데이터사이언스 과제1

```
> #1. 변수와 벡터 생성
> #(1)프린트
> a <- 1
> x <- 3
> a
[1] 1
> print(a)
[1] 1
> print(x)
[1] 3
>
> #(2)c()
> x <- c("fee", "fie", "foe", "fum")
> print(x)
[1] "fee" "fie" "foe" "fum"
> c("Everyone", "loves", "stats.")
[1] "Everyone" "loves"    "stats."
> c(1,1,2,3,5,8,13,21)
[1]  1  1  2  3  5  8 13 21
> c(1*pi, 2*pi, 3*pi, 4*pi)
[1]  3.141593  6.283185  9.424778 12.566371
> c(TRUE, TRUE, FALSE, TRUE)
[1]  TRUE  TRUE FALSE  TRUE
> v1 <- c(1,2,3)
> v2 <- c(4,5,6)
> v3 <- c("A", "B", "C")
>
> #2. 수열
> 1:5
[1] 1 2 3 4 5
> b <- 2:10
> b
[1]  2  3  4  5  6  7  8  9 10
> 10:19
 [1] 10 11 12 13 14 15 16 17 18 19
> 9:0
```

```
 [1] 9 8 7 6 5 4 3 2 1 0
> e <- 10:2
> e
[1] 10  9  8  7  6  5  4  3  2
> seq(from=0, to=20, by=2)
 [1]  0  2  4  6  8 10 12 14 16 18 20
> seq(from=0, to=20, length.out=5)
[1]  0  5 10 15 20
> seq(from=1.0, to=2.0, length.out=5)
[1] 1.00 1.25 1.50 1.75 2.00
> seq(0, 10, by=1)
 [1]  0  1  2  3  4  5  6  7  8  9 10
> seq(0, 10, length=20)
 [1]  0.0000000  0.5263158  1.0526316  1.5789474  2.1052632  2.6315789
 [7]  3.1578947  3.6842105  4.2105263  4.7368421  5.2631579  5.7894737
[13]  6.3157895  6.8421053  7.3684211  7.8947368  8.4210526  8.9473684
[19]  9.4736842 10.0000000
> n <- 0
> 1:n
[1] 1 0
> rep(1, times=5)
[1] 1 1 1 1 1
> rep(1:2, each=2)
[1] 1 1 2 2
> c <- 1:5
> c
[1] 1 2 3 4 5
> rep(c, 5)
 [1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
> rep(c, each=5)
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5
>
> #3. 데이터 유형과 객체
> #(1)Numeric
> a <- 3
> a
[1] 3
>
> #(2)Character
> b <- "Charcter"
> b
```

```
[1] "Charcter"
>
> #(3)paste("붙일 내용", sep="")
> A <- c("a", "b", "c")
> A
[1] "a" "b" "c"
> paste("a", "b", sep="")
[1] "ab"
> paste(A, c("d", "e"))
[1] "a d" "b e" "c d"
> f <- paste(A, 10)
> f
[1] "a 10" "b 10" "c 10"
> paste(A, 10, sep="")
[1] "a10" "b10" "c10"
> paste(A, 1:10, sep="")
 [1] "a1"  "b2"  "c3"  "a4"  "b5"  "c6"  "a7"  "b8"  "c9"  "a10"
> paste("Everybody", "loves", "cats.")
[1] "Everybody loves cats."
> paste("Everybody", "loves", "cats.", sep="-")
[1] "Everybody-loves-cats."
> paste("Everybody", "loves", "cats.", sep="")
[1] "Everybodylovescats."
>
> #(4)Substr(문자열, 시작, 끝)
> ss <- c("Moe", "Larry", "Curly")
> substr("BigDataAnalysis", 1, 4)
[1] "BigD"
> substr(ss, 1, 3)
[1] "Moe" "Lar" "Cur"
>
> #(5)논리값
> c <- TRUE
> c
[1] TRUE
> d <- T
> d
[1] TRUE
> e <- FALSE
> e
[1] FALSE
```

```
> f <- F
> f
[1] FALSE
> a <- 3
> a == pi
[1] FALSE
> a != pi
[1] TRUE
> a < pi
[1] TRUE
> a > pi
[1] FALSE
> a <= pi
[1] TRUE
> a >= pi
[1] FALSE
> a = pi
> a == pi
[1] TRUE
>
> #(6)Matrix
> theData <- c(1.1, 1.2, 2.1, 2.2, 3.1, 3.2)
> mat <- matrix(theData, 2, 3)
> mat
     [,1] [,2] [,3]
[1,]  1.1  2.1  3.1
[2,]  1.2  2.2  3.2
> dim(mat)
[1] 2 3
> mat
     [,1] [,2] [,3]
[1,]  1.1  2.1  3.1
[2,]  1.2  2.2  3.2
> t(mat)
     [,1] [,2]
[1,]  1.1  1.2
[2,]  2.1  2.2
[3,]  3.1  3.2
> mat%*%t(mat)
      [,1]  [,2]
[1,] 15.23 15.86
```

```
[2,] 15.86 16.52
> diag(mat)
[1] 1.1 2.2
> mat
     [,1] [,2] [,3]
[1,]  1.1  2.1  3.1
[2,]  1.2  2.2  3.2
> colnames(mat) <- c("IBM", "MSFT", "GOOG")
> rownames(mat) <- c("IBM", "MSFT")
> mat
     IBM MSFT GOOG
IBM  1.1  2.1  3.1
MSFT 1.2  2.2  3.2
> mat
     IBM MSFT GOOG
IBM  1.1  2.1  3.1
MSFT 1.2  2.2  3.2
> mat[1, ] #첫째 행
 IBM MSFT GOOG
 1.1  2.1  3.1
> mat[ ,3] #셋째 열
 IBM MSFT
 3.1  3.2
> A <- matrix(0,4,5)
> A
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    0    0
[2,]    0    0    0    0    0
[3,]    0    0    0    0    0
[4,]    0    0    0    0    0
> A <- matrix(1:20, 4, 5)
> A
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> A[c(1,4), c(2,3)]
     [,1] [,2]
[1,]    5    9
[2,]    8   12
```

```
> A[c(1,4), c(2,3)] <- 1
> A
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1    1   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    1    1   16   20
> A + 1
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    2    2   14   18
[2,]    3    7   11   15   19
[3,]    4    8   12   16   20
[4,]    5    2    2   17   21
>
> #(7)list
> lst <- list(3.14, "Moe", c(1, 1, 2, 3), mean)
> lst
[[1]]
[1] 3.14

[[2]]
[1] "Moe"

[[3]]
[1] 1 1 2 3

[[4]]
function (x, ...)
UseMethod("mean")
<bytecode: 0x000000001a123268>
<environment: namespace:base>

> a <- 1:10
> b <- matrix(1:10, 2, 5)
> c <- c("name1", "name2")
> alst <- list(x=a, y=b, z=c)
> alst
$x
 [1]  1  2  3  4  5  6  7  8  9 10

$y
```

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

$z
[1] "name1" "name2"

> alst$x
 [1]  1  2  3  4  5  6  7  8  9 10
> blst <- list(d=2:10*10)
> blst
$d
[1]  20  30  40  50  60  70  80  90 100

> alst$x
 [1]  1  2  3  4  5  6  7  8  9 10
> alst[[1]]
 [1]  1  2  3  4  5  6  7  8  9 10
> alst[[1]][2]
[1] 2
> alst[[2]]
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
> ablst <- c(alst, blst)
> ablst
$x
 [1]  1  2  3  4  5  6  7  8  9 10

$y
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10

$z
[1] "name1" "name2"

$d
[1]  20  30  40  50  60  70  80  90 100

> score1 <- list(10, 20, 30, 40, 50)
```

```
> score2 <- list(c("a", "b"))
> score1[score1 > 40]
[[1]]
[1] 50

> score1 > 40
[1] FALSE FALSE FALSE FALSE  TRUE
> score1[c(FALSE, FALSE, FALSE, FALSE, TRUE)]
[[1]]
[1] 50

> score12 <- list(score1, score2)
> score12
[[1]]
[[1]][[1]]
[1] 10

[[1]][[2]]
[1] 20

[[1]][[3]]
[1] 30

[[1]][[4]]
[1] 40

[[1]][[5]]
[1] 50


[[2]]
[[2]][[1]]
[1] "a" "b"


> score12[1]
[[1]]
[[1]][[1]]
[1] 10

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

```
[1] 20

[[1]][[3]]
[1] 30

[[1]][[4]]
[1] 40

[[1]][[5]]
[1] 50


> score12[[2]]
[[1]]
[1] "a" "b"

> score12[[2]][1]
[[1]]
[1] "a" "b"

> score12[[1]][1]
[[1]]
[1] 10

> score12[[1]][2]
[[1]]
[1] 20

> unlist(score1)
[1] 10 20 30 40 50
> unlist(score2)
[1] "a" "b"
> unlist(score12)
[1] "10" "20" "30" "40" "50" "a"   "b"
>
> #(8)데이터프레임
> a <- c(1, 2, 4, 6, 3, 4)
> b <- c(6, 4, 2, 4, 3.2, 4)
> c <- c(7, 6, 4, 2, 5, 6)
> d <- c(2, 4, 3, 1, 5, 6)
> e <- data.frame(a, b, c, d)
```

```
> e
  a   b c d
1 1 6.0 7 2
2 2 4.0 6 4
3 4 2.0 4 3
4 6 4.0 2 1
5 3 3.2 5 5
6 4 4.0 6 6
> data(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
>     newRow    <-    data.frame(Sepal.Length=3.0,    Sepal.Width=3.2,
Petal.Length=1.6, Petal.Width=0.3, Species="newsetosa")
> newRow
  Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1            3         3.2          1.6         0.3 newsetosa
> iris <- rbind(iris, newRow)
> iris
   Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1           5.1         3.5          1.4         0.2    setosa
2           4.9         3.0          1.4         0.2    setosa
3           4.7         3.2          1.3         0.2    setosa
4           4.6         3.1          1.5         0.2    setosa
5           5.0         3.6          1.4         0.2    setosa
6           5.4         3.9          1.7         0.4    setosa
7           4.6         3.4          1.4         0.3    setosa
8           5.0         3.4          1.5         0.2    setosa
9           4.4         2.9          1.4         0.2    setosa
10          4.9         3.1          1.5         0.1    setosa
11          5.4         3.7          1.5         0.2    setosa
12          4.8         3.4          1.6         0.2    setosa
13          4.8         3.0          1.4         0.1    setosa
14          4.3         3.0          1.1         0.1    setosa
15          5.8         4.0          1.2         0.2    setosa
16          5.7         4.4          1.5         0.4    setosa
```

17	5.4	3.9	1.3	0.4	setosa	58	4.9	2.4	3.3	1.0 versicolor
18	5.1	3.5	1.4	0.3	setosa	59	6.6	2.9	4.6	1.3 versicolor
19	5.7	3.8	1.7	0.3	setosa	60	5.2	2.7	3.9	1.4 versicolor
20	5.1	3.8	1.5	0.3	setosa	61	5.0	2.0	3.5	1.0 versicolor
21	5.4	3.4	1.7	0.2	setosa	62	5.9	3.0	4.2	1.5 versicolor
22	5.1	3.7	1.5	0.4	setosa	63	6.0	2.2	4.0	1.0 versicolor
23	4.6	3.6	1.0	0.2	setosa	64	6.1	2.9	4.7	1.4 versicolor
24	5.1	3.3	1.7	0.5	setosa	65	5.6	2.9	3.6	1.3 versicolor
25	4.8	3.4	1.9	0.2	setosa	66	6.7	3.1	4.4	1.4 versicolor
26	5.0	3.0	1.6	0.2	setosa	67	5.6	3.0	4.5	1.5 versicolor
27	5.0	3.4	1.6	0.4	setosa	68	5.8	2.7	4.1	1.0 versicolor
28	5.2	3.5	1.5	0.2	setosa	69	6.2	2.2	4.5	1.5 versicolor
29	5.2	3.4	1.4	0.2	setosa	70	5.6	2.5	3.9	1.1 versicolor
30	4.7	3.2	1.6	0.2	setosa	71	5.9	3.2	4.8	1.8 versicolor
31	4.8	3.1	1.6	0.2	setosa	72	6.1	2.8	4.0	1.3 versicolor
32	5.4	3.4	1.5	0.4	setosa	73	6.3	2.5	4.9	1.5 versicolor
33	5.2	4.1	1.5	0.1	setosa	74	6.1	2.8	4.7	1.2 versicolor
34	5.5	4.2	1.4	0.2	setosa	75	6.4	2.9	4.3	1.3 versicolor
35	4.9	3.1	1.5	0.2	setosa	76	6.6	3.0	4.4	1.4 versicolor
36	5.0	3.2	1.2	0.2	setosa	77	6.8	2.8	4.8	1.4 versicolor
37	5.5	3.5	1.3	0.2	setosa	78	6.7	3.0	5.0	1.7 versicolor
38	4.9	3.6	1.4	0.1	setosa	79	6.0	2.9	4.5	1.5 versicolor
39	4.4	3.0	1.3	0.2	setosa	80	5.7	2.6	3.5	1.0 versicolor
40	5.1	3.4	1.5	0.2	setosa	81	5.5	2.4	3.8	1.1 versicolor
41	5.0	3.5	1.3	0.3	setosa	82	5.5	2.4	3.7	1.0 versicolor
42	4.5	2.3	1.3	0.3	setosa	83	5.8	2.7	3.9	1.2 versicolor
43	4.4	3.2	1.3	0.2	setosa	84	6.0	2.7	5.1	1.6 versicolor
44	5.0	3.5	1.6	0.6	setosa	85	5.4	3.0	4.5	1.5 versicolor
45	5.1	3.8	1.9	0.4	setosa	86	6.0	3.4	4.5	1.6 versicolor
46	4.8	3.0	1.4	0.3	setosa	87	6.7	3.1	4.7	1.5 versicolor
47	5.1	3.8	1.6	0.2	setosa	88	6.3	2.3	4.4	1.3 versicolor
48	4.6	3.2	1.4	0.2	setosa	89	5.6	3.0	4.1	1.3 versicolor
49	5.3	3.7	1.5	0.2	setosa	90	5.5	2.5	4.0	1.3 versicolor
50	5.0	3.3	1.4	0.2	setosa	91	5.5	2.6	4.4	1.2 versicolor
51	7.0	3.2	4.7	1.4 versicolor		92	6.1	3.0	4.6	1.4 versicolor
52	6.4	3.2	4.5	1.5 versicolor		93	5.8	2.6	4.0	1.2 versicolor
53	6.9	3.1	4.9	1.5 versicolor		94	5.0	2.3	3.3	1.0 versicolor
54	5.5	2.3	4.0	1.3 versicolor		95	5.6	2.7	4.2	1.3 versicolor
55	6.5	2.8	4.6	1.5 versicolor		96	5.7	3.0	4.2	1.2 versicolor
56	5.7	2.8	4.5	1.3 versicolor		97	5.7	2.9	4.2	1.3 versicolor
57	6.3	3.3	4.7	1.6 versicolor		98	6.2	2.9	4.3	1.3 versicolor

99	5.1	2.5	3.0	1.1	versicolor
100	5.7	2.8	4.1	1.3	versicolor
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
103	7.1	3.0	5.9	2.1	virginica
104	6.3	2.9	5.6	1.8	virginica
105	6.5	3.0	5.8	2.2	virginica
106	7.6	3.0	6.6	2.1	virginica
107	4.9	2.5	4.5	1.7	virginica
108	7.3	2.9	6.3	1.8	virginica
109	6.7	2.5	5.8	1.8	virginica
110	7.2	3.6	6.1	2.5	virginica
111	6.5	3.2	5.1	2.0	virginica
112	6.4	2.7	5.3	1.9	virginica
113	6.8	3.0	5.5	2.1	virginica
114	5.7	2.5	5.0	2.0	virginica
115	5.8	2.8	5.1	2.4	virginica
116	6.4	3.2	5.3	2.3	virginica
117	6.5	3.0	5.5	1.8	virginica
118	7.7	3.8	6.7	2.2	virginica
119	7.7	2.6	6.9	2.3	virginica
120	6.0	2.2	5.0	1.5	virginica
121	6.9	3.2	5.7	2.3	virginica
122	5.6	2.8	4.9	2.0	virginica
123	7.7	2.8	6.7	2.0	virginica
124	6.3	2.7	4.9	1.8	virginica
125	6.7	3.3	5.7	2.1	virginica
126	7.2	3.2	6.0	1.8	virginica
127	6.2	2.8	4.8	1.8	virginica
128	6.1	3.0	4.9	1.8	virginica
129	6.4	2.8	5.6	2.1	virginica
130	7.2	3.0	5.8	1.6	virginica
131	7.4	2.8	6.1	1.9	virginica
132	7.9	3.8	6.4	2.0	virginica
133	6.4	2.8	5.6	2.2	virginica
134	6.3	2.8	5.1	1.5	virginica
135	6.1	2.6	5.6	1.4	virginica
136	7.7	3.0	6.1	2.3	virginica
137	6.3	3.4	5.6	2.4	virginica
138	6.4	3.1	5.5	1.8	virginica
139	6.0	3.0	4.8	1.8	virginica
140	6.9	3.1	5.4	2.1	virginica
141	6.7	3.1	5.6	2.4	virginica
142	6.9	3.1	5.1	2.3	virginica
143	5.8	2.7	5.1	1.9	virginica
144	6.8	3.2	5.9	2.3	virginica
145	6.7	3.3	5.7	2.5	virginica
146	6.7	3.0	5.2	2.3	virginica
147	6.3	2.5	5.0	1.9	virginica
148	6.5	3.0	5.2	2.0	virginica
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica
151	3.0	3.2	1.6	0.3	newsetosa

```
> dim(iris)
[1] 151   5
> newcol <- 1:151
> cbind(iris, newcol)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	newcol
1	5.1	3.5	1.4	0.2	setosa	1
2	4.9	3.0	1.4	0.2	setosa	2
3	4.7	3.2	1.3	0.2	setosa	3
4	4.6	3.1	1.5	0.2	setosa	4
5	5.0	3.6	1.4	0.2	setosa	5
6	5.4	3.9	1.7	0.4	setosa	6
7	4.6	3.4	1.4	0.3	setosa	7
8	5.0	3.4	1.5	0.2	setosa	8
9	4.4	2.9	1.4	0.2	setosa	9
10	4.9	3.1	1.5	0.1	setosa	10
11	5.4	3.7	1.5	0.2	setosa	11
12	4.8	3.4	1.6	0.2	setosa	12
13	4.8	3.0	1.4	0.1	setosa	13
14	4.3	3.0	1.1	0.1	setosa	14
15	5.8	4.0	1.2	0.2	setosa	15
16	5.7	4.4	1.5	0.4	setosa	16
17	5.4	3.9	1.3	0.4	setosa	17
18	5.1	3.5	1.4	0.3	setosa	18
19	5.7	3.8	1.7	0.3	setosa	19
20	5.1	3.8	1.5	0.3	setosa	20
21	5.4	3.4	1.7	0.2	setosa	21
22	5.1	3.7	1.5	0.4	setosa	22
23	4.6	3.6	1.0	0.2	setosa	23
24	5.1	3.3	1.7	0.5	setosa	24

25	4.8	3.4	1.9	0.2	setosa	25
26	5.0	3.0	1.6	0.2	setosa	26
27	5.0	3.4	1.6	0.4	setosa	27
28	5.2	3.5	1.5	0.2	setosa	28
29	5.2	3.4	1.4	0.2	setosa	29
30	4.7	3.2	1.6	0.2	setosa	30
31	4.8	3.1	1.6	0.2	setosa	31
32	5.4	3.4	1.5	0.4	setosa	32
33	5.2	4.1	1.5	0.1	setosa	33
34	5.5	4.2	1.4	0.2	setosa	34
35	4.9	3.1	1.5	0.2	setosa	35
36	5.0	3.2	1.2	0.2	setosa	36
37	5.5	3.5	1.3	0.2	setosa	37
38	4.9	3.6	1.4	0.1	setosa	38
39	4.4	3.0	1.3	0.2	setosa	39
40	5.1	3.4	1.5	0.2	setosa	40
41	5.0	3.5	1.3	0.3	setosa	41
42	4.5	2.3	1.3	0.3	setosa	42
43	4.4	3.2	1.3	0.2	setosa	43
44	5.0	3.5	1.6	0.6	setosa	44
45	5.1	3.8	1.9	0.4	setosa	45
46	4.8	3.0	1.4	0.3	setosa	46
47	5.1	3.8	1.6	0.2	setosa	47
48	4.6	3.2	1.4	0.2	setosa	48
49	5.3	3.7	1.5	0.2	setosa	49
50	5.0	3.3	1.4	0.2	setosa	50
51	7.0	3.2	4.7	1.4	versicolor	51
52	6.4	3.2	4.5	1.5	versicolor	52
53	6.9	3.1	4.9	1.5	versicolor	53
54	5.5	2.3	4.0	1.3	versicolor	54
55	6.5	2.8	4.6	1.5	versicolor	55
56	5.7	2.8	4.5	1.3	versicolor	56
57	6.3	3.3	4.7	1.6	versicolor	57
58	4.9	2.4	3.3	1.0	versicolor	58
59	6.6	2.9	4.6	1.3	versicolor	59
60	5.2	2.7	3.9	1.4	versicolor	60
61	5.0	2.0	3.5	1.0	versicolor	61
62	5.9	3.0	4.2	1.5	versicolor	62
63	6.0	2.2	4.0	1.0	versicolor	63
64	6.1	2.9	4.7	1.4	versicolor	64
65	5.6	2.9	3.6	1.3	versicolor	65

66	6.7	3.1	4.4	1.4	versicolor	66
67	5.6	3.0	4.5	1.5	versicolor	67
68	5.8	2.7	4.1	1.0	versicolor	68
69	6.2	2.2	4.5	1.5	versicolor	69
70	5.6	2.5	3.9	1.1	versicolor	70
71	5.9	3.2	4.8	1.8	versicolor	71
72	6.1	2.8	4.0	1.3	versicolor	72
73	6.3	2.5	4.9	1.5	versicolor	73
74	6.1	2.8	4.7	1.2	versicolor	74
75	6.4	2.9	4.3	1.3	versicolor	75
76	6.6	3.0	4.4	1.4	versicolor	76
77	6.8	2.8	4.8	1.4	versicolor	77
78	6.7	3.0	5.0	1.7	versicolor	78
79	6.0	2.9	4.5	1.5	versicolor	79
80	5.7	2.6	3.5	1.0	versicolor	80
81	5.5	2.4	3.8	1.1	versicolor	81
82	5.5	2.4	3.7	1.0	versicolor	82
83	5.8	2.7	3.9	1.2	versicolor	83
84	6.0	2.7	5.1	1.6	versicolor	84
85	5.4	3.0	4.5	1.5	versicolor	85
86	6.0	3.4	4.5	1.6	versicolor	86
87	6.7	3.1	4.7	1.5	versicolor	87
88	6.3	2.3	4.4	1.3	versicolor	88
89	5.6	3.0	4.1	1.3	versicolor	89
90	5.5	2.5	4.0	1.3	versicolor	90
91	5.5	2.6	4.4	1.2	versicolor	91
92	6.1	3.0	4.6	1.4	versicolor	92
93	5.8	2.6	4.0	1.2	versicolor	93
94	5.0	2.3	3.3	1.0	versicolor	94
95	5.6	2.7	4.2	1.3	versicolor	95
96	5.7	3.0	4.2	1.2	versicolor	96
97	5.7	2.9	4.2	1.3	versicolor	97
98	6.2	2.9	4.3	1.3	versicolor	98
99	5.1	2.5	3.0	1.1	versicolor	99
100	5.7	2.8	4.1	1.3	versicolor	100
101	6.3	3.3	6.0	2.5	virginica	101
102	5.8	2.7	5.1	1.9	virginica	102
103	7.1	3.0	5.9	2.1	virginica	103
104	6.3	2.9	5.6	1.8	virginica	104
105	6.5	3.0	5.8	2.2	virginica	105
106	7.6	3.0	6.6	2.1	virginica	106

```
107      4.9      2.5      4.5      1.7  virginica  107
108      7.3      2.9      6.3      1.8  virginica  108
109      6.7      2.5      5.8      1.8  virginica  109
110      7.2      3.6      6.1      2.5  virginica  110
111      6.5      3.2      5.1      2.0  virginica  111
112      6.4      2.7      5.3      1.9  virginica  112
113      6.8      3.0      5.5      2.1  virginica  113
114      5.7      2.5      5.0      2.0  virginica  114
115      5.8      2.8      5.1      2.4  virginica  115
116      6.4      3.2      5.3      2.3  virginica  116
117      6.5      3.0      5.5      1.8  virginica  117
118      7.7      3.8      6.7      2.2  virginica  118
119      7.7      2.6      6.9      2.3  virginica  119
120      6.0      2.2      5.0      1.5  virginica  120
121      6.9      3.2      5.7      2.3  virginica  121
122      5.6      2.8      4.9      2.0  virginica  122
123      7.7      2.8      6.7      2.0  virginica  123
124      6.3      2.7      4.9      1.8  virginica  124
125      6.7      3.3      5.7      2.1  virginica  125
126      7.2      3.2      6.0      1.8  virginica  126
127      6.2      2.8      4.8      1.8  virginica  127
128      6.1      3.0      4.9      1.8  virginica  128
129      6.4      2.8      5.6      2.1  virginica  129
130      7.2      3.0      5.8      1.6  virginica  130
131      7.4      2.8      6.1      1.9  virginica  131
132      7.9      3.8      6.4      2.0  virginica  132
133      6.4      2.8      5.6      2.2  virginica  133
134      6.3      2.8      5.1      1.5  virginica  134
135      6.1      2.6      5.6      1.4  virginica  135
136      7.7      3.0      6.1      2.3  virginica  136
137      6.3      3.4      5.6      2.4  virginica  137
138      6.4      3.1      5.5      1.8  virginica  138
139      6.0      3.0      4.8      1.8  virginica  139
140      6.9      3.1      5.4      2.1  virginica  140
141      6.7      3.1      5.6      2.4  virginica  141
142      6.9      3.1      5.1      2.3  virginica  142
143      5.8      2.7      5.1      1.9  virginica  143
144      6.8      3.2      5.9      2.3  virginica  144
145      6.7      3.3      5.7      2.5  virginica  145
146      6.7      3.0      5.2      2.3  virginica  146
147      6.3      2.5      5.0      1.9  virginica  147
148      6.5      3.0      5.2      2.0  virginica  148
149      6.2      3.4      5.4      2.3  virginica  149
150      5.9      3.0      5.1      1.8  virginica  150
151      3.0      3.2      1.6      0.3  newsetosa  151
> name <- c("john", "peter", "jennifer")
> gender <- factor(c("m", "m", "f"))
> hw1 <- c(60, 60, 80)
> hw2 <- c(40, 50, 30)
> grades <- data.frame(name, gender, hw1, hw2)
> grades
      name gender hw1 hw2
1    john      m  60  40
2   peter      m  60  50
3 jennifer     f  80  30
> grades[1, 2]
[1] m
Levels: f m
> grades[ , "name"]
[1] john     peter     jennifer
Levels: jennifer john peter
> grades$name
[1] john      peter     jennifer
Levels: jennifer john peter
> grades[grades$gender == "m", ]
    name gender hw1 hw2
1  john      m  60  40
2 peter      m  60  50
> grades[ , "hw1"]
[1] 60 60 80
> data(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> subset(iris, select=Species, subset=(Petal.Length > 1.7))
       Species
25      setosa
```

45	setosa	91	versicolor
51	versicolor	92	versicolor
52	versicolor	93	versicolor
53	versicolor	94	versicolor
54	versicolor	95	versicolor
55	versicolor	96	versicolor
56	versicolor	97	versicolor
57	versicolor	98	versicolor
58	versicolor	99	versicolor
59	versicolor	100	versicolor
60	versicolor	101	virginica
61	versicolor	102	virginica
62	versicolor	103	virginica
63	versicolor	104	virginica
64	versicolor	105	virginica
65	versicolor	106	virginica
66	versicolor	107	virginica
67	versicolor	108	virginica
68	versicolor	109	virginica
69	versicolor	110	virginica
70	versicolor	111	virginica
71	versicolor	112	virginica
72	versicolor	113	virginica
73	versicolor	114	virginica
74	versicolor	115	virginica
75	versicolor	116	virginica
76	versicolor	117	virginica
77	versicolor	118	virginica
78	versicolor	119	virginica
79	versicolor	120	virginica
80	versicolor	121	virginica
81	versicolor	122	virginica
82	versicolor	123	virginica
83	versicolor	124	virginica
84	versicolor	125	virginica
85	versicolor	126	virginica
86	versicolor	127	virginica
87	versicolor	128	virginica
88	versicolor	129	virginica
89	versicolor	130	virginica
90	versicolor	131	virginica

```
132  virginica
133  virginica
134  virginica
135  virginica
136  virginica
137  virginica
138  virginica
139  virginica
140  virginica
141  virginica
142  virginica
143  virginica
144  virginica
145  virginica
146  virginica
147  virginica
148  virginica
149  virginica
150  virginica
>     subset(iris,     select=c(Sepal.Length,     Petal.Length,     Species),
subset=c(Sepal.Width==3.0 & Petal.Width==0.2))
    Sepal.Length Petal.Length Species
2            4.9          1.4  setosa
26           5.0          1.6  setosa
39           4.4          1.3  setosa
> head(with(iris, Species))
[1] setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica
> name <- c("Moe", "Larry", "Curly", "Harry")
> year.born <- c(1887, 1982, 1983, 1964)
> place.born <- c("BensonHurst", "Philadelphia", "Brooklyn", "Moscow")
> born <- data.frame(name, year.born, place.born)
> born
   name year.born    place.born
1   Moe      1887   BensonHurst
2 Larry      1982  Philadelphia
3 Curly      1983      Brooklyn
4 Harry      1964        Moscow
> name <- c("Curly", "Moe", "Larry")
> year.died <- c(1952, 1975, 1975)
> died <- data.frame(name, year.died)
```

```
> died
   name year.died
1 Curly      1952
2   Moe      1975
3 Larry      1975
> merge(born, died, by="name")
   name year.born    place.born year.died
1 Curly      1983      Brooklyn      1952
2 Larry      1982  Philadelphia      1975
3   Moe      1887   BensonHurst      1975
> data(mtcars)
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
> colnames(mtcars)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
[11] "carb"
> mtcars[1:5, c("mpg", "cyl")]
                   mpg cyl
Mazda RX4         21.0   6
Mazda RX4 Wag     21.0   6
Datsun 710        22.8   4
Hornet 4 Drive    21.4   6
Hornet Sportabout 18.7   8
> mtcars[(mtcars$gear > 3) & (mtcars$cyl > 7 | mtcars$mpg > 21),
c("mpg", "cyl", "gear")]
               mpg cyl gear
Datsun 710    22.8   4    4
Merc 240D     24.4   4    4
Merc 230      22.8   4    4
Fiat 128      32.4   4    4
Honda Civic   30.4   4    4
Toyota Corolla 33.9  4    4
Fiat X1-9     27.3   4    4
Porsche 914-2 26.0   4    5
Lotus Europa  30.4   4    5
```

```
Ford Pantera L 15.8    8    5
Maserati Bora  15.0    8    5
Volvo 142E     21.4    4    4
>
> #(9)벡터에 있는 원소 선택
> fib <- c(0, 1, 1, 2, 3, 5, 8, 13, 21, 34)
> fib
 [1]  0  1  1  2  3  5  8 13 21 34
> fib[1]
[1] 0
> fib[3]
[1] 1
> fib[1:3]
[1] 0 1 1
> fib[c(1, 2, 4, 8)]
[1]  0  1  2 13
> fib[-1]
 [1]  1  1  2  3  5  8 13 21 34
> fib[-c(1:3)]
[1]  2  3  5  8 13 21 34
> fib < 10
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
FALSE
> fib[fib < 10]
[1] 0 1 1 2 3 5 8
> fib%%2 == 0
 [1]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
TRUE
> fib[fib%%2 == 0]
[1]  0  2  8 34
> c <- 1:10
> c
 [1]  1  2  3  4  5  6  7  8  9 10
> d <- 1:5
> d[c(1, 3)]
[1] 1 3
> c[c(2, 3)]
[1] 2 3
> d[c(1:3, 5)]
[1] 1 2 3 5
> c[c > 5 & c < 10]
```

```
[1] 6 7 8 9
> c[as.logical((c > 8) + (c < 3))]
[1]  1  2  9 10
> years <- c(1960, 1964, 1976, 1994)
> names(years)<- c("Kennedy", "Johnson", "Carter", "Clinton")
> years
Kennedy Johnson  Carter Clinton
   1960    1964    1976    1994
> years["Carter"]
Carter
  1976
> years["Clinton"]
Clinton
   1994
>
> #(10)자료형 데이터 구조 변환
> as.numeric("3.14")
[1] 3.14
> as.integer(3.14)
[1] 3
> as.numeric("foo")
[1] NA
경고메시지(들):
강제형변환에 의해 생성된 NA 입니다
> as.character(101)
[1] "101"
> as.numeric(FALSE)
[1] 0
> as.numeric(TRUE)
[1] 1
>
> #(11)문자열을 날짜로 변환
> Sys.Date( ) #case-sensitive
[1] "2018-10-03"
> as.Date("2018-09-28")
[1] "2018-09-28"
> as.Date("09/28/2018")
Error in charToDate(x) : 문자열이 표준서식을 따르지 않습니다
> as.Date("09/28/2018", format="%m/%d/%Y")
[1] "2018-09-28"
>
```

```
> #(12)날짜를 문자열로 변환
> as.Date("09/28/2018", format="%m/%d/%Y")
[1] "2018-09-28"
> format(Sys.Date())
[1] "2018-10-03"
> as.character(Sys.Date())
[1] "2018-10-03"
> format(Sys.Date(), format="%m/%d/%Y")
[1] "10/03/2018"
> format(Sys.Date(), '%a')
[1] "수"
> format(Sys.Date(), '%b')
[1] "10"
> format(Sys.Date(), '%B')
[1] "10월"
> format(Sys.Date(), '%d')
[1] "03"
> format(Sys.Date(), '%m')
[1] "10"
> format(Sys.Date(), '%y')
[1] "18"
> format(Sys.Date(), '%Y')
[1] "2018"
>
> #(13)벡터의 기본 연산
> x <- c(0, 1, 2, 3, 5, 8, 13, 21, 34)
> y <- log(x+1)
> y
[1] 0.0000000 0.6931472 1.0986123 1.3862944 1.7917595 2.1972246 2.6390573
[8] 3.0910425 3.5553481
> mean(x)
[1] 9.666667
> median(x)
[1] 5
> sd(x)
[1] 11.33578
> var(x)
[1] 128.5
> cor(x, y)
[1] 0.9011302
> c <- 1:10
```

```
> c
 [1]  1  2  3  4  5  6  7  8  9 10
> 1/c
 [1] 1.0000000 0.5000000 0.3333333 0.2500000 0.2000000 0.1666667
0.1428571
 [8] 0.1250000 0.1111111 0.1000000
> c^2
 [1]   1   4   9  16  25  36  49  64  81 100
> c^2 + 1
 [1]   2   5  10  17  26  37  50  65  82 101
> log(c)
 [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595
1.9459101
 [8] 2.0794415 2.1972246 2.3025851
> sapply(c, log)
 [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595
1.9459101
 [8] 2.0794415 2.1972246 2.3025851
> c <- 1:10
> c
 [1]  1  2  3  4  5  6  7  8  9 10
> d <- (1:10)*10
> d
 [1]  10  20  30  40  50  60  70  80  90 100
> c + d
 [1]  11  22  33  44  55  66  77  88  99 110
> c * d
 [1]   10   40   90  160  250  360  490  640  810 1000
> d ^ c
 [1] 1.000000e+01 4.000000e+02 2.700000e+04 2.560000e+06 3.125000e+08
 [6] 4.665600e+10 8.235430e+12 1.677722e+15 3.874205e+17 1.000000e+20
> var(c)
[1] 9.166667
> log(c)
 [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595
1.9459101
 [8] 2.0794415 2.1972246 2.3025851
> sum((c - mean(c))^2)/(length(c)-1)
[1] 9.166667
> c <- 1:10
> c[log(c) < 2]
```

```
[1] 1 2 3 4 5 6 7
> c[log(c) < 2] <- 3
> c
 [1]  3  3  3  3  3  3  3  8  9 10
> length(c) <- 20
> c
 [1]  3  3  3  3  3  3  3  8  9 10 NA NA NA NA NA NA NA NA NA NA
> c[25] <- 1
> c
 [1]  3  3  3  3  3  3  3  8  9 10 NA NA NA NA NA NA NA NA NA NA
NA NA NA NA
[25]  1
> length(c) <- 10
> c
 [1]  3  3  3  3  3  3  3  8  9 10
>
> #4. 유용한 기타 함수
> #(1)데이터를 저장하고 출력하는 방법
> a <- c(1, 2, 3, 4, 5)
> write.csv(a, "test.csv")
> b <- read.csv("test.csv")
> save(a, file="test.Rdata")
> a <- 0
> load("test.Rdata")
> a
[1] 1 2 3 4 5
> print(a)
[1] 1 2 3 4 5
> rm(a) #또는 rm(list=c("a"))
> ls()
 [1] "A"          "ablst"      "alst"       "b"          "blst"
 [6] "born"       "c"          "d"          "died"       "e"
[11] "f"          "fib"        "gender"     "grades"     "hw1"
[16] "hw2"        "iris"       "lst"        "mat"        "mtcars"
[21] "n"          "name"       "newcol"     "newRow"     "place.born"
[26] "score1"     "score12"    "score2"     "ss"         "theData"
[31] "v1"         "v2"         "v3"         "x"          "y"
[36] "year.born"  "year.died"  "years"
> rm(list=c("a")) #rm(a)로 이미 지움
경고메시지(들):
In rm(list = c("a")) : 객체 'a'를 찾을 수 없습니다
```

```
> ls()
 [1] "A"          "ablst"      "alst"       "b"          "blst"
 [6] "born"       "c"          "d"          "died"       "e"
[11] "f"          "fib"        "gender"     "grades"     "hw1"
[16] "hw2"        "iris"       "lst"        "mat"        "mtcars"
[21] "n"          "name"       "newcol"     "newRow"     "place.born"
[26] "score1"     "score12"    "score2"     "ss"         "theData"
[31] "v1"         "v2"         "v3"         "x"          "y"
[36] "year.born"  "year.died"  "years"
> rm(list=ls(all=TRUE))
> ls()
character(0)
```