x)$

❑ Tree-based model

Outlook	Temperature Numeric	Temperature Nominal	Humidity Numeric	Humidity Nominal	Windy	Play
overcast	83	hot	86	high	FALSE	yes
overcast	64	cool	65	normal	TRUE	yes
overcast	72	mild	90	high	TRUE	yes
overcast	81	hot	75	normal	FALSE	yes
rainy	70	mild	96	high	FALSE	yes
rainy	68	cool	80	normal	FALSE	yes
rainy	65	cool	70	normal	TRUE	no
rainy	75	mild	80	normal	FALSE	yes
rainy	71	mild	91	high	TRUE	no
sunny	85	hot	85	high	FALSE	no
sunny	80	hot	90	high	TRUE	no
sunny	72	mild	95	high	FALSE	no
sunny	69	cool	70	normal	FALSE	yes
sunny	75	mild	70	normal	TRUE	yes

❑ Tree-based model



Play entropy(9,5)
= 9/14*log(9/14) + 5/14*log(5/14) = 0.94

Information gain(outlook)
= 0.94 - (4/14*0 + 5/14*0.97 + 5/14*0.97)
= 0.23

overcast entropy(4,0)
= 4/4*log(4/4) + 0/4*log(0/4) = 0

sunny entropy(4,0)
= 2/5*log(2/5) + 3/5*log(3/5) = 0.97

rainy entropy(4,0)
= 3/5*log(3/5) + 2/5*log(2/5) = 0.97

❑ Tree-based model

```
library(C50)
library(caret)

data(iris)
summary(iris);head(iris)

split = createDataPartition(y=iris$Species, p=0.7, list=F)
trainData = iris[split,]
testData = iris[-split,]
dim(trainData);dim(testData)

# C50
trainDataX = trainData[,1:4]
trainDataY = trainData[,5]
model = C5.0( trainDataX, trainDataY )
summary( model )
```

```
#boosting
model = C5.0( trainDataX, trainDataY, trials=10)
summary( model )
plot(model)

testDataX = testData[,1:15]
testDataY = testData[,16]
pred = predict(model, testDataX, type="class")
sum(pred==testDataY ) / length(pred)

# rpart
rpart_model = train(Species~., method="rpart",
data=trainData)
print(rpart_model$finalModel)

library(rattle)
fancyRpartPlot(rpart_model$finalModel)

rpart_pred = predict(rpart_model, testData)
sum(rpart_pred==testDataY ) / length(rpart_pred)
```

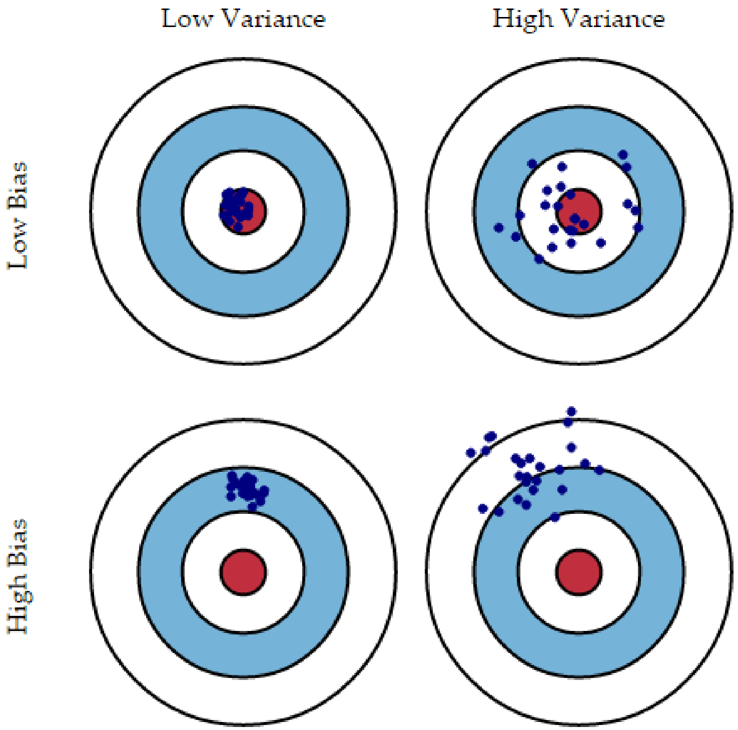
Cross Validation

❑ Model error

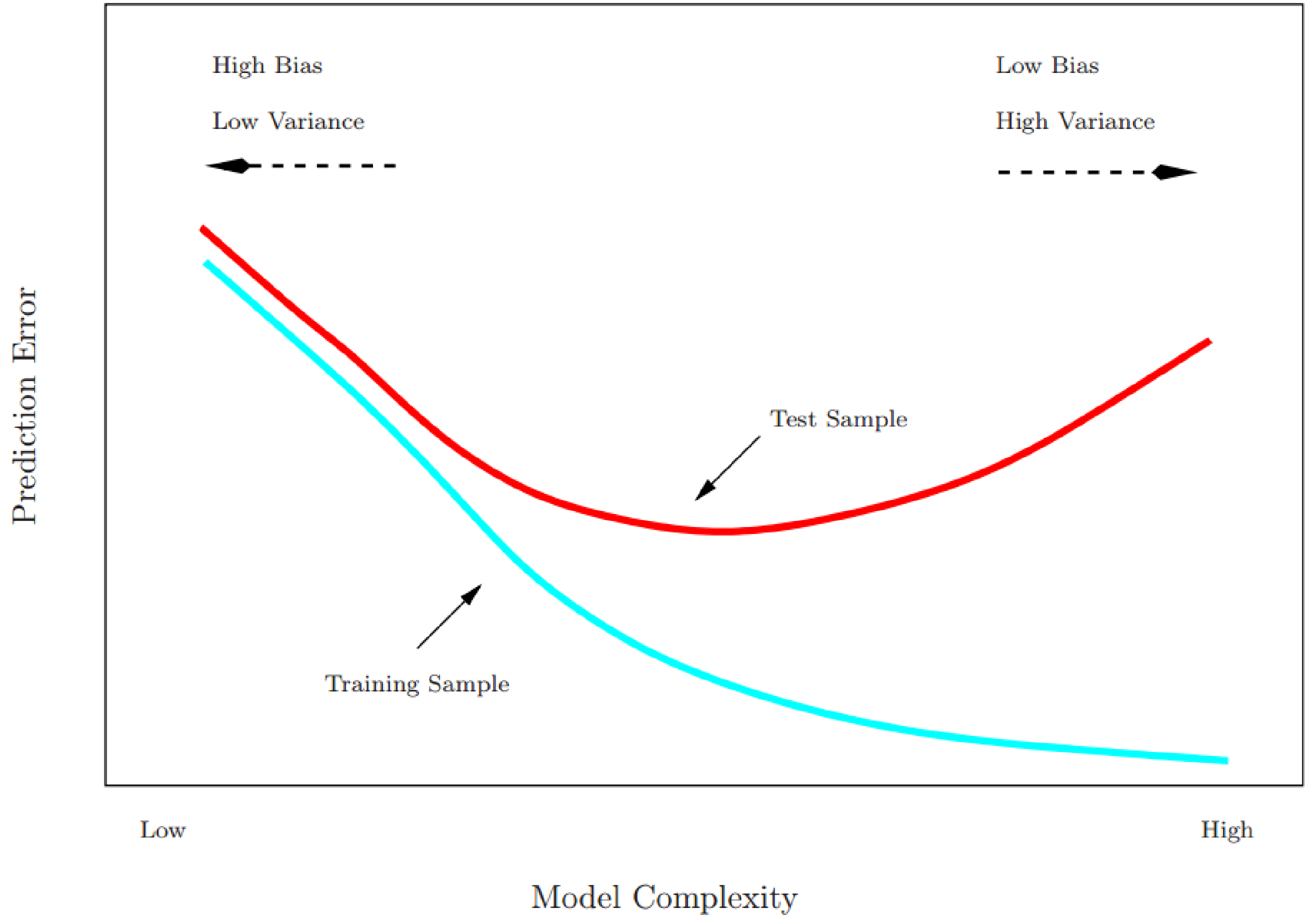
$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\hat{f}(x) - E[\hat{f}(x)]\right]^2 + \sigma_e^2$$

$$Err(x) = \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{Irreducible\ Error}$$

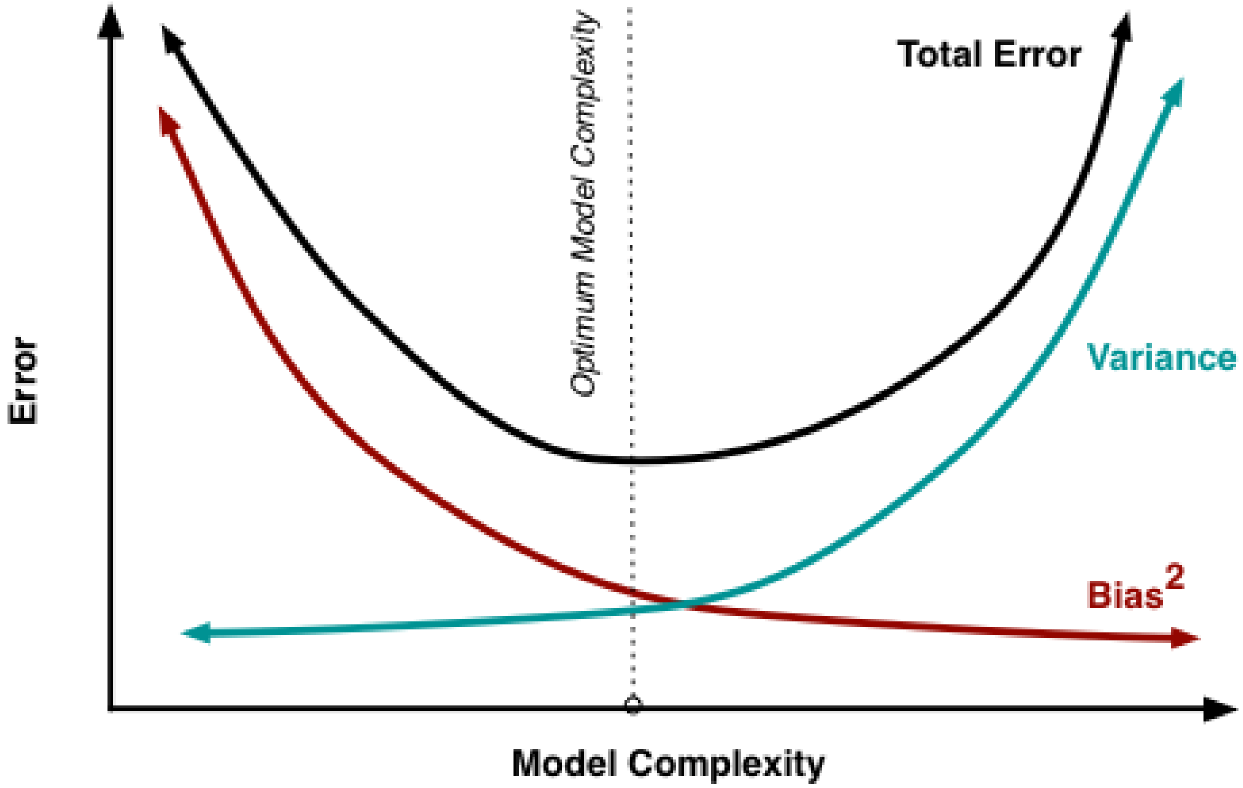
- Bias error : 모델의 예측 값과 실제 값의 차이, bias error가 높다는 것은 모델 성능이 안좋은 것을 의미
- Variance : 모델이 서로 다른 데이터에 대해 같은 성능을 보장하는가. bias error가 낮은데도 variance가 높다는 것은 training data 에 대해 over-fit되었음을 의미



❑ Model error


High Bias
Low Variance
Low Bias
High Variance
Test Sample
Training Sample
Prediction Error
Low
High
Model Complexity

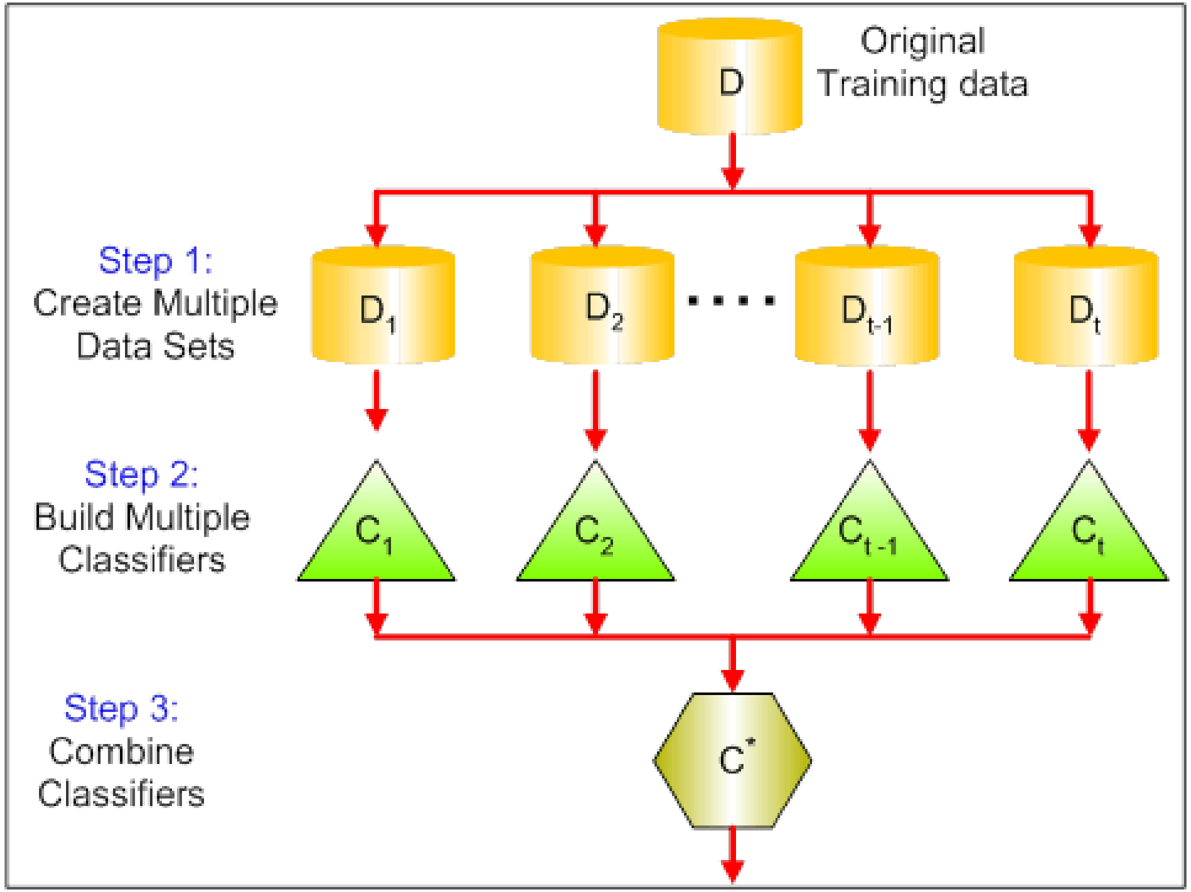
❑ Model error



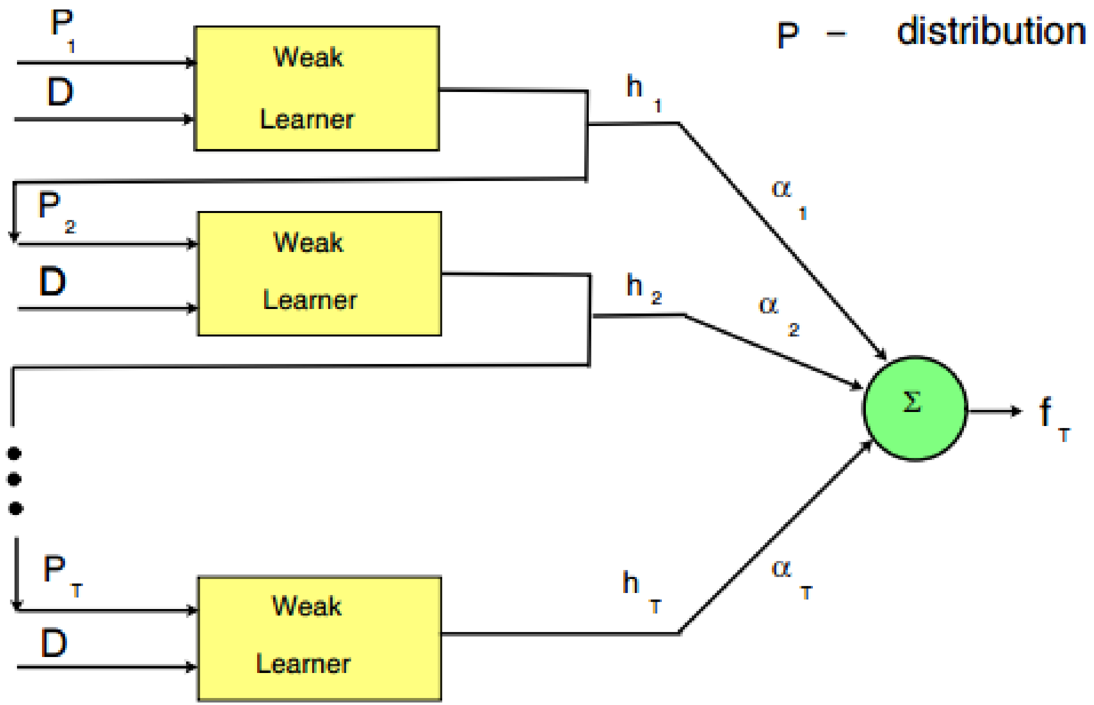
Error
Optimum Model Complexity
Total Error
Variance
Bias²
Model Complexity

Ensemble learning

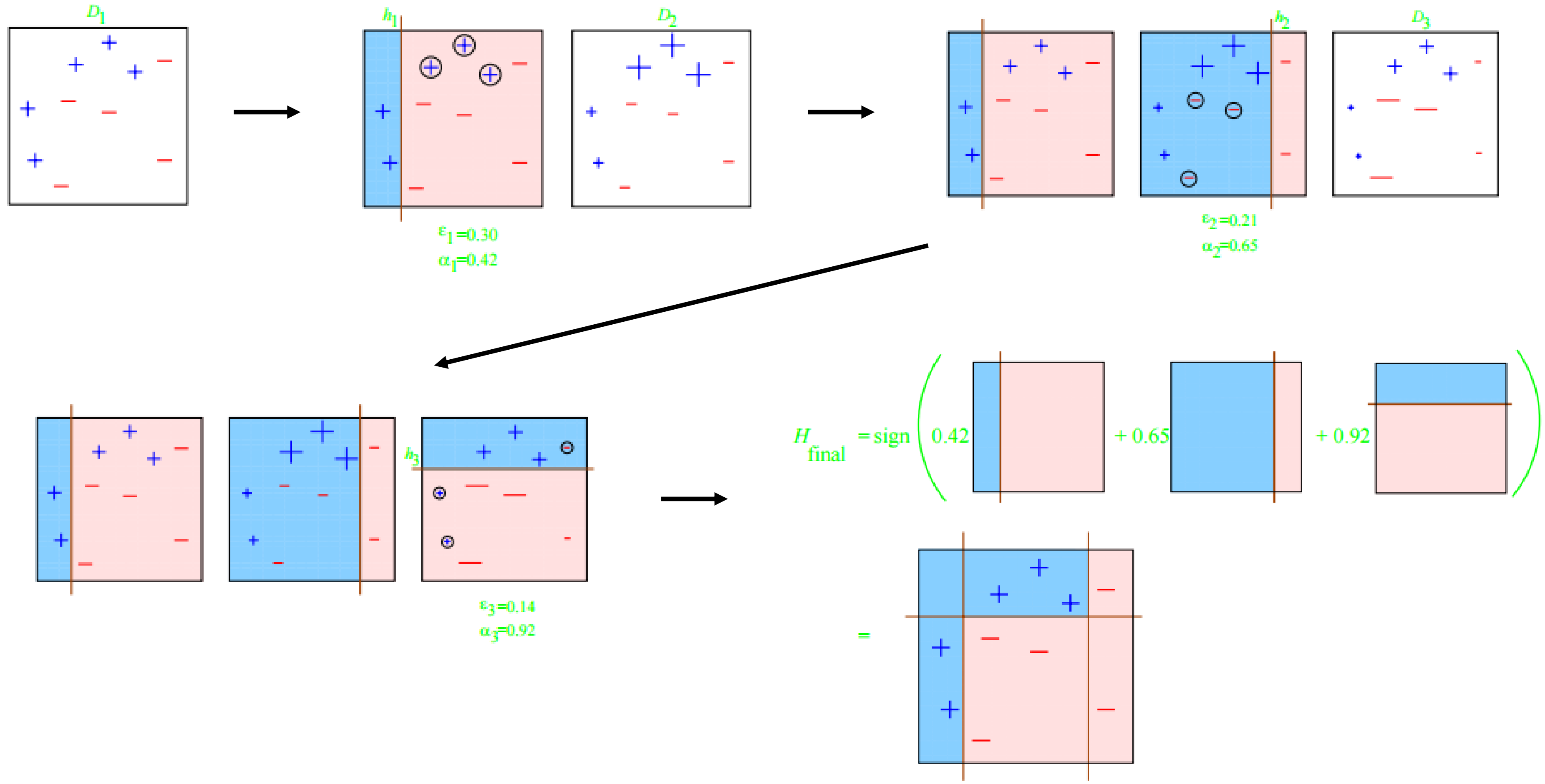
❑ Bagging(Bootstrap aggregating)



❑ Boosting

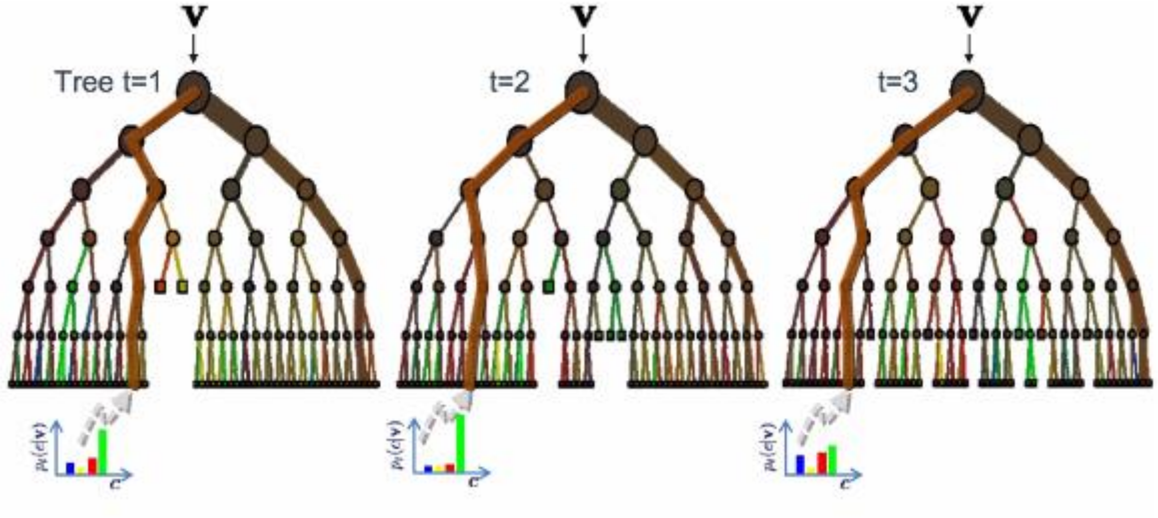

D – data
P – distribution
P_1
D
Weak Learner
h_1
α_1
P_2
D
Weak Learner
h_2
α_2
P_T
D
Weak Learner
h_T
α_T
Σ
f_T

❑ Boosting



D1
h1
D2
ε1=0.30
α1=0.42
h2
D3
ε2=0.21
α2=0.65
h3
ε3=0.14
α3=0.92
H final = sign (0.42 + 0.65 + 0.92)
=

❑ Random forests



1. bootstrap samples
2. bootstrap variables
3. grow multiple trees and votes

장점 : 정확도

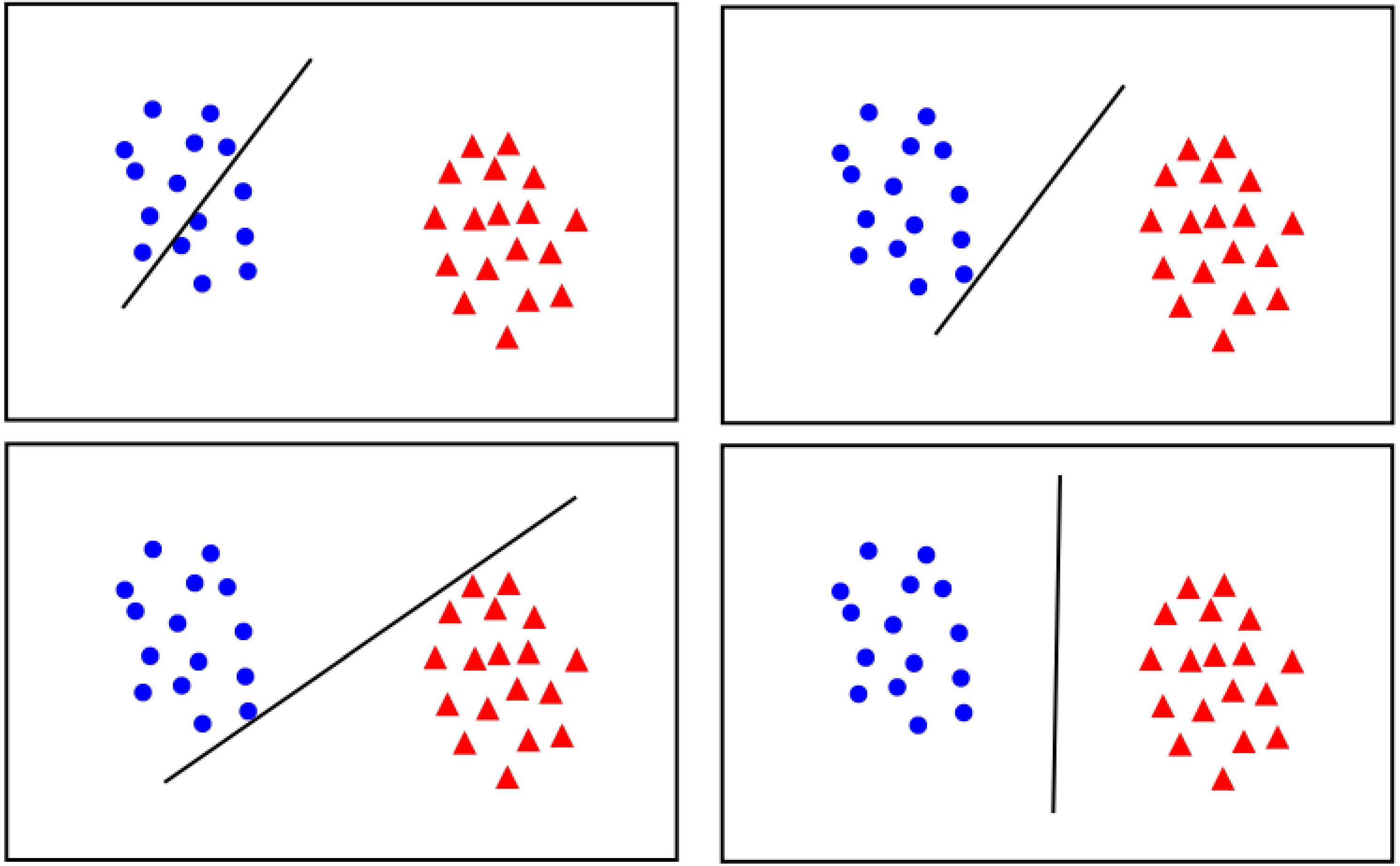
단점 : 속도, 해석력, 과적합

❑ Ensemble

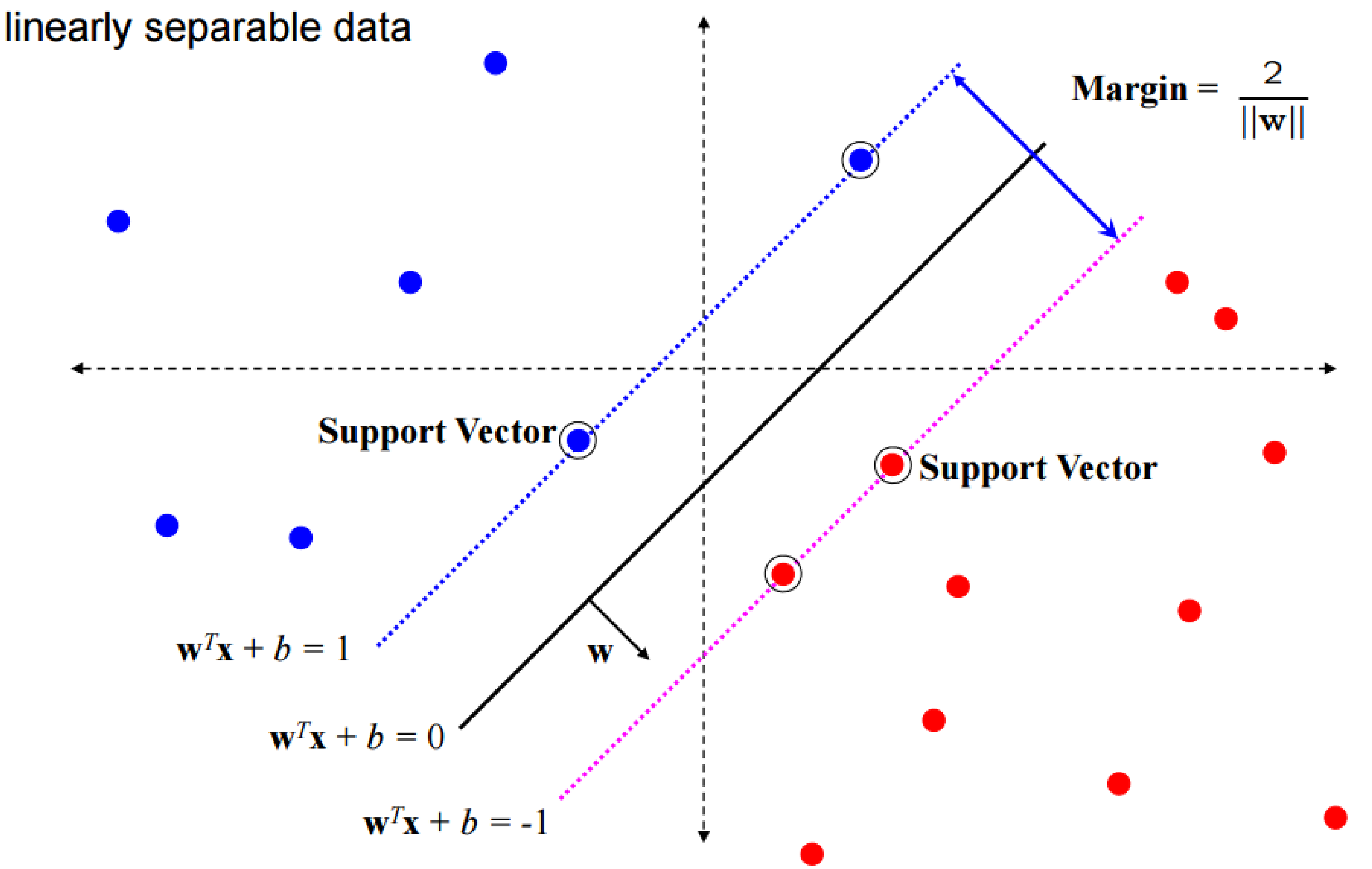
```
library(ISLR)
library(ggplot2)
library(caret)

data(Wage);summary(Wage);head(Wage)

split = createDataPartition(y=Wage$wage, p=0.7, list=F)
trainData = Wage[split,]; testData = Wage[-split,]
dim(trainData);dim(testData)

#rpart
rpart_model = train(wage ~ ., method="rpart", data=trainData, verbose=F)
qplot(predict(rpart_model,testData), wage, data=testData)

# boosting
boost_model = train(wage ~ ., method="gbm", data=trainData, verbose=F)
qplot(predict(boost_model,testData), wage, data=testData)


# randomforest(bagging)
rf_model = train(wage ~ ., method="rf", data=trainData, verbose=F)
qplot(predict(rf_model,testData), wage, data=testData)
```

Support Vector Machine

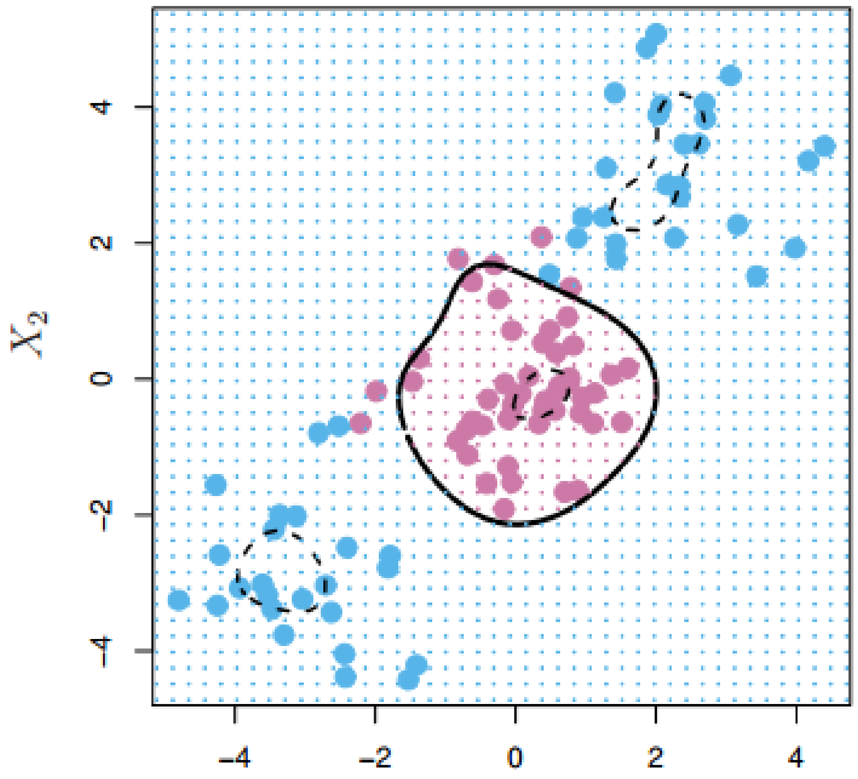
❑ SVM



❑ SVM


linearly separable data
Margin = $\frac{2}{||\mathbf{w}||}$
Support Vector
Support Vector
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

❑ SVM



KODB 한국데이터베이스진흥원
BIGDATA ACADEMY 빅데이터아카데미

❑ SVM

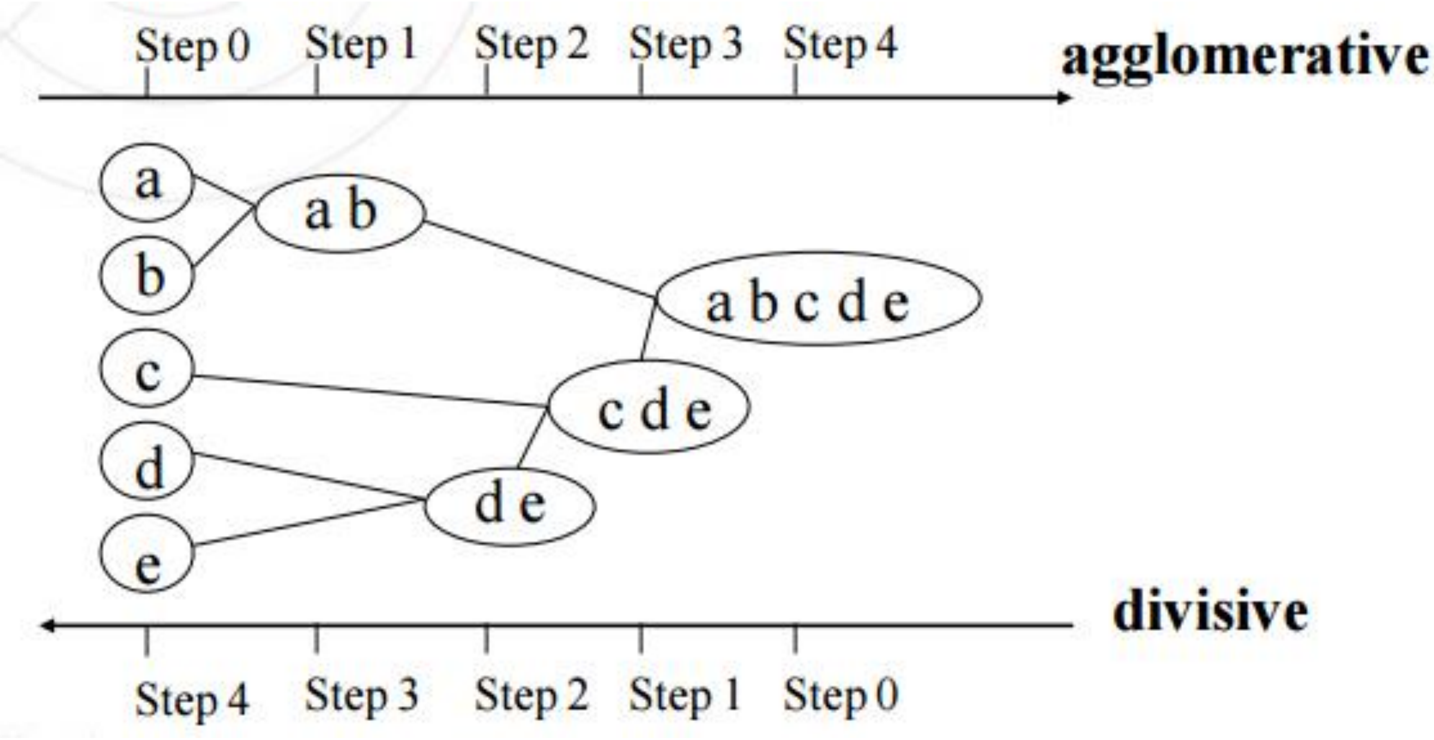
```
library(caret)

#R CMD INSTALL kernlab_0.9-24.zip
x = iris[,1:4]
y = iris[,5]

folds = createMultiFolds(y, k = 10, times = 5)

#Linear SVM
L_model = train(x,y,method="svmLinear",tuneLength=5,
trControl=trainControl(method='repeatedCV',index=folds,classProbs=TRUE))

#Poly SVM
P_model = train(x,y,method="svmPoly",tuneLength=5,
trControl=trainControl(method='repeatedCV',index=folds,classProbs=TRUE))

#Fit a Radial SVM
R_model = train(x,y,method="svmRadial",tuneLength=5,
trControl=trainControl(method='repeatedCV',index=folds,classProbs=TRUE))
```

❑ SVM

```
#Compare 3 models
resamps = resamples(list(Linear = L_model, Poly = P_model,
Radial = R_model))
summary(resamps)
bwplot(resamps, metric = "Accuracy")
densityplot(resamps, metric = "Accuracy", auto.key=TRUE)

pred = predict(L_model,x,type='prob')

library(caTools)
colAUC(pred,y,plot=TRUE)
```

Clustering

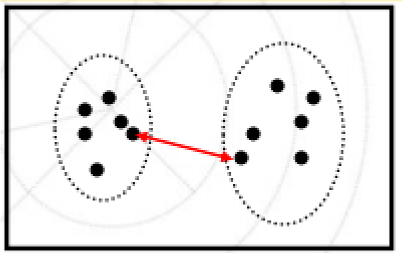
❑ Hierarchical clustering



Hierarchical clustering

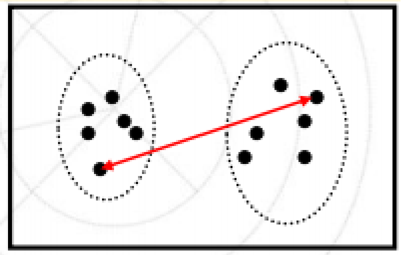
❖ 클러스터간의 유사도 측정 방법

- Single Link



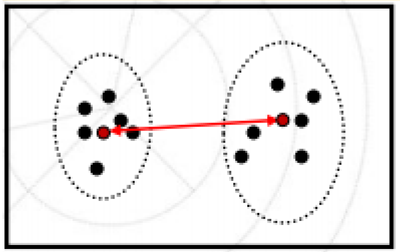
가장 가까운 점의 거리
(neighbouring joining)

- Complete Link



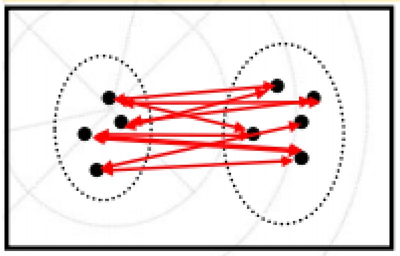
가장 먼 점의 거리

- Median



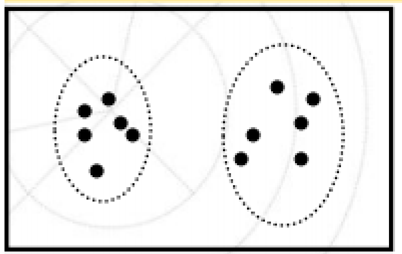
중앙값 거리
(centroid)

- Average Link



모든 점들의 평균 거리

- Ward Link



점에서 중심까지의 편차에
대한 제곱을 합한 것

Hierarchical clustering

- Similarity measures between clusters : Lance-Williams formula

$$d(i+j,k)=a_i d(i,k)+a_j d(j,k)+b d(i,j)+c|d(i,k)-d(j,k)|$$

Single-link	$a_i=a_j=0.5$; $b=0$; $c=-0.5$ $\quad d(i+j,k)=\min\{d(i,k),d(j,k)\}$
Complete-link	$a_i=a_j=0.5$; $b=0$; $c=0.5$ $\quad d(i+j,k)=\max\{d(i,k),d(j,k)\}$
Centroid	$a_i=\frac{n_i}{n_i+n_j}$ $\quad a_j=\frac{n_j}{n_i+n_j}$ $\quad b=-\frac{n_i n_j}{(n_i+n_j)^2}$ $\quad c=0$ $\quad d(i+j,k)=d(\mu_{i+j},\mu_k)$
Median	$a_i=a_j=0.5$; $b=-0.25$; $c=0$
(Average link)	$a_i=\frac{n_i}{n_i+n_j}$ $\quad a_j=\frac{n_j}{n_i+n_j}$ $\quad b=c=0$ $\quad d(C_i,C_j)=\frac{1}{n_i n_j}\sum_{a\in C_i,b\in C_j} d(a,b)$
Ward's Method (minimum variance)	$a_i=\frac{n_k+n_i}{n_k+n_i+n_j}$ $\quad a_j=\frac{n_k+n_j}{n_k+n_i+n_j}$ $\quad b=-\frac{n_k}{n_k+n_i+n_j}$ $\quad c=0$

KODB 한국데이터베이스진흥원
BIGDATA ACADEMY 빅데이터 아카데미

K-means clustering

- Cluster : C1 ~ Ck
  - 모든 데이터는 적어도 하나의 클러스터에 속한다
  - 하나 이상의 클러스터에 속하는 데이터는 없다
- WCV(within cluster variation) 이 최소화 되도록 클러스터 조정

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \text{WCV}(C_k) \right\}.$$

- K개의 클러스터의 WCV 의 총합이 최소가 되도록 데이터를 K 개의 클러스터로 나눔

$$\text{WCV}(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2, \qquad \underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

❑ K-means clustering