R 기초

Package 설치와 활용

Packages

- Package 설치
  - R에서는 핵심프로그램만 설치 후 필요한 것은 package로 추가
  - Package 설치과정 소개
  - Package를 작업 공간에 로드 하는 방법 소개
    - Java와 같은 프로그램을 설치하고 연결이 필요한 경우가 있음
  - Package 업데이트, 제거, 언로드 하는 법 소개

- Excel 파일을 읽어오는 package 소개
  - Package: 'xlsx'와 'readxl'

자바설치되어야함.

Packages

Package 설치

- Package 설치: install.packages("패키지명"), 다운로드 후 설치
- Package를 작업 공간에 로드: library(패키지명), require(패키지명)
  - xlsx는 Java설치 후 path 지정해야 로드
- Package 업데이트, 제거, 언로드
  - 업데이트: update.packages(ask=F)
  - remove.packages(패키지명)
  - detach("package:패키지명",unload=T)

Excel 파일을 읽어오는 package 소개

- xlsx::read.xlsx, readxl::read_excel
- Package에 따라 인수의 형태가 다름

R 기초

문자열 처리

문자열처리 : 문과와구분하지않음(둘다 char)

- Text mining을 위한 작업
- 문자로 이루어진 자료 불러오기 readlines
- 문자열 처리
  - ☑ 문자열 분해(분할) 단어별로 개별로
  - ☑ 문자열 결합
  - ☑ 문자열 대체(변환)
  - ☑ 문자열 선택(참조)
- 문자벡터 처리

실습

문자열처리

- 문자자료 읽기: scan(), readLines()
- 문자열 처리
  - 문자열 분해(분할): strsplit()
  - 문자열 결합: paste()
  - 문자열 대체(변환): sub(), gsub()
    - 소문자로 변환: tolower(), 대문자로 변환: toupper()
  - 문자열 선택(참조): 논리비교, grep()
  - 기타 문자열 함수: nchar(), stringr::str_trim()
- 문자벡터 처리: c(), union(), intersect(), setdiff(), setequal()
  - union() = unique(c())

```
##############
##  R Packages
##############
## * R홈페이지 > cran > mirrors > packages에서 제공
## * 2021/12/29 현재 18653개 (참고: 20/02/20 15332개, 20/10/5 16387개) packages 제공

## Package 설치
## 설치된 패키지 확인
installed.packages()
## * 기본 패키지는 R 폴더 > library
## * 추가 패키지는 C:₩Users₩***₩Documents₩R₩win-library₩4.1

install.packages("xlsx")    # Excel 파일 읽기
install.packages("readxl")  # Excel 파일 읽기

## 설치가 안되는 경우
## * R홈페이지 > cran > mirrors > packages
## * package 선택 > r-release 압축파일 다운로드
## * RStudio: Tools메뉴 > install packages > Package Archive에서 다운로드한 파일선택

## 일부 패키지는 Java, Perl를 활용함 => 관련 컴파일러 필요
## * ex) xlsx
## JRE(JDK) 설치와 Path 지정 후

## 작업공간 상에 불러오기
? read.xlsx
require(xlsx)
? read.xlsx
detach("package:xlsx",unload=T)
? read.xlsx
? xlsx::read.xlsx

if(!require(xlsx))
{
  install.packages("xlsx")
  library(xlsx)
}

## Packages Update
## *  RStudio: Tools메뉴 > Check for Package Updates > 업데이트할 패키지 선택
update.packages(ask=F)

## Package 제거
# remove.packages("xlsx")

##################
## Excel 파일 읽기
```

메모리상에서 지우기

⇒ no document, 메모리 떠있음

↵ T/F 리턴

install됐는지 확인 - 안한경우 설치

한경우 가져오기

이미 설치됐는데 또 설치하면 혼란

ㄴ 안물어보기

→ 설치/제거 : " " (O)

불러오거나 라이브러리 : " " (X)

? read.xlsx

```
##################
Data1 <- read.xlsx("Employee.xlsx", sheetName = "data", encoding="UTF-8")  # sheetName = "data"
str(Data1)  # 문자 => factor. 문자 => 문자
Data2 <- readxl::read_excel("Employee.xlsx",sheet="data")
str(Data2)    # tbl_df, 문자 => 문자
Data2 <- readxl::read_excel("Employee.xlsx",sheet="data",range="B1:I10",col_names=T)
## RStudio GUI: File > import Dataset > From Excel

View(Data2)

## * Excel 파일로 저장
#### - xlsx::write.xlsx(),
#### - install.packages("writexl"); writexl::write_xlsx()
```

저장별도함수

UTF-8(BOM)
: 첫글자가 특수한문자(안보임)
한글이없더라도 encoding을 UTF-8로 해야함

```
##############
##  문자열처리
##############

## * 문자열 자료
Dream <- scan("I have a dream.txt",what="",encoding="UTF-8")
Dreams <- readLines("I have a dream.txt",encoding="UTF-8")  # ₩n이 나올때 까지
Email <- readLines("email.txt")

## 주요 작업
## * 문자열 분해(분할)
## * 문자열 결합
## * 문자열 대체(변환)
#### - 소문자 -> 대문자, 대문자 -> 소문자
## * 문자열 추출

## Dream에서 5글자로 이루어진 단어
#### - nchar: 개별문자열 길이함수
Dream5 <- Dream[nchar(Dream) == 5]
Dream5 <- unique(Dream5)    # 숫자도 가능

## 변환
## * sub: 첫 번째 하위문자 대체
## * gsub: 모든 하위문자 대체
Dream5[34]
test <- sub("'s","",Dream5)
gsub("p","k",Dream5)
gsub(".","",Dream5)

## * Regular Expression Syntax
#### - [:punct:]: !"#$%&'()*+,-.
####                 /:;<=>?@[₩]^_`{|}~
gsub("[[:punct:]]","", Dream5)

#### - 숫자를 *** 표시로 안보이게
Email[1:10]
gsub("₩₩d+","***",Email[1:10])

## 분할(분해)
#### * strsplit: 구분자로 문자열 분할 (리스트로 출력)
# 경로포함된 파일명에서 파일명만 가져오기: lapply, sapply
paths <- c("/home/mike/trials.csv", "/home/mike/data/errors.csv", "/home/mike/corr/reject.doc")
test <- strsplit(paths,"/")
test[[1]]
length(test[[1]])
# filenames <- NULL
# for (i in 1:3)
```

char

한단어씩

한문장씩

사람으로 되어있는 데이터 → scan
말고는 readLines가 더 간편함

UTF-8(BOM)의 특수한 첫글자 포함수의

① 2 5 2 4

(중복포함)

타인어의 Set 과같은 역할 — 중복제거

첫번째 's가 사라짐

모든 p가 k로 바뀜

— 잘못된사용

정규식

*5글자모음에 문장부호가 붙은 단어도 포함됨

모든문장부호(특수문자)를 없앰

암기

String split

test[[1]] ⇒ "" "home" "mike" "trials.csv"

length(test[[1]]) : 벡터 test의 길이 ⇒ 4 (length(test[[1]][4])⇒"trials.csv")

응용: 마지막 요소는 test(length(test[[1]]))

lapply, sapply → 하윤의 빠르고쉬운 형태(유료)

```
#    filenames <- c(filenames,test[[i]][length(test[[i]])])
# filenames

## Email에서 naver.com의 메일리스트
n <- length(Email)
test <- strsplit(Email,"@")
test[[1]][2]

#### - 각 리스트의 두번째 원소: lapply, sapply
emaillist <- NULL
for (i in 1:n)
  if (test[[i]][2]=="naver.com")
    emaillist <- c(emaillist,test[[i]][1])
emaillist[1:10]


#### * substr: 하위문자열 추출함수
test <- "Statistics"
substr(test,1,4)
n <- nchar(Email)
substr(Email,n-8,n)[1:10]
# test <- ifelse(substr(Email,n-8,n)=="naver.com",substr(Email,1,n-10),NA)
# test[!is.na(test)][1:10]

#### * grep(문자열,문자벡터,value=F,invert=F,ignore.case=F, fixed=F)
#### ignore.case = F : 대소문자 구별하지 함
#### fixed = T: 문자열과 정확하게 일치하는 것만 출력
grep("naver.com",Email[1:10])
grep("naver.com",Email[1:10],value=T)
grep("naver.com",Email[1:10],invert=T,value=T)  # not match

## 문자열 연결
## * paste: 문자열 연결함수
#### - 개별문자열에 추가
paste("To",Email[1:10])
paste("To",Email[1:10],sep=":")
paste("To",Email[1:10],sep="")
paste("To",Email[1:10],"AAA")
#### - 하나의 문자열로 연결
Mails <- paste(Email[1:10],collapse=", ")
test <- strsplit(Mails,",")[[1]]
## 공백제거: str_trim(x, side=c("both","left","right"))
gsub(" ","",test[[1]])
# install.packages("stringr")
# library(stringr)
stringr::str_trim(test,side="left")
```

```
## 대문자를 소문자로: tolower()
## 소문자를 대문자로: toupper()
toupper(Dreams[2])
tolower(Dreams[2])

## 문자벡터와 문자벡터 결합:(숫자도 가능)
### 단순히 결합: c()
### 중복인 경우 하나만 출력(수치형가능): union()
### 중복인 것만 출력(수치형가능): intersect()
### x에 y에 없는 것만 출력(수치형가능): setdiff(x,y)
### 순서관계없이 동일여부 확인: setequal(x,y)
X = c("A","B","C")
Y <- c("B","A","D")
c(X,Y)
union(X,Y)
intersect(X,Y)
setdiff(X,Y)
setequal(X,Y)
```

문자비교시 많이 사용

합집합

교집합

차집합 (X-Y)

문자열 결합

→ X,Y 합친후 unique 한것과 같은결과