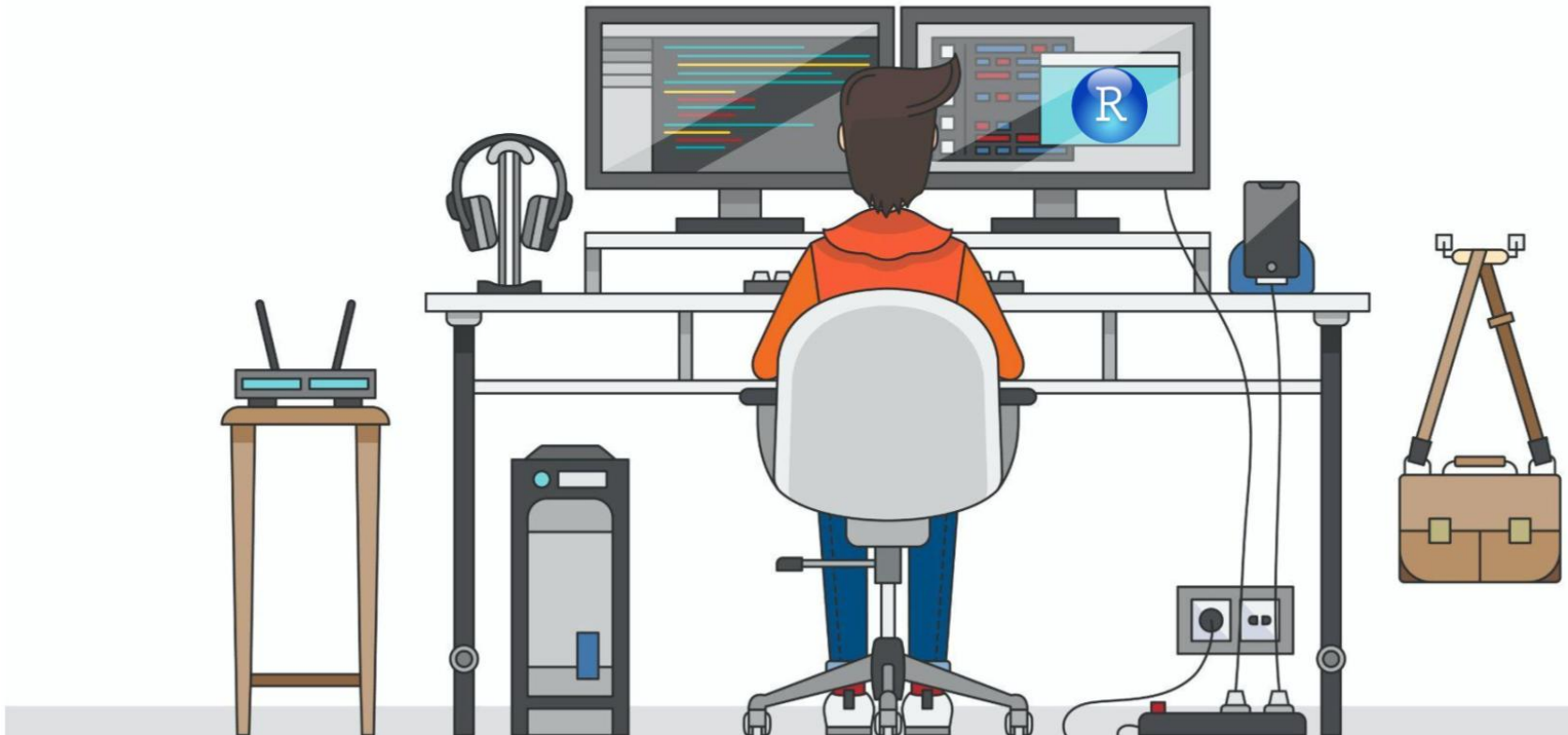
R 프로그래밍 기초다지기

8강 - 문자열 다루기

슬기로운통계생활

Issac Lee



문자열(String) 다루기

문자열의 중요성

문자 데이터의 중요성

- 우리의 일상에서 빼놓을 수 없는 것이 바로 글자!
- 문자를 잘 다루는 능력, 문자를 데이터로 바꾸는 능력은 너무나 중요함.

이번 강의 목표

- 문자를 다루는 R 함수들의 기본적인 사용법을 익히자.
- 문자열 관련 R 패키지들 맛보기

문자열 예제

- `hometown.txt` 파일에는 아래와 같은 백석의 시가 들어 있습니다.

고향(故鄕)_백석

나는 북관(北關)에 혼자 앓아 누워서 어느 아침 의원(醫員)을 뵈이었다.

의원은 여래(如來) 같은 상을 하고 관공(關公)의 수염을 드리워서 먼 옛적 어느 나라 신선 같은데 새끼손톱 길게 돋은 손을 내어 묵묵하니 한참 맥을 짚더니 문득 물어 고향이 어데냐 한다.

평안도 정주라는 곳이라 한즉 그러면 아무개 씨 고향이란다. 그러면 아무개 씨 아느나 한즉 의원은 빙긋이 웃을을 띠고 막역지간이라며 수염을 쓸는다.

나는 아버지로 섬기는 이라 한즉 의원은 또다시 넌지시 웃고 말없이 팔을 잡아 맥을 보는데 손길은 따스하고 부드러워 고향도 아버지도 아버지의 친구도 다 있었다.

문자열 불러오기

- 텍스트 파일의 각 줄이 벡터의 원소가 됨

```
hometown <- readLines("./data/hometown.txt",
                      encoding = "UTF-8")
```

```
head(hometown)
```

```
## [1] "고향(故鄕) 백석"
## [2] ""
## [3] "나는 북관(北關)에 혼자 앓아
## [4] "어느 아침 의원(醫員)을 뵈이
## [5] ""
## [6] "의원은 여래(如來) 같은 상을
```

```
class(hometown)
```

```
## [1] "character"
```

```
length(hometown)
```

```
## [1] 22
```

특정 단어를 검색 `grep()`

특정 단어를 포함한 줄의 위치

- 문법: `grep(패턴, 문자열 벡터)`

```
index <- grep("아버지", hometown)
```

- 18번째, 22번째 줄이 "아버지"를 포함

```
hometown[index]
```

```
## [1] "나는 아버지로 섬기는 이라 한
## [2] "고향도 아버지도 아버지의 친구
```

문자의 길이를 재는 `nchar()`

글자 수를 세어 줌

- 문법: `nchar(문자열)`

```
hometown[1]
```

```
## [1] "고향(故鄕) 백석"
```

```
nchar(hometown[1])
```

```
## [1] 9
```

- 주의점: 공백과 특수 문자도 글자 하나!

여러 개의 문자열을 이어주는 `paste()`

공백의 유무에 따른 함수 선택

```
paste("백석", "고향")
```

```
## [1] "백석 고향"
```

```
paste0("백석", "고향")
```

```
## [1] "백석고향"
```

- 벡터화 코드

```
paste0(1:5, c("st", "nd", "rd", rep("th", 2)))
```

```
## [1] "1st" "2nd" "3rd" "4th" "5th"
```

`paste()` 함수의 주요 옵션

- 문자열 사이를 채워줄 때 `sep`

```
paste("1st", "2nd", "3rd")
```

```
## [1] "1st 2nd 3rd"
```

```
paste("1st", "2nd", "3rd", sep = ", ")
```

```
## [1] "1st, 2nd, 3rd"
```

```
paste(hometown[1], hometown[3])
```

```
## [1] "고향(故鄕) 백석 나는 북관(北關)에 혼자 앓아 누워서"
```

`paste()` 함수의 주요 옵션 2

벡터의 원소들을 문자열로

- 벡터의 각 원소들을 하나로 합쳐서 긴 문자열을 만드는 데에 사용

```
paste(1:5, collapse="")
```

```
## [1] "12345"
```

```
paste(hometown[1:3], collapse=", ")
```

```
## [1] "고향(故鄕) 백석, , 나는 북관(北關)에 혼자 앓아 누워서"
```

문자열의 부분을 가져오는 `substr()`

- 문법: `substr(문자열, 시작점, 끝점)`

```
hometown[1]
```

```
## [1] "고향(故鄕) 백석"
```

```
substr(hometown[1], 3, 6)
```

```
## [1] "(故鄕)"
```

문자열을 나눠주는 `strsplit()`

- 문법: `strsplit(문자열 벡터, 패턴)`

```
hometown[3]
```

```
## [1] "나는 북관(北關)에 혼자 앓아 누워서"
```

```
strsplit(hometown[3], split = " ")
```

```
## [[1]]
## [1] "나는"          "북관(北關)에" "혼자"
## [4] "앓아"          "누워서"
```

- 결과가 리스트로 나옴에 주의

특정 문자 바꿔주기 `gsub()`

- 문법: `gsub(찾을 패턴, 바꿀 내용, 문자열벡터)`

```
hometown[6:7]
```

```
## [1] "의원은 여래(如來) 같은 상을 하고 관공(關公)의 수염을 드리워서"
## [2] "먼 옛적 어느 나라 신선 같은데"
```

```
gsub(" ", "", hometown[6:7])
```

```
## [1] "의원은여래(如來)같은상을하고관공(關公)의수염을드리워서"
## [2] "먼옛적어느나라신선같은데"
```

괄호 안의 문자들은 어떻게 지울까?

Regular expression

- 괄호 안의 한자들을 일괄 삭제 하고 싶다면, 괄호 안 한자들을 모두 가져올 수 없는 노릇

```
hometown[6]
```

```
## [1] "의원은 여래(如來) 같은 상을 하고 관공(關公)의 수염을 드리워서"
```

```
hometown[6] |>
  {\(.) gsub("\\(如來\\)", "", .)}() |>
  {\(.) gsub("\\(關公\\)", "", .)}()
```

```
## [1] "의원은 여래 같은 상을 하고 관공의 수염을 드리워서"
```

정규 표현식

- 복잡한 문자열 패턴들을 일정 규칙을 사용해서 표현

규칙 1. 특수문자들 앞에는 백슬래쉬 2개를 붙여줌

```
gsub("\\.", "", "statistics.playbook")
```

```
## [1] "statisticsplaybook"
```

```
gsub("\\?", "", "statistics?playbook")
```

```
## [1] "statisticsplaybook"
```

정규 표현식

규칙 2. 대문자는 `Not`을 의미

`\\d` - 숫자 (0-9)

`\\w` - 문자

`\\s` - 공백

`\\D` - 숫자가 아닌 것

`\\W` - 문자가 아닌 것

`\\S` - 공백이 아닌 것

```
gsub("\\d", "", "stat.123")
```

```
## [1] "stat."
```

```
gsub("\\D", "", "stat.123")
```

```
## [1] "123"
```

정규 표현식

규칙 3. 점은 줄바꿈을 제외한 모든 문자을 의미

```
random_string <- c("123-123",
                   "123.123",
                   "statistics.playbook",
                   "r-programming")
grep("\\d\\d\\d.\\d\\d\\d", random_string)
```

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

```
grep("\\.", random_string)
```

```
## [1] 2 3
```

정규표현식

규칙 4. 대괄호를 사용하여 `[]` 매칭 조건을 설정할 수 있음.

```
random_string <- c("123-123",
                   "123.123",
                   "123*123",
                   "123!123")
grep("\\d\\d\\d[.*]\\d\\d\\d", random_string)
```

```
## [1] 2 3
```

```
grep("\\d\\d\\d[!-]\\d\\d\\d", random_string)
```

```
## [1] 1 4
```

정규표현식

규칙 4. 매칭 갯수 설정

```
test_string <- c("슬기로운.통계생활", "슬기로운*PlayBOOK")
```

`*` - 0 또는 그 이상

`+` - 1 또는 그 이상

`?` - 0 또는 1

`{3}` - 딱 3개

`{3, 5}` - 3개에서 5개

```
gsub("슬\\w*", "", test_string)
```

```
## [1] ".통계생활" "*PlayBOOK"
```

```
gsub("슬\\w{2}", "", test_string
```

```
## [1] "운.통계생활" "운*PlayBOOK"
```

정규표현식

- `[]` 대괄호 안의 문자 매치
- `[^]` 대괄호 안의 문자 외 매치
- `|` 또는
- `()` 그룹

```
ex_str <- c("Mr. 슬통", "Mr 마통"
            "Ms. Lee", "Mr. R")
m1 <- regexpr("Mr\\.", ex_str)
m2 <- regexpr("Mr\\.?", ex_str)
m3 <- regexpr("M(r|s)\\.?", ex_s
```

```
regmatches(ex_str, m1)
```

```
## [1] "Mr." "Mr."
```

```
regmatches(ex_str, m2)
```

```
## [1] "Mr." "Mr"  "Mr."
```

```
regmatches(ex_str, m3)
```

```
## [1] "Mr." "Mr"  "Ms." "Mr."
```

정규표현식

```
m4 <- regexpr("M(r|s)\\.?\\s[A-Z]\\w*", ex_str)
regmatches(ex_str, m4)
```

```
## [1] "Ms. Lee" "Mr. R"
```

```
m5 <- regexpr("M(r|s)\\.?\\s[가-힣]\\w*", ex_str)
regmatches(ex_str, m5)
```

```
## [1] "Mr. 슬통" "Mr 마통"
```

```
m6 <- regexpr("M(r|s)\\.?\\s[A-Z|가-힣]\\w*", ex_str)
regmatches(ex_str, m6)
```

백석 시 한자 걸러내기

```
hometown[1]
```

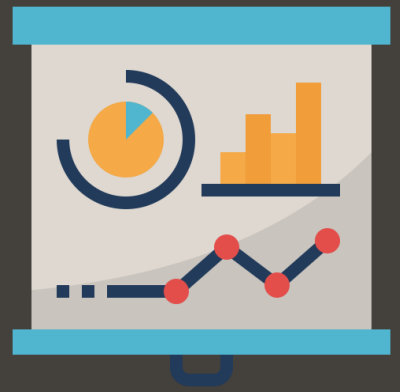
```
## [1] "고향(故鄕) 백석"
```

```
gsub("\\([^가-힣]\\w*\\)", "", hometown[1])
```

```
## [1] "고향 백석"
```

```
total <- paste(hometown, collapse = "\n")
total <- gsub("\\([^가-힣]\\w*\\)", "", total)
file_con <- file("./data/output.txt", encoding="UTF-8")
writeLines(total, file_con)
close(file_con)
```

다음시간



시각화

참고자료 및 사용교재

[1] The art of R programming

- R 공부하시는 분이면 꼭 한번 보셔야 하는 책입니다.
- 위 교재의 한글 번역본 빅데이터 분석 도구 R 프로그래밍도 있습니다. 도서 제목 클릭하셔서 구매하시면 저의 사리사욕을 충당하는데 도움이 됩니다.

[2] Regular Expressions (Regex) Tutorial: How to Match Any Pattern of Text

- Corey Schafer 정규표현식 유튜브 강의 (영어)
- 정규 표현식 내용 기반이 된 영상입니다. 제가 좋아하는 유튜버! :)