

Big Data – data frame



About..

컴퓨터소프트웨어공학과
김 원 일



Data Frame – 1

• 또 다른 2차원 배열을 다루는 자료형

- matrix와 달리 다양한 자료형을 포함할 수 있는 matrix
- matrix는 기본적으로 모든 자료형이 동일한 경우에만 사용
- 일반적인 구조체나 클래스의 자료형 선언과 동일
- "변수 <- data.frame(data1, data2, data3 ...) [ENTER]" 형식으로 생성

```
> num <- c( 1, 3, 5, 7 )
> log <- c( T, T, F, T )
> char <- c( "a", "b", "c", "d" )
> df <- data.frame( num, log, char )
> df
```

	num	log	char
1	1	TRUE	a
2	3	TRUE	b
3	5	FALSE	c
4	7	TRUE	d

```
> |
```



Data Frame – 2

- 새로운 데이터 추가

- “변수\$컬럼명 <- c(...) [ENTER]” 형태로 컬럼 추가 가능

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

> df$name <- c( "kim", "park", "lee", "choi" )
> df
  num  log char name
1   1 TRUE   a  kim
2   3 TRUE   b park
3   5 FALSE  c  lee
4   7 TRUE   d choi
> str( df )
'data.frame':   4 obs. of  4 variables:
 $ num : num  1 3 5 7
 $ log : logi TRUE TRUE FALSE TRUE
 $ char: chr  "a" "b" "c" "d"
 $ name: chr  "kim" "park" "lee" "choi"
> |
```



Data Frame – 3

- data frame 간 결합

- rbind(), cbind()를 이용하여 matrix와 같이 결합 가능

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말
> df1 <- data.frame( num, log )
> num1 <- c( 9, 11, 13, 15 )
> log1 <- c( T, F, T, T )
> df2 <- data.frame( num1, log1 )
> df1
```

```
  num  log
1   1 TRUE
2   3 TRUE
3   5 FALSE
4   7 TRUE
```

```
> df2
  num1 log1
1    9 TRUE
2   11 FALSE
3   13 TRUE
4   15 TRUE
> |
```

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말
> df3 <- cbind( df1, df2 )
> df3
  num  log num1 log1
1   1 TRUE    9 TRUE
2   3 TRUE   11 FALSE
3   5 FALSE  13 TRUE
4   7 TRUE   15 TRUE
> |
```



Utilities - 1

- ls()
 - 작업 공간 내에 선언된 모든 변수 목록의 확인
- str(변수 / 데이터)
 - 변수 / 데이터의 종류와 값을 확인할 수 있음
- length(변수 / 데이터)
 - 데이터의 길이를 확인할 수 있음

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지 윈도우즈 도움말

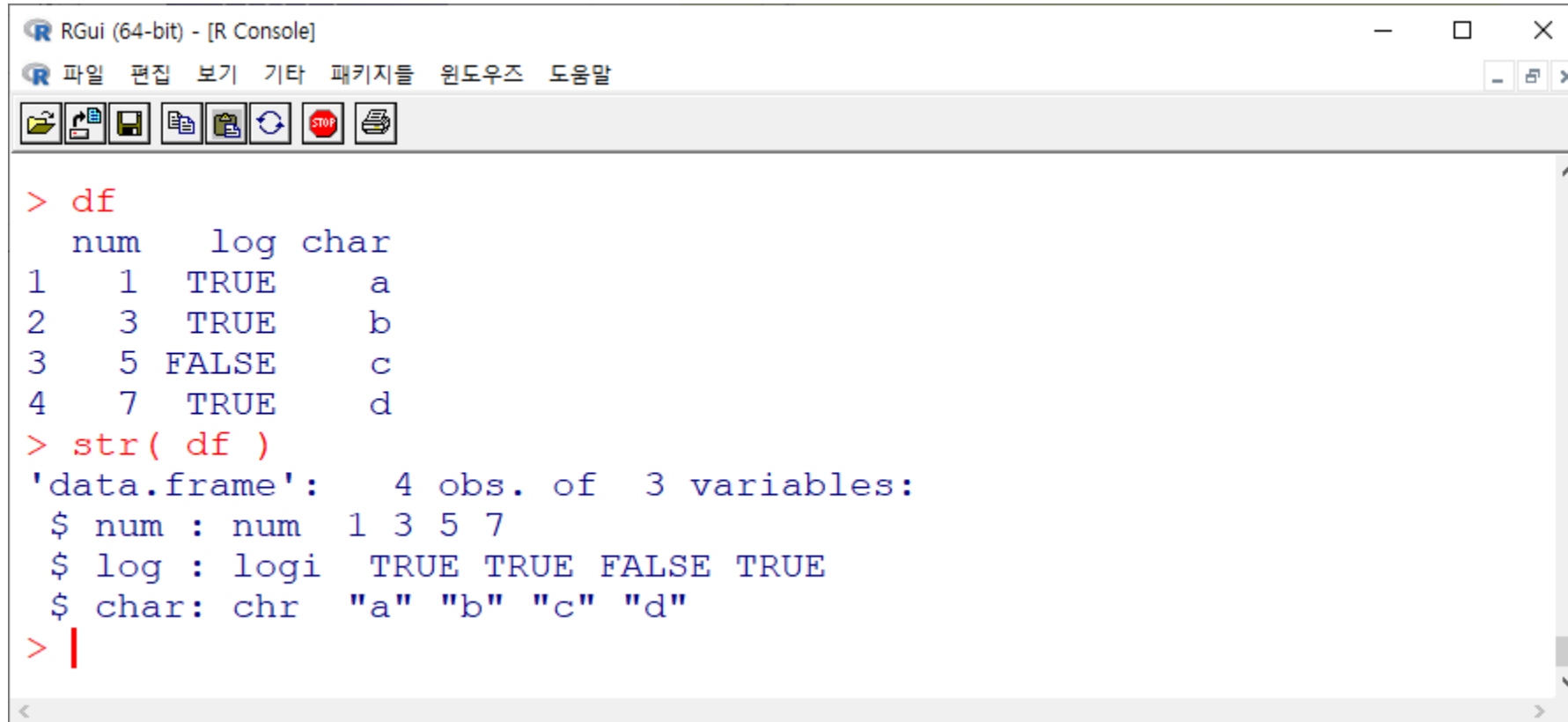
> ls( )
[1] "a"          "b"          "c"
[4] "d"          "mtx"        "userInfo.addr"
[7] "userInfo.name" "userInfo.phone" "vec"

> vec <- c( 1, 2, 3 : 7 )
> mtx <- matrix( vec )
> str( vec )
num [1:7] 1 2 3 4 5 6 7
> str( mtx )
num [1:7, 1] 1 2 3 4 5 6 7
> |
```



Utilities - 2

- **str() 함수로 내부 객체 정보 확인**
 - str(R-object) 형식으로 자료형과 내용을 확인할 수 있음



```
> df
  num  log char
1   1 TRUE   a
2   3 TRUE   b
3   5 FALSE  c
4   7 TRUE   d
> str(df)
'data.frame':   4 obs. of  3 variables:
 $ num : num  1 3 5 7
 $ log : logi TRUE TRUE FALSE TRUE
 $ char: chr  "a" "b" "c" "d"
> |
```



Utilities - 3

- `str()`로 각종 정보 확인
 - 모든 R 변수의 정보를 확인할 수 있음

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말
[Icons: File Explorer, R Script, Save, Print, Run, Stop, Copy]

> str( num )
num [1:4] 1 3 5 7
> str( char )
chr [1:4] "a" "b" "c" "d"
> str( log )
logi [1:4] TRUE TRUE FALSE TRUE
> str( a )
num 2.68e+08
> str( d )
num 69708
> str( userInfo.name )
chr "wikim"
> |
```



- length()로 데이터의 길이 확인
 - 모든 종류의 데이터 길이 확인이 가능

Big Data - sentence



About..

컴퓨터소프트웨어공학과
김 원 일



비교 문장 - 1

- C언어의 if 문장을 그대로 사용 가능

- if ~ else if ~ else ~ 문장 형식 사용 가능

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지를 윈도우즈 도움말
[Icons: File Explorer, R Script, R Package, R Project, R Help, R Stop, R Print]

> if( 777 == a ) print( a )
[1] 777
> if( 777 = a ) print( a )
에러: 예기치 않은 '='입니다 in "if( 777 ="
> if( 777 == a ) print( a )
[1] 777
> if( 333 == a ) print( a ) else print( a + 3 )
[1] 780
> if( 333 == a ) print( a ) else if( a == 777 ) print( a + 3 ) else print( a - 7 )
[1] 780
> if( 333 == a ) print( a ) else if( a != 777 ) print( a + 3 ) else print( a - 7 )
[1] 770
> |
```



비교 문장 - 2

• 조건에 따른 여러 문장 처리

- {, }로 묶고, ";"으로 다수의 문장을 실행할 수 있음
- 여러 문장이 각기 실행되는 형태로, 결과는 한 문장씩 실행되어 출력되는 형태
- (,)는 문장을 묶을 수 없고, 연산자로 사용되어 문법 오류가 발생

```
> if( 777 == a ) { print( a ), print( b ) }
에러: 예기치 않은 ', '입니다 in "if( 777 == a ) { print( a ),"
> if( 777 == a ) { print( a ); print( b ); }
[1] 777
[1] 4096
> if( 777 == a ) ( print( a ); print( b ); )
에러: 예기치 않은 ';'입니다 in "if( 777 == a ) ( print( a );"
> |
```



반복문 - 1

- “for(i in data) { #반복할 문장들 }
– “i”는 data의 값을 순차적으로 갖는 변수로 사용

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지를 윈도우즈 도움말
[Icons]

> for( i in 1 : 9 ) {
+   result = 2 * i
+   print( result )
+ }
[1] 2
[1] 4
[1] 6
[1] 8
[1] 10
[1] 12
[1] 14
[1] 16
[1] 18
> |
```

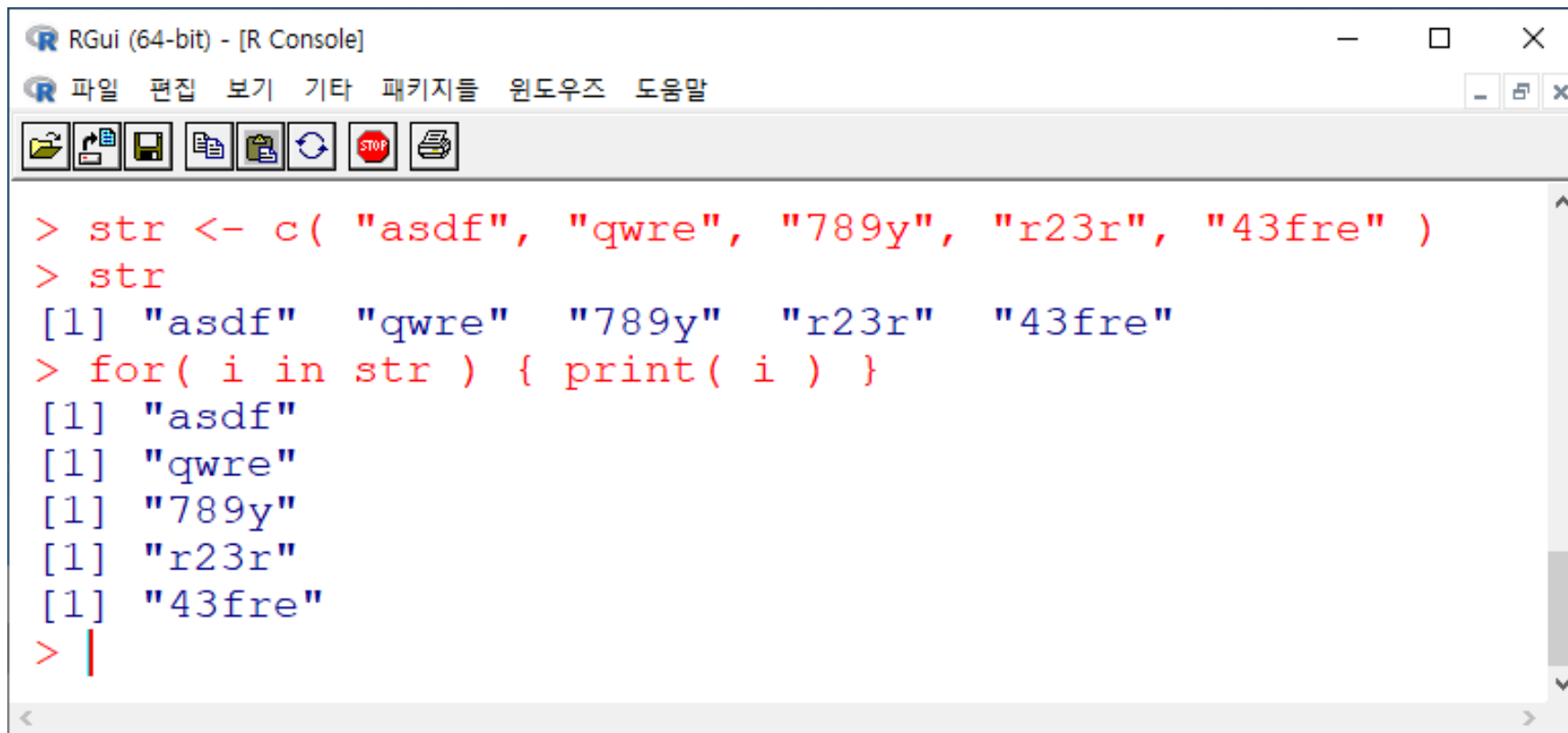
```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지를 윈도우즈 도움말
[Icons]

> for( i in vec ) {
+   result = 2 * i
+   print( result )
+ }
[1] 2
[1] 4
[1] 6
[1] 8
[1] 10
[1] 12
[1] 14
> vec
[1] 1 2 3 4 5 6 7
> |
```



• 문자열 반복문

- for() 문장을 동일하게 사용하여 문자열 순서 접근 가능

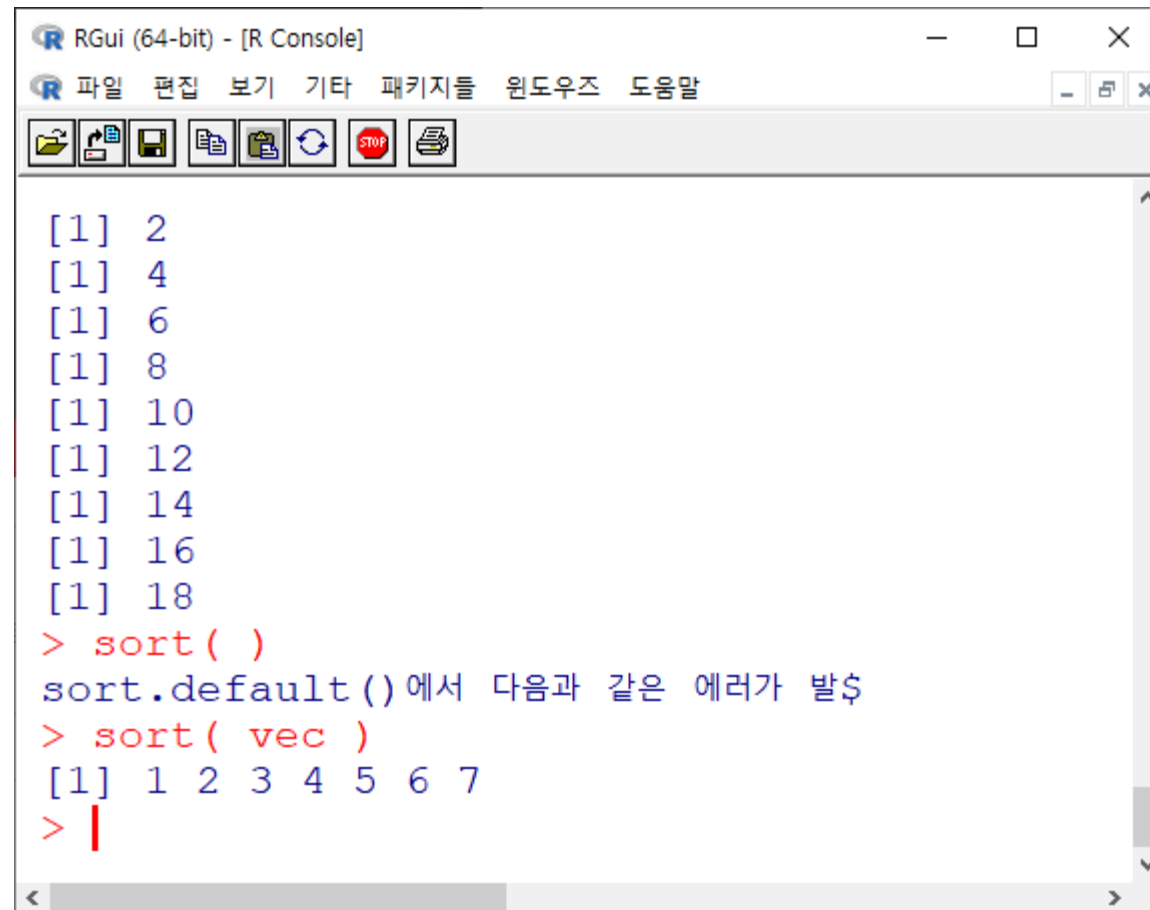


```
> str <- c( "asdf", "qwre", "789y", "r23r", "43fre" )
> str
[1] "asdf"  "qwre"  "789y"  "r23r"  "43fre"
> for( i in str ) { print( i ) }
[1] "asdf"
[1] "qwre"
[1] "789y"
[1] "r23r"
[1] "43fre"
> |
```



• 데이터의 값을 정렬

- "sort(data) [ENTER]"로 값들을 정렬된 상태로 출력 가능



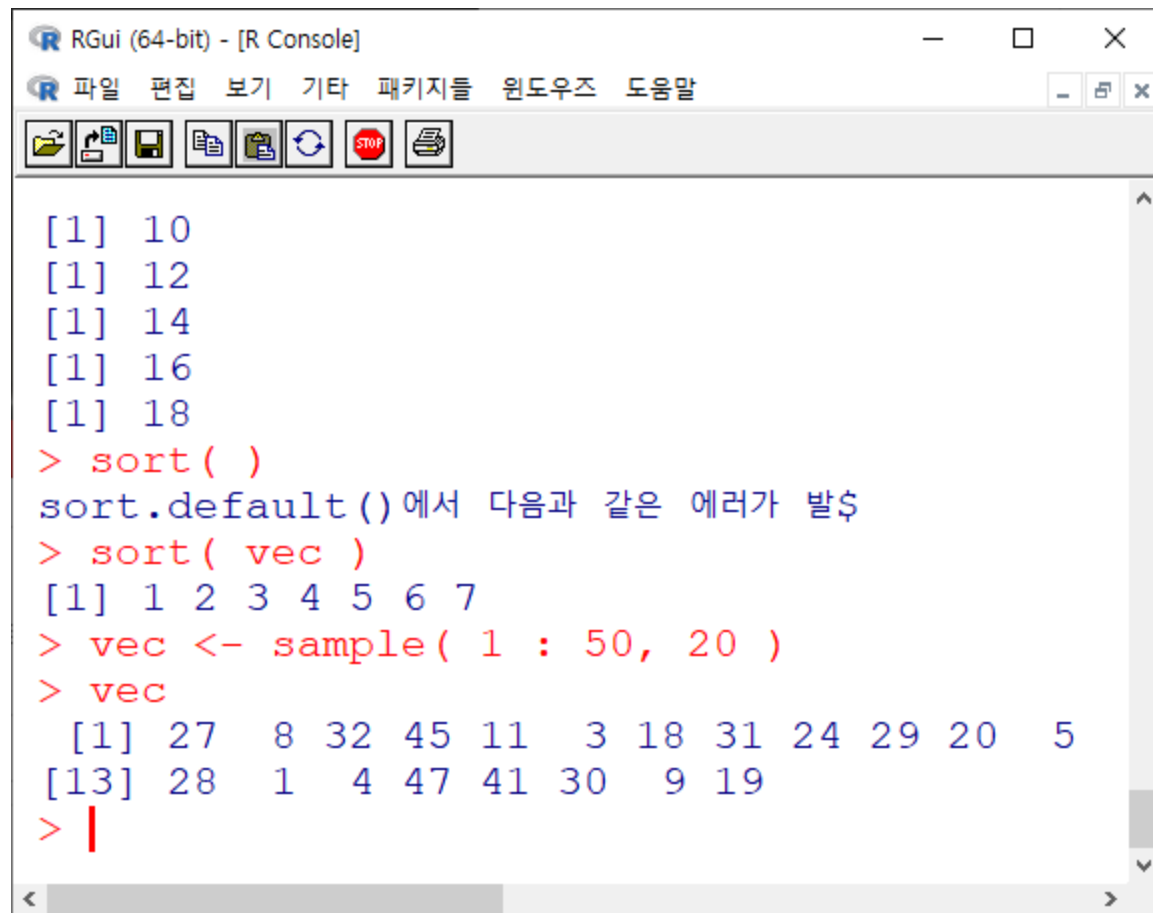
```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지를 윈도우즈 도움말
[1] 2
[1] 4
[1] 6
[1] 8
[1] 10
[1] 12
[1] 14
[1] 16
[1] 18
> sort( )
sort.default()에서 다음과 같은 에러가 발$
> sort( vec )
[1] 1 2 3 4 5 6 7
> |
```



샘플 (랜덤) 데이터 생성

- 샘플 데이터 생성

- "sample(data, count) [ENTER]"로 샘플 데이터 생성



```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말

[1] 10
[1] 12
[1] 14
[1] 16
[1] 18
> sort( )
sort.default()에서 다음과 같은 에러가 발$
> sort( vec )
[1] 1 2 3 4 5 6 7
> vec <- sample( 1 : 50, 20 )
> vec
[1] 27 8 32 45 11 3 18 31 24 29 20 5
[13] 28 1 4 47 41 30 9 19
> |
```



• 샘플 데이터의 값 정렬

- "decreasing = T/F"를 통해 오름차순과 내림차순 설정 가능

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지를 윈도우즈 도움말
sort.default()에서 다음과 같은 예러가 발$
> sort( vec )
[1] 1 2 3 4 5 6 7
> vec <- sample( 1 : 50, 20 )
> vec
[1] 27  8 32 45 11  3 18 31 24 29 20  5
[13] 28  1  4 47 41 30  9 19
> sort( vec )
[1]  1  3  4  5  8  9 11 18 19 20 24 27
[13] 28 29 30 31 32 41 45 47
> sort( vec, decreasing = T )
[1] 47 45 41 32 31 30 29 28 27 24 20 19
[13] 18 11  9  8  5  4  3  1
> |
```




rank(), order()

- rank : 값 크기에 의한 순위 번호
- order : 정렬 순서 번호

```
RGui (64-bit) - [R Console]
파일 편집 보기 기타 패키지들 윈도우즈 도움말
[1] 1 3 4 5 8 9 11 18 19 20 24 27
[13] 28 29 30 31 32 41 45 47
> sort( vec, decreasing = T )
[1] 47 45 41 32 31 30 29 28 27 24 20 19
[13] 18 11 9 8 5 4 3 1
> rand( vec )
rand(vec)에서 다음과 같은 에러가 발생했$
> rank( vec )
[1] 12 5 17 19 7 2 8 16 11 14 10 4
[13] 13 1 3 20 18 15 6 9
> order( vec )
[1] 14 6 15 12 2 19 5 7 20 11 9 1
[13] 13 10 18 8 3 17 4 16
> |
```