

R 프로그래밍 언어

통계처리 빅데이터에 사용
표현력이 좋다.

국민대학교 컴퓨터공학부
강 승 식

R data type: Object, Attribute

- 기본 class:
 - character, numeric(실수), integer, complex, logical
- 기본 object: vector
 - 한 가지 type의 데이터를 1개 이상 포함
 - 생성: `vector()` – type, length
 - Number: numeric object
 - 정수: `1L`, `inf`(무한대수), `NaN`(Not a Number)
 - List 클래스는 타입이 다른 데이터 허용
- 기타 클래스: `matrix`, `data.frame`, `table`

Vector와 List

- 벡터 생성

- `c()`, `vector(type, length)`, `as.*(object)`, `matrix(nrow=x, ncol=y)`
- `cbind()`, `rbind()` – vector를 binding하여 matrix 생성

```
> x <- c("hello", "world")
> y <- vector("character", length=5)
> x <- 0:8
> as.character(x)
```

```
> x <-c("hello", "world")
> x
[1] "hello" "world"
> y <-vector("character", length=5)
> y
[1] "" "" "" "" ""
> x <-0:8
> x
[1] 0 1 2 3 4 5 6 7 8
> as.character(x)
[1] "0" "1" "2" "3" "4" "5" "6" "7" "8"
> |
```

```
> x <- matrix(1:6, nrow=2, ncol=3)      # column-wise
> dim(x)
> a <- cbind(1:4, 11:14, 1:3) # 세번째 인자 개수 불일치
> b <- rbind(1:4, 11:14)
> a[1,2] <- 100      # 특정값을 변경 가능
> x <- list(1, 3.14, 2+3i, "hello", T)
```

```

>
> x <- matrix(1:6, nrow=2, ncol=3)
> x
      [,1] [,2] [,3]
[1,]     1     3     5
[2,]     2     4     6
> dim(x)
[1] 2 3
>
> a <- cbind(1:4, 11:14, 1:3)
경고메시지(들):
In cbind(1:4, 11:14, 1:3) :
  number of rows of result is not a multiple of vector length (arg 3)
> a
      [,1] [,2] [,3]
[1,]     1    11     1
[2,]     2    12     2
[3,]     3    13     3
[4,]     4    14     1
> b <- rbind(1:4, 11:14)
> b
      [,1] [,2] [,3] [,4]
[1,]     1     2     3     4
[2,]    11    12    13    14
> |

```

Factor, Table, wordcloud

```
> x <- c("apple", "banana", "apple", "apple", "banana")
> y <- factor(x) # 타입변환(symbol), 중복 제거
```

```
> f <- c("apple", "banana")
```

```
> y <- factor(x, levels=f) # levels를 f로 지정 중복 제거
```

```
> words <- table(y) # words <- table(x)
```

```
>
> x <- c("apple", "banana", "apple", "apple", "banana")
> x
[1] "apple" "banana" "apple" "apple" "banana"
> y <- factor(x)
> y
[1] apple banana apple apple banana
Levels: apple banana
> f <- c("apple", "banana")
> f
[1] "apple" "banana"
> y <- factor(x, levels=f)
> y
[1] apple banana apple apple banana
Levels: apple banana
>
> words <- table(y)
> words
y
apple banana
    3      2
> |
```

```
> install.packages("wordcloud") # 패키지 설치 - 1회만!
```

```
> library(wordcloud)
```

```
> pal <- brewer.pal(9, "Set1") # 컬러 인코딩 방식
```

```
> n <- 2 #빈도수 n 이상 디스플레이
```

```
> wordcloud(names(words), freq=words, scale=c(5,1), rot.per=0.25, min.freq=n,
random.order=F, random.color=T, colors=pal)
```

apple
banana

Factor, Table, wordcloud

> # word list와 frequency를 R 외부에서 계산했을 때 처리 방안은?

```
> words <- c("This", "is", "a", "sample", "sentence")
```

```
> w2 <- factor(words)
```

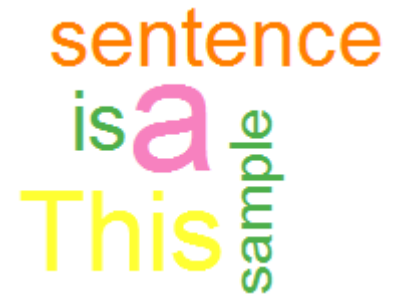
```
> freq <- c(30, 20, 50, 10, 20)
```

```
> length(words)
```

```
> wordcloud(w2, freq, scale=c(5,1), rot.per=0.25, min.freq=n,  
random.order=F, random.color=T, colors=pal)
```

```
> x <- matrix(1:10, nrow=2, ncol=length(words))
```

```
> x <- rbind(w=unclass(words), f=unclass(freq))
```



wordcloud 사용법-1

```
> w1 <- c("apple", "banana", "cherry", "donuts", "endive")
```

소팅 순서 유지

```
> w2 <- factor(w1)
```

```
> f1 <- c("2", "4", "1", "6", "3") # keyword 빈도
```

```
> f2 <- as.integer(f1) # f2 <- c(2, 4, 1, 6, 3)
```

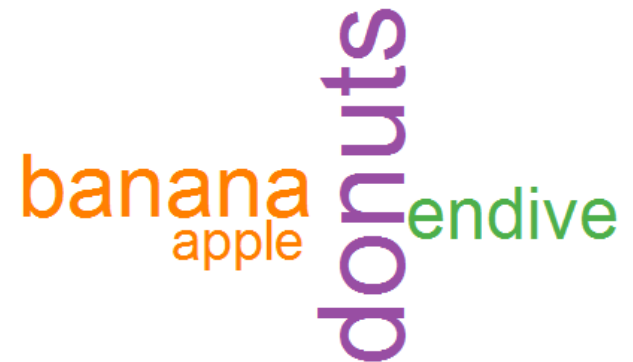
```
> install.packages("wordcloud") # 설치는 1회만...
```

```
> library(wordcloud)
```

```
> pal <- brewer.pal(9, "Set1") # 컬러 인코딩 방식
```

```
> n <- 2 # 빈도수 n 이상 디스플레이
```

```
> wordcloud(w2, f2, scale=c(5,1), rot.per=0.25, min.freq=n,  
random.order=F, random.color=T, colors=pal)
```



wordcloud 사용법-2

```
> word <- read.table("c:/Temp/wordlist.txt")  
> freq <- read.table("c:/Temp/frequency.txt")  
> wordcloud(word, freq, scale=c(5,1), rot.per=0.25,  
min.freq=1, random.order=F, random.color=T, colors=pal)
```

File <wordlist.txt> : *소팅 순서 유지*

C C++ Csharp Java Javascript Objective-C PHP Python Ruby SQL

File <frequency.txt>

95 85 90 100 76 70 68 120 55 50



wordcloud: 텍스트 파일 입력

```
> nouns2 <- read.table("c:/Temp/test.txt")
```

```
# 입력 파일: list of keywords -- nouns2의 클래스는 'data.frame'
```

```
> words <- table(nouns2)      # 빈도 계산 및 테이블 생성
```

```
# words의 클래스는 'table' --- '가나다' 순 소팅 및 빈도 계산
```

```
> library(wordcloud)
```

```
> pal <- brewer.pal(9, "Set1")  # 컬러 인코딩 방식
```

```
> n <- 5                        # 빈도수 n 이상 디스플레이
```

```
> wordcloud(names(words), freq=words, scale=c(5,1), rot.per=0.25,  
  min.freq=n, random.order=F, random.color=T, colors=pal)
```

영어 텍스트 문서 분석 예제

```
> install.packages("tm")      # 설치는 1회만...

> text <- c("bbb", "ccc", "aaa", "bbb", "bbb", "ccc")  # 영어 텍스트 문서
> corpus <- Corpus(DataframeSource(data.frame(text)))
> corpus <- tm_map(corpus, content_transformer(removePunctuation))
> corpus <- tm_map(corpus, content_transformer(tolower))
> corpus <- tm_map(corpus, content_transformer(PlainTextDocument))
> corpus <- tm_map(corpus, content_transformer(function(x) removeWords(x,
stopwords("english")))))

> tdm <- TermDocumentMatrix(corpus)
> m <- as.matrix(tdm)
> v <- sort(rowSums(m), decreasing=TRUE)
> wordcloud(names(v), freq=v, scale=c(5,1), rot.per=0.25, min.freq=n, random.order=F,
  random.color=T, colors=pal)

# 데이터 프레임을 생성하는 방법
> d <- data.frame(word=names(v), freq=v)
> wordcloud(d$word, d$freq, scale=c(5,1), rot.per=0.25, min.freq=n, random.order=F,
  random.color=T, colors=pal)
```

If문, for문

```
txt <- read.csv("test.txt", header=FALSE)
txt$V1 <- as.character(txt$V1)
for (i in 1:nrow(txt)) {
  if (nchar(txt[i,1]) > 0) {
    nouns <- txt[i, 1]
    nouns <- nouns[nchar(nouns) >= 2]
    print nouns
  }
}
```