05 CHAPTER

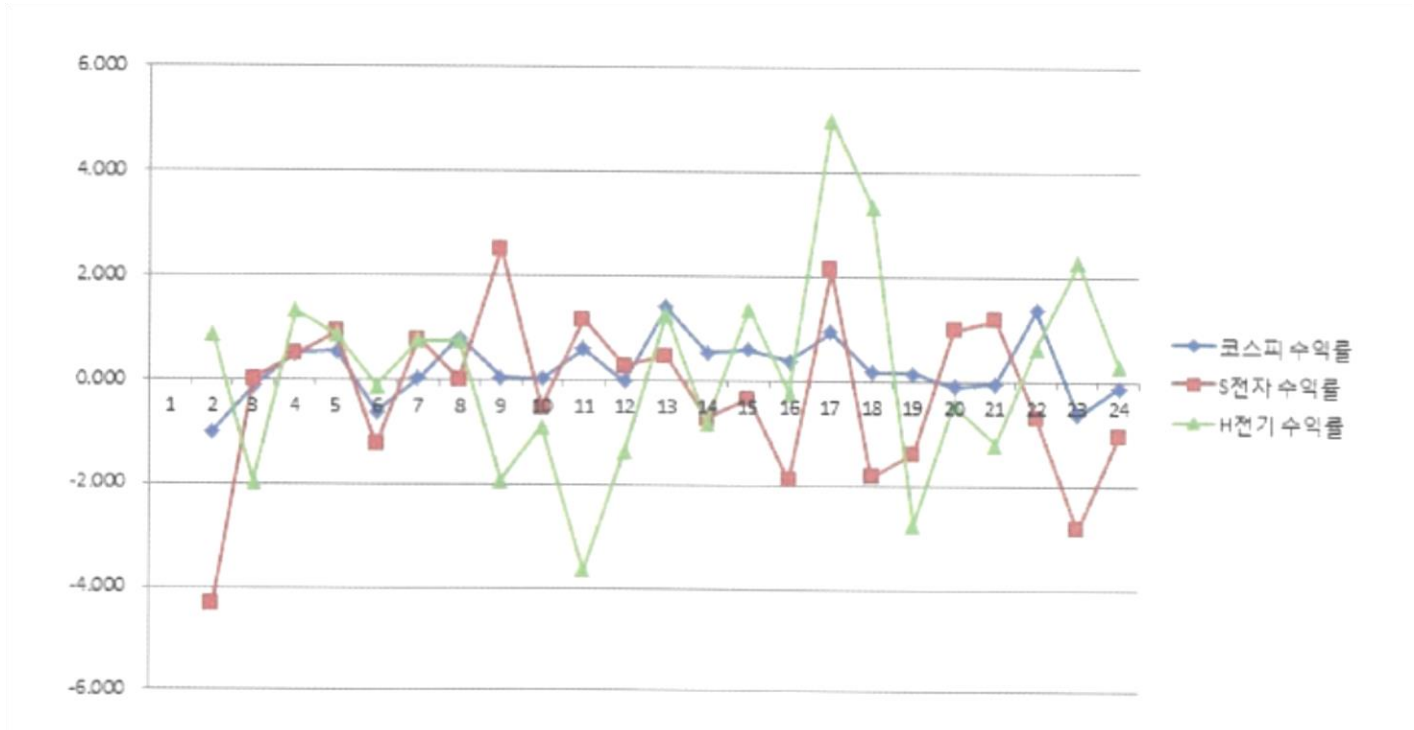
Regression

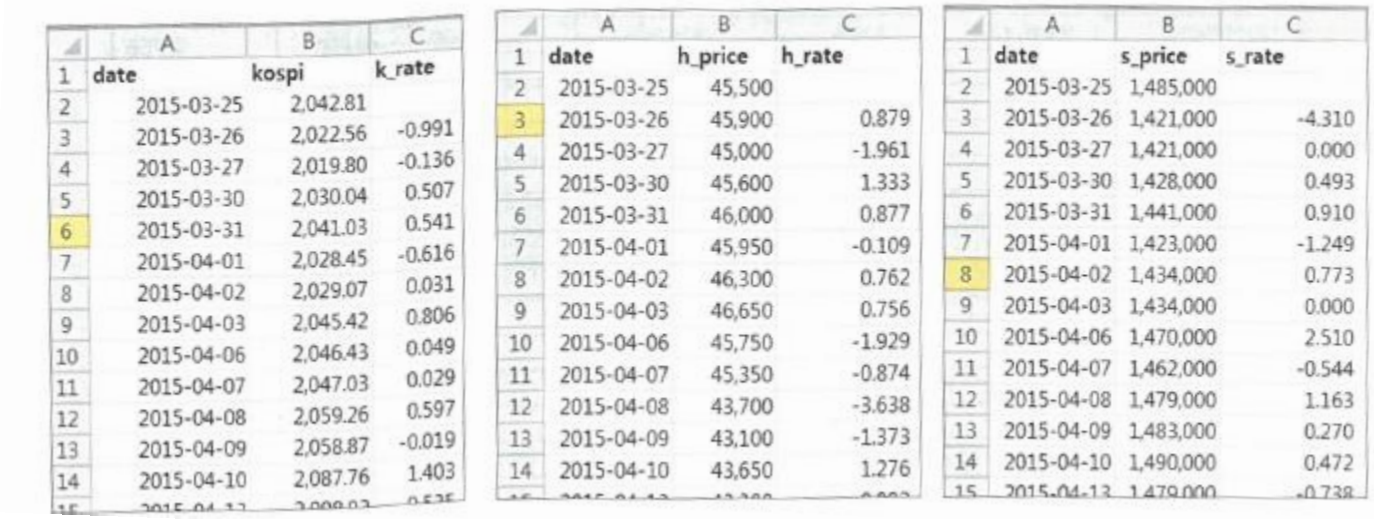
데이터분석을 위해서는 약간의 업무지식이
있어야 되는데
이거를 Domain Knowledge라고 합니다.

$$\frac{(\text{오늘코스피지수} - \text{전일코스피지수})}{\text{전일코스피지수}} * 100$$

날짜	코스피지수	등락률
2015/03/25	2,042.81	
2015/03/26	2,022.56	-0.991281617
2015/03/27	2,019.80	-0.136460723
2015/03/30	2,030.04	0.506980889
2015/03/31	2,041.03	0.541368643
2015/04/01	2,028.45	-0.616355468
2015/04/02	2,029.07	0.03056521
2015/04/03	2,045.42	0.805787873
2015/04/06	2,046.43	0.049378612
2015/04/07	2,047.03	0.029319351
2015/04/08	2,059.26	0.597450941
2015/04/09	2,058.87	-0.018938842
2015/04/10	2,087.76	1.403196899
2015/04/13	2,098.92	0.5345442
2015/04/14	2,111.72	0.60983744
2015/04/15	2,119.96	0.390203247
2015/04/16	2,139.90	0.940583785
2015/04/17	2,143.50	0.16823216
2015/04/20	2,146.71	0.149755073
2015/04/21	2,144.79	-0.089439188
2015/04/22	2,143.89	-0.04196215
2015/04/23	2,173.41	1.376936317
2015/04/24	2,159.80	-0.626204904

	A	B	C	D	E	F	G
1	날짜	코스피지수	코스피 수익률	S전자 종가	S전자 수익률	H전기 종가	H전기 수익률
2	2015/03/25	2,042.81		1,485,000		45,500	
3	2015/03/26	2,022.56	-0.991	1,421,000	-4.310	45,900	0.879
4	2015/03/27	2,019.80	-0.136	1,421,000	0.000	45,000	-1.961
5	2015/03/30	2,030.04	0.507	1,428,000	0.493	45,600	1.333
6	2015/03/31	2,041.03	0.541	1,441,000	0.910	46,000	0.877
7	2015/04/01	2,028.45	-0.616	1,423,000	-1.249	45,950	-0.109
8	2015/04/02	2,029.07	0.031	1,434,000	0.773	46,300	0.762
9	2015/04/03	2,045.42	0.806	1,434,000	0.000	46,650	0.756
10	2015/04/06	2,046.43	0.049	1,470,000	2.510	45,750	-1.929
11	2015/04/07	2,047.03	0.029	1,462,000	-0.544	45,350	-0.874
12	2015/04/08	2,059.26	0.597	1,479,000	1.163	43,700	-3.638
13	2015/04/09	2,058.87	-0.019	1,483,000	0.270	43,100	-1.373
14	2015/04/10	2,087.76	1.403	1,490,000	0.472	43,650	1.276
15	2015/04/13	2,098.92	0.535	1,479,000	-0.738	43,300	-0.802
16	2015/04/14	2,111.72	0.610	1,474,000	-0.338	43,900	1.386
17	2015/04/15	2,119.96	0.390	1,446,000	-1.900	43,800	-0.228
18	2015/04/16	2,139.90	0.941	1,477,000	2.144	46,000	5.023
19	2015/04/17	2,143.50	0.168	1,450,000	-1.828	47,550	3.370
20	2015/04/20	2,146.71	0.150	1,430,000	-1.379	46,250	-2.734
21	2015/04/21	2,144.79	-0.089	1,444,000	0.979	46,050	-0.432
22	2015/04/22	2,143.89	-0.042	1,461,000	1.177	45,500	-1.194
23	2015/04/23	2,173.41	1.377	1,451,000	-0.684	45,800	0.659
24	2015/04/24	2,159.80	-0.626	1,410,000	-2.826	46,850	2.293
25	2015/04/27	2,157.54	-0.105	1,395,000	-1.064	47,000	0.320
26	평균	AVERAGE	0.240	AVERAGE	-0.259	AVERAGE	0.159
27	표준편차	STDEV	0.590	STDEV	1.574	STDEV	1.943


6.000
4.000
2.000
0.000
-2.000
-4.000
-6.000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
코스피 수익률
S전자 수익률
H전기 수익률

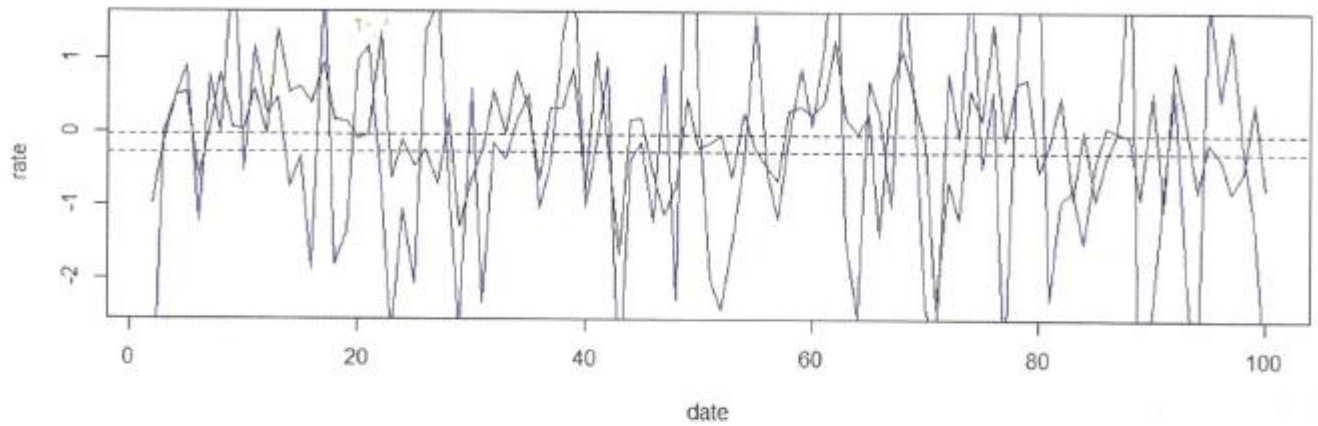


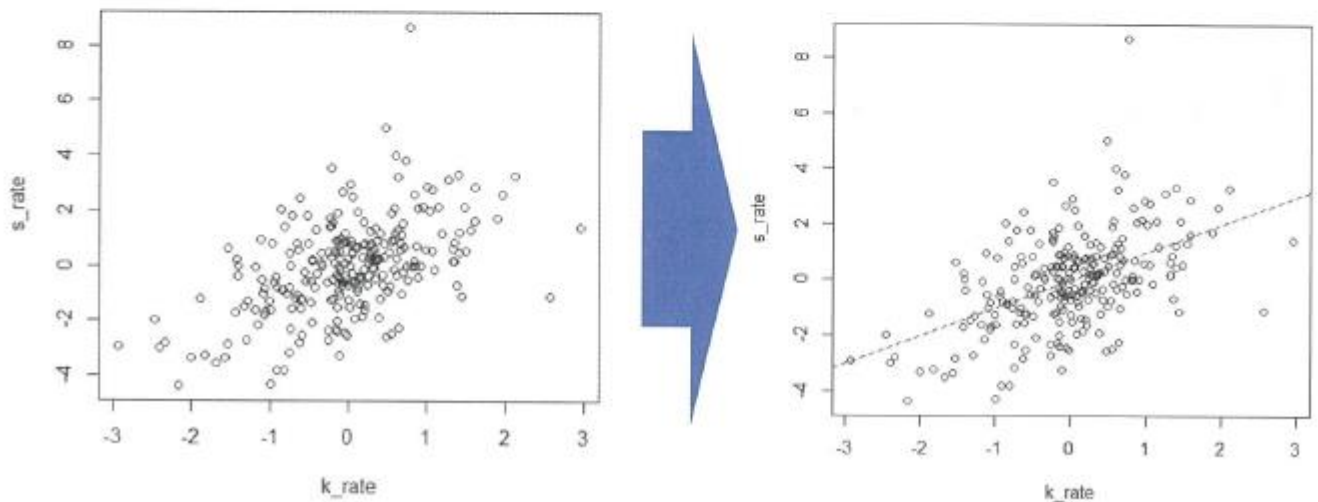
	A	B	C
1	date	kospi	k_rate
2	2015-03-25	2,042.81	
3	2015-03-26	2,022.56	-0.991
4	2015-03-27	2,019.80	-0.136
5	2015-03-30	2,030.04	0.507
6	2015-03-31	2,041.03	0.541
7	2015-04-01	2,028.45	-0.616
8	2015-04-02	2,029.07	0.031
9	2015-04-03	2,045.42	0.806
10	2015-04-06	2,046.43	0.049
11	2015-04-07	2,047.03	0.029
12	2015-04-08	2,059.26	0.597
13	2015-04-09	2,058.87	-0.019
14	2015-04-10	2,087.76	1.403

	A	B	C
1	date	h_price	h_rate
2	2015-03-25	45,500	
3	2015-03-26	45,900	0.879
4	2015-03-27	45,000	-1.961
5	2015-03-30	45,600	1.333
6	2015-03-31	46,000	0.877
7	2015-04-01	45,950	-0.109
8	2015-04-02	46,300	0.762
9	2015-04-03	46,650	0.756
10	2015-04-06	45,750	-1.929
11	2015-04-07	45,350	-0.874
12	2015-04-08	43,700	-3.638
13	2015-04-09	43,100	-1.373
14	2015-04-10	43,650	1.276

	A	B	C
1	date	s_price	s_rate
2	2015-03-25	1,485,000	
3	2015-03-26	1,421,000	-4.310
4	2015-03-27	1,421,000	0.000
5	2015-03-30	1,428,000	0.493
6	2015-03-31	1,441,000	0.910
7	2015-04-01	1,423,000	-1.249
8	2015-04-02	1,434,000	0.773
9	2015-04-03	1,434,000	0.000
10	2015-04-06	1,470,000	2.510
11	2015-04-07	1,462,000	-0.544
12	2015-04-08	1,479,000	1.163
13	2015-04-09	1,483,000	0.270
14	2015-04-10	1,490,000	0.472

```
> k_index <- read.csv("K_index.csv" , header = T , stringsAsFactors = F)
> s_stock <- read.csv("S_stock.csv" , header = T , stringsAsFactors = F)
> h_stock <- read.csv("H_stock.csv" , header = T , stringsAsFactors = F)
> all_data <- merge(merge(k_index , s_stock) , h_stock)
> head(all_data)
        date   kospi k_rate s_price s_rate h_price h_rate
1 2015-03-25 2042.81     NA 1485000     NA   45500     NA
2 2015-03-26 2022.56 -0.991 1421000 -4.310   45900  0.879
3 2015-03-27 2019.80 -0.136 1421000  0.000   45000 -1.961
4 2015-03-30 2030.04  0.507 1428000  0.493   45600  1.333
5 2015-03-31 2041.03  0.541 1441000  0.910   46000  0.877
6 2015-04-01 2028.45 -0.616 1423000 -1.249   45950 -0.109
> str(all_data)
'data.frame':   249 obs. of  7 variables:
 $ date   : chr  "2015-03-25" "2015-03-26" "2015-03-27" "2015-03-30" ...
 $ kospi  : num  2043 2023 2020 2030 2041 ...
 $ k_rate : num  NA -0.991 -0.136 0.507 0.541 -0.616 0.031 0.806 0.049 0.029 ...
 $ s_price: num  1485000 1421000 1421000 1428000 1441000 ...
 $ s_rate : num  NA -4.31 0 0.493 0.91 ...
 $ h_price: num  45500 45900 45000 45600 46000 ...
 $ h_rate : num  NA 0.879 -1.961 1.333 0.877 ...
```


rate
1
0
-1
-2
0
20
40
60
80
100
date


s_rate
8
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
k_rate
s_rate
8
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
k_rate

General linear model : $y_i = w^T * x_i + w_o = \Sigma_{j=o}^{N}(w^j x_i^j)$

Cost function = $\Sigma_i(y - y_i)^2 = \Sigma_i(y - \Sigma_{j=0}^{N}(w^j x_i^j))^2 = \Sigma_i e_i^2$

```
> s_lm <- lm(s_rate ~ k_rate , data = all_data)
> h_lm <- lm(h_rate ~ k_rate , data = all_data)
```

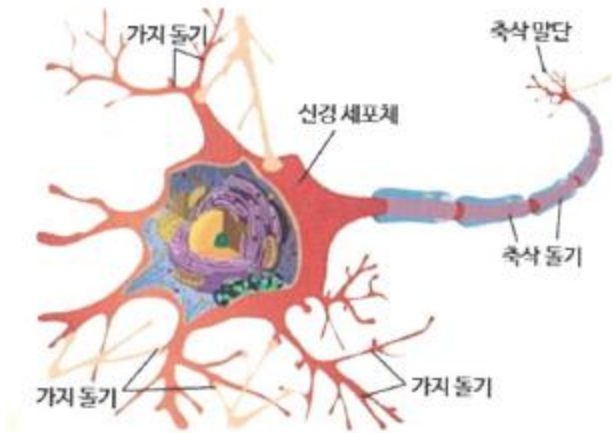
```
> summary(s_lm)

Call:
lm(formula = s_rate ~ k_rate, data = all_data)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6504 -0.8373 -0.0862  0.7410  7.9590

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.03500    0.09273  -0.377    0.706
k_rate       1.00133    0.10647   9.404   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.46 on 246 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.2644,    Adjusted R-squared:  0.2615
F-statistic: 88.44 on 1 and 246 DF,  p-value: < 2.2e-16
```

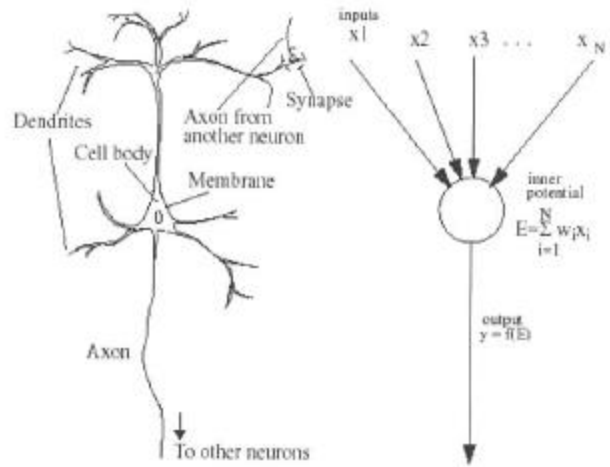
```
> summary(h_lm)

Call:
lm(formula = h_rate ~ k_rate, data = all_data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.1433 -1.0825 -0.0597  0.8846  4.7971

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1263     0.1021   1.237    0.217
k_rate        0.6348     0.1173   5.414 1.47e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.608 on 246 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.1065,    Adjusted R-squared:  0.1028
F-statistic: 29.31 on 1 and 246 DF,  p-value: 1.466e-07
```

06

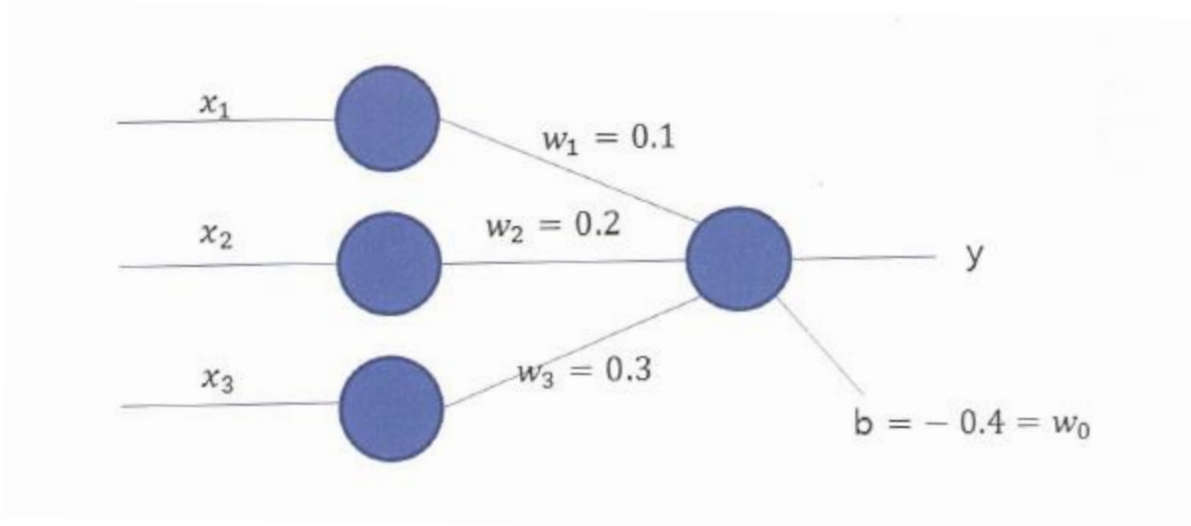
CHAPTER

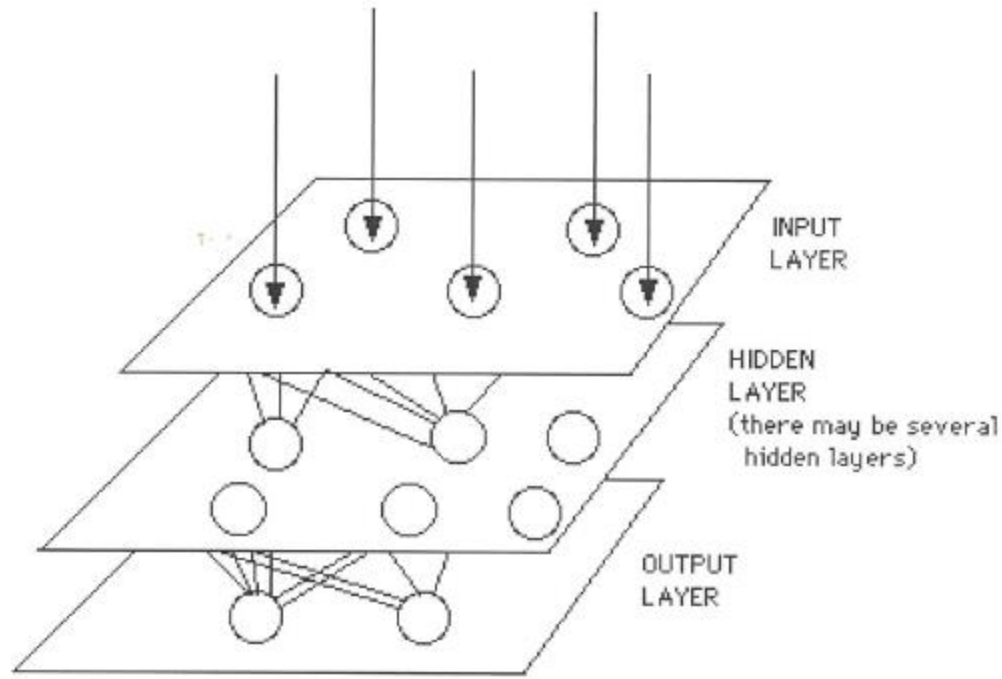
Neural Network


가지 돌기
축삭 말단
신경 세포체
축삭 돌기
가지 돌기
가지 돌기


Dendrites
Axon from another neuron
Synapse
Cell body
Membrane
Axon
To other neurons

inputs
x1
x2
x3
x N
inner potential
output

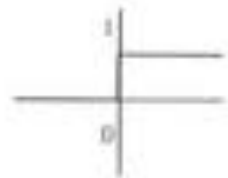

x_1
x_2
x_3
$w_1 = 0.1$
$w_2 = 0.2$
$w_3 = 0.3$
y
b = − 0.4 = w_0


INPUT
LAYER
HIDDEN
LAYER
(there may be several
hidden layers)
OUTPUT
LAYER

Activation functions

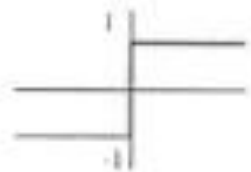
- **Step** function

$$step_t(x) = \begin{cases} 1 & x > t \\ 0 & otherwise \end{cases}$$



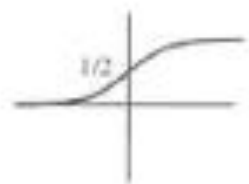
- **Sign** function

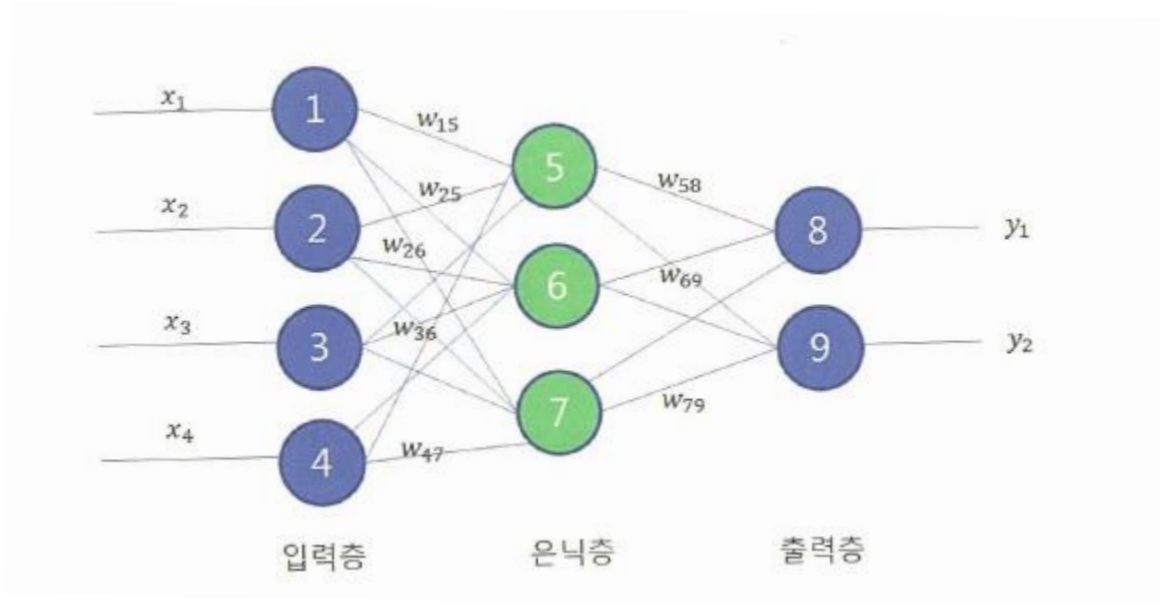
$$sign(x) = \begin{cases} +1 & x \geq 0 \\ -1 & altrimenti \end{cases}$$

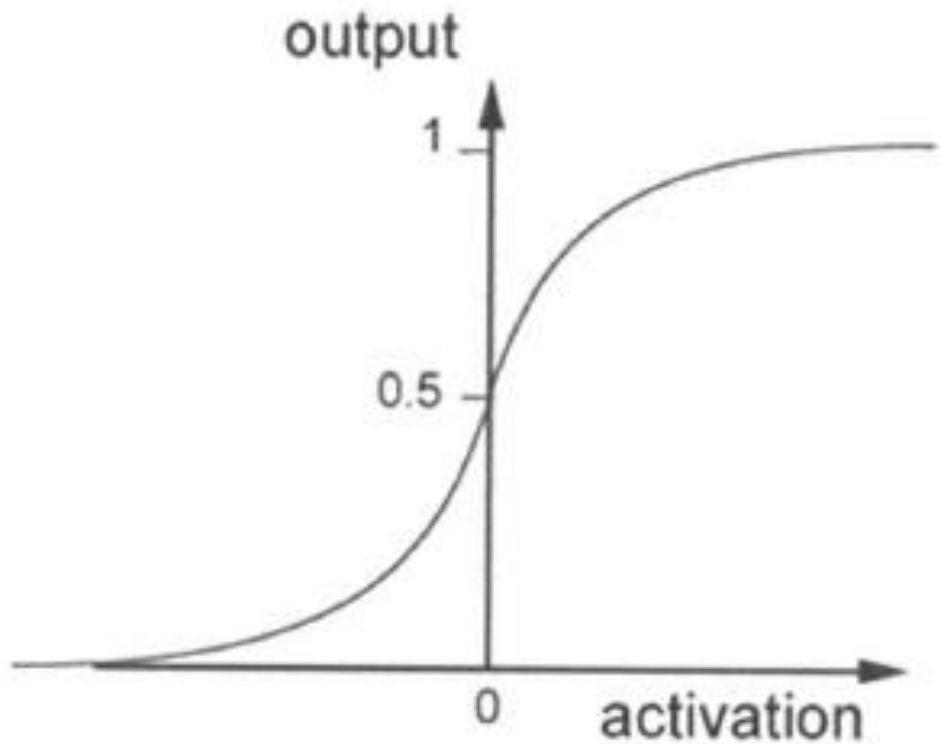


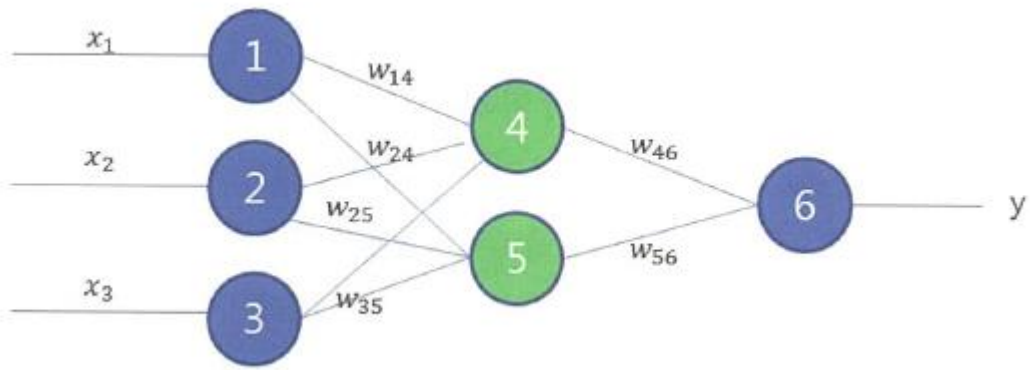
- **Sigmoid** function

$$sigmoide(x) = \frac{1}{1+e^{-x}}$$




x_1
x_2
x_3
x_4
1
2
3
4
5
6
7
8
9
w_{15}
w_{25}
w_{26}
w_{36}
w_{47}
w_{58}
w_{69}
w_{79}
y_1
y_2
입력층
은닉층
출력층


output
1
0.5
0
activation


x_1
x_2
x_3
1
2
3
4
5
6
w_{14}
w_{24}
w_{25}
w_{35}
w_{46}
w_{56}
y

S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
ans18	ans19	ans20	ans21	ans22	ans23	ans24	ans25	ans26	ans27	ans28	ans29	ans30	accident
3	4	3	5	3	4	4	4	3	2	3	4	1	suicide
1	3	4	4	3	2	5	1	5	1	2	3	5	general
1	3	4	1	1	3	4	1	4	3	2	4	4	general
5	5	4	1	4	1	1	1	3	4	4	5	1	violence
2	4	2	4	3	3	1	1	1	3	3	3	1	general
1	4	3	2	3	1	2	4	4	5	2	3	2	general
2	4	1	4	5	4	1	2	5	3	3	1	2	suicide
2	2	4	2	4	2	3	1	3	2	3	5	3	general
3	1	4	1	2	2	3	1	4	4	2	5	4	general
1	4	3	5	1	5	4	4	4	5	3	3	1	suicide
1	5	4	5	2	3	2	4	5	1	2	3	4	general
2	1	2	4	2	2	5	2	1	2	2	4	2	general
1	5	5	4	2	1	1	3	4	5	2	3	2	general
1	3	4	3	4	2	1	2	2	2	3	1	3	general
5	2	5	3	2	3	1	2	1	4	5	5	3	violence
2	2	1	2	1	3	3	3	4	3	1	4	4	general
3	5	3	1	3	2	4	5	3	3	2	4	5	general
1	3	3	4	3	2	3	4	1	2	3	4	5	general
1	2	1	4	2	2	3	3	5	5	2	4	3	general
1	1	4	2	4	3	4	1	2	5	2	4	2	general

```
> prob <- read.csv("problem.csv" , header=T , stringsAsFactors = F)
> head(prob)
  ans1 ans2 ans3 ans4 ans5 ans6 ans7 ans8 ans9 ans10 ans11 ans12 ans13 ans14 ans15 ans16 ans17 ans18 ans19 ans20
1    5    1    5    4    1    3    3    1    4     3     1     5     1     3     1     3     2     3     4     3
2    1    1    1    2    3    2    3    1    2     2     1     1     3     1     5     4     2     1     3     4
3    2    3    2    1    1    1    3    2    4     5     4     3     1     2     1     4     5     1     3     4
4    3    2    1    4    2    3    3    1    4     3     5     1     1     2     4     4     1     5     5     4
5    3    3    3    3    2    2    3    2    2     3     4     1     1     5     5     3     4     2     4     2
6    2    1    3    1    1    3    1    3    2     5     5     1     1     3     5     5     4     1     4     3
  ans21 ans22 ans23 ans24 ans25 ans26 ans27 ans28 ans29 ans30 accident
1     5     3     4     4     4     3     2     3     4     1  suicide
2     4     3     2     5     1     5     1     2     3     5  general
3     1     1     3     4     1     4     3     2     4     4  general
4     1     4     1     1     1     3     4     4     5     1 violence
5     4     3     3     1     1     1     3     3     3     1  general
6     2     3     1     2     4     4     5     2     3     2  general
```

```
> for(i in 1:30) {
+   #0에서 1사이의 값으로 바꾼다.
+   prob[i] <- prob[i] * (1/5)
+
+ }
> head(prob)
```

	ans1	ans2	ans3	ans4	ans5	ans6	ans7	ans8	ans9	ans10	ans11	ans12	ans13	ans14	ans15	ans16	ans17	ans18	ans19	ans20
1	1.0	0.2	1.0	0.8	0.2	0.6	0.6	0.2	0.8	0.6	0.2	1.0	0.2	0.6	0.2	0.6	0.4	0.6	0.8	0.6
2	0.2	0.2	0.2	0.4	0.6	0.4	0.6	0.2	0.4	0.4	0.2	0.2	0.6	0.2	1.0	0.8	0.4	0.2	0.6	0.8
3	0.4	0.6	0.4	0.2	0.2	0.2	0.6	0.4	0.8	1.0	0.8	0.6	0.2	0.4	0.2	0.8	1.0	0.2	0.6	0.8
4	0.6	0.4	0.2	0.8	0.4	0.6	0.6	0.2	0.8	0.6	1.0	0.2	0.2	0.4	0.8	0.8	0.2	1.0	1.0	0.8
5	0.6	0.6	0.6	0.6	0.4	0.4	0.6	0.4	0.4	0.6	0.8	0.2	0.2	1.0	1.0	0.6	0.8	0.4	0.8	0.4
6	0.4	0.2	0.6	0.2	0.2	0.6	0.2	0.6	0.4	1.0	1.0	0.2	0.2	0.6	1.0	1.0	0.8	0.2	0.8	0.6

	ans21	ans22	ans23	ans24	ans25	ans26	ans27	ans28	ans29	ans30	accident
1	1.0	0.6	0.8	0.8	0.8	0.6	0.4	0.6	0.8	0.2	suicide
2	0.8	0.6	0.4	1.0	0.2	1.0	0.2	0.4	0.6	1.0	general
3	0.2	0.2	0.6	0.8	0.2	0.8	0.6	0.4	0.8	0.8	general
4	0.2	0.8	0.2	0.2	0.2	0.6	0.8	0.8	1.0	0.2	violence
5	0.8	0.6	0.6	0.2	0.2	0.2	0.6	0.6	0.6	0.2	general
6	0.4	0.6	0.2	0.4	0.8	0.8	1.0	0.4	0.6	0.4	general

```
> #정규화 함수
> normalize <- function(x) {
+   return((x-min(x)) / diff(range(x)))
+ }
```

```
> prob$accident2 <- with(prob , ifelse(accident=="suicide" | accident=="violence" , 1 , 0))
> head(prob)
  ans1 ans2 ans3 ans4 ans5 ans6 ans7 ans8 ans9 ans10 ans11 ans12 ans13 ans14 ans15 ans16 ans17 ans18 ans19 ans20
1  1.0  0.2  1.0  0.8  0.2  0.6  0.6  0.2  0.8   0.6   0.2   1.0   0.2   0.6   0.2   0.6   0.4   0.6   0.8   0.6
2  0.2  0.2  0.2  0.4  0.6  0.4  0.6  0.2  0.4   0.4   0.2   0.2   0.6   0.2   1.0   0.8   0.4   0.2   0.6   0.8
3  0.4  0.6  0.4  0.2  0.2  0.2  0.6  0.4  0.8   1.0   0.8   0.6   0.2   0.4   0.2   0.8   1.0   0.2   0.6   0.8
4  0.6  0.4  0.2  0.8  0.4  0.6  0.6  0.2  0.8   0.6   1.0   0.2   0.2   0.4   0.8   0.8   0.2   1.0   1.0   0.8
5  0.6  0.6  0.6  0.6  0.4  0.4  0.6  0.4  0.4   0.6   0.8   0.2   0.2   1.0   1.0   0.6   0.8   0.4   0.8   0.4
6  0.4  0.2  0.6  0.2  0.2  0.6  0.2  0.6  0.4   1.0   1.0   0.2   0.2   0.6   1.0   1.0   0.8   0.2   0.8   0.6
  ans21 ans22 ans23 ans24 ans25 ans26 ans27 ans28 ans29 ans30 accident accident2
1   1.0   0.6   0.8   0.8   0.8   0.6   0.4   0.6   0.8   0.2  suicide         1
2   0.8   0.6   0.4   1.0   0.2   1.0   0.2   0.4   0.6   1.0  general         0
3   0.2   0.2   0.6   0.8   0.2   0.8   0.6   0.4   0.8   0.8  general         0
4   0.2   0.8   0.2   0.2   0.2   0.6   0.8   0.8   1.0   0.2 violence         1
5   0.8   0.6   0.6   0.2   0.2   0.2   0.6   0.6   0.6   0.2  general         0
6   0.4   0.6   0.2   0.4   0.8   0.8   1.0   0.4   0.6   0.4  general         0
```

```
> library(nnet)
> prob <- prob[-31]
> m1 <- nnet(accident2 ~ . , data = prob , size=10)
# weights:  321
initial  value 9.405984
iter  10 value 2.901986
iter  20 value 0.861853
final  value 0.000082
converged
> r1 <- predict(m1 , prob)
> head(r1)
        [,1]
1 1.0000000
2 0.0000000
3 0.0000000
4 0.9999949
5 0.0000000
6 0.0000000
```

```
> cbind(prob$accident2 , r1>0.5)
   [,1] [,2]
1     1    1
2     0    0
3     0    0
4     1    1
5     0    0
6     0    0
7     1    1
8     0    0
9     0    0
10    1    1
11    0    0
12    0    0
13    0    0
14    0    0
15    1    1
16    0    0
17    0    0
18    0    0
19    0    0
20    0    0
21    1    1
```

```
> sum(as.numeric(r1>0.5) != prob$accident2)
[1] 0
```

```
> #같은 방법(다른 패키지)
> library(neuralnet)
필요한 패키지를 로딩중입니다: grid
필요한 패키지를 로딩중입니다: MASS
Warning message:
패키지 'neuralnet'는 R 버전 3.0.3에서 작성되었습니다
> xnam <- paste0("ans", 1:30)
> fmla <- as.formula(paste("accident2 ~ ", paste(xnam, collapse= "+")))
> m2 <- neuralnet(fmla , data = prob , hidden = 10)
> plot(m2)
```