6. 문자열, 비정형 데이터 처리

문자열 벡터에서 특정 문자열 패턴 찾기 : grep()

grep(pattern,x)는 문자열 벡터 x에서 특정 부분 문자열 pattern을 찾기 위해 호출한다.

시작 텍스트를 찾기 위해서는 “^”를 사용하며, 끝 텍스트를 찾기 위해서는 “$”를 사용한다.

❑ **사용법 : grep(pattern, x)**

❑ **예제**

```
>Sample <- c("BU1", "BU2", "BM1", "BD1", "BU3")
#시작 텍스트 찾기
>grep(“^BU”, Sample)
[1] 1 2 5
#끝 텍스트 찾기
>grep(“D1$”, Sample)
[1] 4
```

```
> Sample<-c(Sample, "BU")
> Sample
[1] "BU1" "BU2" "BM1" "BD1" "BU3" "BU"
#숫자가 들어가있는 텍스트를 찾으려면 “\d”를 사용한다.
> grep('\\d',Sample)
[1] 1 2 3 4 5
#문자가 들어가있는 텍스트를 찾으려면 “\D”를 사용한다.
> grep('\\D',Sample)
[1] 1 2 3 4 5 6
```

문자열 벡터에서 특정 문자열 패턴 찾기 : grep()와 grepl()의 차이

문자열 벡터에서 특정 문자열 패턴을 찾는 명령은 grep()외에 grepl()도 있다.

차이는 grep는 벡터내 위치를 출력하고, grepl은 논리값(TRUE, FALSE)를 출력한다는 점이다.

❑ **사용법 : grepl(pattern, x)**

❑ **예제**

```
>Sample <- c("BU1", "BU2", "BM1", "BD1", "BU3")
#시작 텍스트 찾기
> grepl("^BU", Sample)
[1]  TRUE  TRUE FALSE FALSE  TRUE
#끝 텍스트 찾기
>grepl("D1$", Sample)
[1] FALSE FALSE FALSE  TRUE FALSE
```

grep와 grepl에서 정규표현식

정규표현식은 와일드 카드의 일종이다. 문자열 관련 다양한 클래스를 축약해 정의한 것이다.

❑ **[au] : a나 u문자 중 하나라도 포함한 문자열을 뜻한다.**

❑ **예제**

```
>grep("[au]",c("Equator","North Pole","South Pole"))
[1] 1 3
```

❑ **마침표(.) : 단일 문자를 나타낸다.**

❑ **예제**

```
>grep("o.e",c("Equator","North Pole","South Pole"))
[1] 2 3
```

→ 여기서는 o이후에 문자 하나가 오고 뒤에 e가 나오는 3개의 문자로 된 문자열을 찾는다.

```
>grep("N..t",c("Equator","North Pole","South Pole"))
[1] 2
>grep(".",c("abc","de","f.g"))
[1] 1 2 3
```

→ 위 명령에서 마침표는 메타문자이므로 3이나오는것이 아니라 1 2 3 이 다 나왔다.

문자열 마침표를 반영하려면 '\'를 사용해 마침표의 메타문자 성격에서 '벗어나야' 한다.

```
>grep("\\.",c("abc","de","f.g"))
[1] 3
```

문자열의 길이를 계산하기 : nchar()

nchar(x)는 문자열 x의 길이를 계산한다.

❑ 사용법 : nchar(x)

❑ 예제

nchar함수를 이용하면 문자열의 길이가 리턴된다.

>nchar("South Pole")

[1] 10

#length함수를 이용하면 문자열 vector의 길이가 리턴된다.

>length(“South Pole”)

[1] 1

>s<-c(“Moe”, “Larry”, “Curly”)

#벡터내 문자열의 길이를 출력하고 싶으면 nchar를 사용한다.

>nchar(s)

[1] 3 5 5

#벡터의 길이를 출력하고 싶으면 length를 사용한다.

>length(s)

[1] 3

여러 문자열을 합치기 : paste()

paste(x1, x2, x3,…)는 문자열 x1, x2, x3 등을 하나의 긴 문자열로 합쳐준다.

❑ 사용법 : paste(x1, x2, x3, sep=“.”)

❑ 예제

```
>paste("North","Pole")
[1] "North Pole"
#단어간 공백대신 다른 문자를 넣고 싶으면 sep옵션을 사용한다.
>paste(“North","Pole",sep=".")
[1] North.Pole
#공백없이 여러 문자열을 합치려면
>paste0(“North”,”Pole”)
[1] "NorthPole“
```

예제에서 선택 인수 sep는
문자열 사이에 기본으로 사용되는 띄어쓰기 대신
다른 문자를 넣고 싶을 때 사용한다.

```
#벡터에 대해서도 적용 가능하다.
> s<-c("Everybody","BoA")
> paste(s,"loves","stats.")
[1] "Everybody loves stats." "BoA loves stats."
> paste0(s,"loves","stats.")
[1] "Everybodylovesstats." "BoAlovesstats."
```

paste나 paste0을 벡터에 적용하면
벡터원소에 paste를 적용한 결과를 리턴한다.

형식에 맞춰 문자열 조합하기 : sprintf()

sprintf(“문자열 %d”, 변수값)은 형식에 맞춰 문자열을 조합해준다.

❑ 사용법 : sprintf(“문자열 %d”, 변수값)

❑ 예제

```
>i<-8
>s<-sprintf("the square of %d is %d,i,i^2)
 [1] "the square of 8 is 64"
```

부분 문자열을 뽑아내기 : substr()

substr(x, start, stop)은 주어진 문자열 x에서 start부터 stop까지의 범위에 위치한 부분 문자열을 출력한다.

❑ 사용법 : substr(“문자열”, 시작, 종료)

❑ 예제

```
>substr("Equator",3,5)
[1] uat
```

특정 문자 기준으로 나누기 : strsplit()

strsplit(x, split)은 x에서 문자열 split을 기준으로 나눠 부분 문자열의 리스트를 만든다.

❑ 사용법 : strsplit(“문자열”, 문자열 나누기 기준)

❑ 예제

```
>strsplit("6-16-2011",split="-")
[[1]]
[1] "6" "16" "2011"
```

특정 패턴이 발생하는 모든 위치 찾기 : gregexpr()

gregexpr(pattern,text)는 text내에서 patten이 발생하는 모든 위치를 찾아준다.

❑ 사용법 : gregexpr(검색대상, “문자열”)

❑ 예제

```
>gregexpr("iss","Mississippi")
[[1]]
[1] 2 5
```

하위 문자열 대체하기 : sub(), gsub()

sub(old, new, string)는 첫번째 하위 문자열만 대체하고, gsub(old, new, string)는 모든 하위 문자열을 대체한다.

❑ 사용법 : sub(old, new, string), gsub(old, new, string)

❑ 예제

```
> s <- "Curly is the smart one. Curly is funny, too."
> sub(pattern="curly", replacement="Moe", x=s)
[1] “Moe is the smart one. Moe is funny, too."
> sub(pattern="curly", replacement="Moe", x=s, ignore.case=TRUE)
[1] "Moe is the smart one. Curly is funny, too."
> gsub(pattern="Curly", replacement="Moe", x=s)
[1] "Moe is the smart one. Moe is funny, too."
```