

R 데이터(텍스트) 마이닝 및 시각화 분석

Seongmin Mun

17th January 2022



아주대학교
AJOU UNIVERSITY

Outline

소개

공지 사항

벡터

행렬

리스트

데이터 프레임

소개

강의자 소개

Seongmin Mun



Skills & Endorsements

Research Background

- NLP
- Machine Learning
- Data analysis
- Data Visualization
- Web Development
- Statistics

Computer Languages

- Java
- Python
- PHP
- JavaScript
- SQL
- R
- HTML/CSS

Statistics Software

- R
- SAS
- SPSS
- MATLAB

<http://seongminmun.com/>

강의 대상

- ▶ R프로그래밍으로 텍스트 마이닝 혹은 데이터 분석을 어떻게 하는지 궁금하신 분들
- ▶ R 프로그래밍에 관심이 있으신 분들

강의 목표

- ▶ 수강생들에게 다소 어렵게 느껴지는 프로그래밍을 차근차근 배움으로써 프로그래밍 활용에 대한 두려움을 낮추고 자심감을 향상시킨다.
- ▶ 텍스트 마이닝의 전반적인 과정 (i.e., 데이터 수집, 자연어 처리, 데이터 분석, 데이터 시각화)을 체계적으로 배움으로써 텍스트 마이닝에 대한 이해를 높이고 프로그래밍을 활용한 데이터 분석 기술을 학습한다.

공지 사항

기본 준비 환경 - R설치



[\[Home\]](#)

Download

CRAN

R Project

About R

Logo

Contributors

What's New?

Reporting Bugs

Conferences

Search

Get Involved: Mailing Lists

Get Involved: Contributing

[Developer Pages](#)

[R Blog](#)

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).

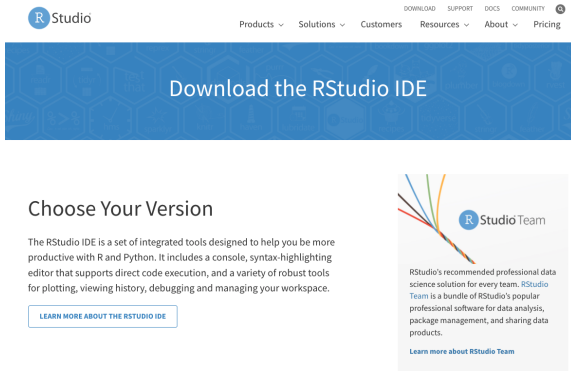
If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **R version 4.1.2 (Bird Hippie)** has been released on 2021-11-01.
- **R version 4.0.5 (Shake and Throw)** was released on 2021-03-31.
- Thanks to the organisers of useR! 2020 for a successful online conference. Recorded tutorials and talks from the conference are available on the [R Consortium YouTube channel](#).
- You can support the R Foundation with a renewable subscription as a [supporting member](#)

<https://www.r-project.org/>

기본 준비 환경 - R Studio



Download the RStudio IDE

Choose Your Version

The RStudio IDE is a set of integrated tools designed to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace.

[LEARN MORE ABOUT THE RSTUDIO IDE](#)

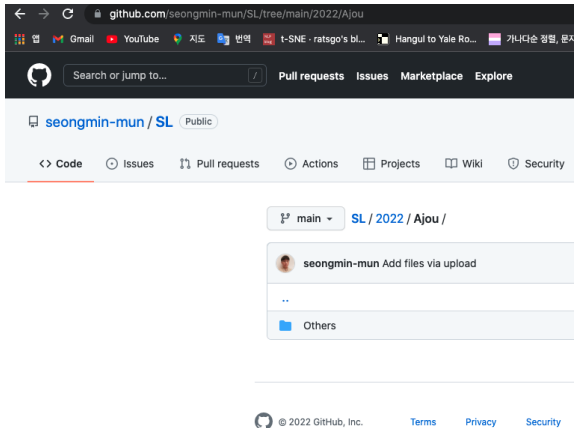
RStudio Team

RStudio's recommended professional data science solution for every team. **RStudio Team** is a bundle of RStudio's popular professional software for data analysis, package management, and sharing data products.

[Learn more about RStudio Team](#)

<https://www.rstudio.com/products/rstudio/download/>

강의 자료 - 강의자 깃헙



<https://github.com/seongmin-mun/SL/tree/main/2022/Ajou>

벡터

숫자 벡터 생성하기

```
> # 1.1 숫자 벡터
> # 정수: 정수로 이루어진 벡터
> a <- c(1, 2, 3, 4, 5)
> a
[1] 1 2 3 4 5
>
> # 실수: 소수점을 포함하고 있는 벡터
> b <- c(1, 2.1, 3, 4, 5.1)
> b
[1] 1.0 2.1 3.0 4.0 5.1
>
> # 벡터의 결합
> c <- c(a,b)
> c
[1] 1.0 2.0 3.0 4.0 5.0 1.0 2.1 3.0 4.0 5.1
```

슬라이싱을 활용한 벡터 생성

```
> # 슬라이싱을 활용한 벡터 생성
> # 정방향
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10
> # 역방향
> 10:1
[1] 10 9 8 7 6 5 4 3 2 1
> # 음수
> -4:8
[1] -4 -3 -2 -1 0 1 2 3 4 5 6 7 8
> # 실수
> 0.7:8
[1] 0.7 1.7 2.7 3.7 4.7 5.7 6.7 7.7
```

생각하기

- ▶ 소수점 단위로 증가하려면 어떻게 해야하는가?

생각하기

- ▶ 반복되는 벡터는 어떻게 생성할까?

기본 연산하기-사칙연산

```
> # 기본 연산하기
> # 사칙연산
> 1 + 1
[1] 2
>
> 2 - 1
[1] 1
>
> 2 * 2
[1] 4
>
> 4 / 2
[1] 2
```

기본 연산하기-몫과 나머지

```
> #몫과 나머지
> #나눗셈 (실수로 표현된다.)
> 10 / 3
[1] 3.333333
> #몫
> 10 %% 3
[1] 3
> #나머지
> 10 %% 3
[1] 1
>
> #거듭제곱
> 2^1
[1] 2
> 2^2
[1] 4
> 2^3
[1] 8
```

문자 벡터 생성하기

```
> #기존의 데이터를 모두 제거
> rm(list=ls())
>
> # 문자 벡터 생성
> alphabets <- c("a","b","c")
> alphabets
[1] "a" "b" "c"
>
```

자료형 변환하기

```
> # 숫자 형을 문자 형으로 변환하기
> numbers <- c(1,2,3)
> numbers
[1] 1 2 3
>
> # 자료형 확인
> mode(numbers)
[1] "numeric"
>
> # 문자 형으로 변환하기
> changeed <- as.character(numbers)
> changeed
[1] "1" "2" "3"
> mode(changeed)
[1] "character"
```

생각하기

- ▶ 문자열을 붙이는건 어떻게 할까?

생각하기

- ▶ 이름이 같고 번호가 붙은 변수 생성

생각하기

- ▶ 문자열을 나누는건 어떻게 할까?

논리 벡터 생성하기

```
> #기존의 데이터를 모두 제거
> rm(list=ls())
>
> # 논리 벡터 생성
> a <- c(TRUE, FALSE, TRUE); a
[1] TRUE FALSE TRUE
>
> a <- c(T, F, T); a
[1] TRUE FALSE TRUE
```


비교 연산으로 논리값 생성하기-1

```
> # 비교 연산으로 논리값 생성하기
>
> # 단일 변수
> a <- 5
> a > 3
[1] TRUE
>
> # 연속 변수
> a <- 1:5
> a > 3
[1] FALSE FALSE FALSE TRUE TRUE
```

비교 연산으로 논리값 생성하기-2

```
> # 이상, 이하
> a <- 1:5
> a >= 3
[1] FALSE FALSE TRUE TRUE TRUE
> a <= 3
[1] TRUE TRUE TRUE FALSE FALSE
>
> # 같다
> a == 3
[1] FALSE FALSE TRUE FALSE FALSE
>
> # 다르다
> a != 3
[1] TRUE TRUE FALSE TRUE TRUE
```

행렬

행렬 생성하기

```
> #기존의 데이터를 모두 제거
> rm(list=ls())
>
> # 2.1 행렬 생성하기
> #행렬 생성하기
> a <- 1:10
> a
[1] 1 2 3 4 5 6 7 8 9 10
> b <- matrix(a, nrow=2, ncol=5)
> b
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
>
> c <- matrix(a, nrow=5)
```

행렬 결합하기-1

```

> # 2.2 행렬 결합하기
> # 행렬 결합하기
> a <- 1:10
> a
[1] 1 2 3 4 5 6 7 8 9 10
> b <- matrix(a,nrow=2,ncol=5)
> b
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
> #열 결합
> c <- cbind(b,c(11,12))
> c
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    3    5    7    9   11
[2,]    2    4    6    8   10   12

```

행렬 결합하기-2

```
> #행 결합하기
> d <- rbind(c,1:6)
> d
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	1	3	5	7	9	11
[2,]	2	4	6	8	10	12
[3,]	1	2	3	4	5	6

```
>
```

행렬 출력하기

```

> #2.3행렬 출력하기
> d[1,2]
[1] 3
>
> d[1,]
[1] 1 3 5 7 9 11
>
> d[,2]
[1] 3 4 2
>
> #1,3열만 추출하기
> d[,c(1,3)]
      [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    1    3

```

행렬 수정하기

```
> #2.4행렬 수정하기
>
> a <- 1:4; a
[1] 1 2 3 4
> b <- matrix(a,nrow=2,ncol=2); b
      [,1] [,2]
[1,]    1    3
[2,]    2    4
>
>
> b[1,1] <- 20220001;b
      [,1] [,2]
[1,] 20220001    3
[2,]         2    4
```


행렬 연산하기

```
> #2.5행렬 연산하기()
> a <- matrix(1:4, c(2,2)); a
      [,1] [,2]
[1,]    1    3
[2,]    2    4
> b <- matrix(10^(1:4), c(2,2)); b
      [,1] [,2]
[1,]   10 1000
[2,]  100 10000
> a * b
      [,1] [,2]
[1,]   10 3000
[2,]  200 40000
> a %% b
      [,1] [,2]
[1,]  310 31000
[2,]  420 42000
```

리스트

리스트 생성 및 호출-1

```
> #기존의 데이터를 모두 제거
> rm(list=ls())
>
> #3.1리스트 생성 및 호출
> a <- list(1:10);a
[[1]]
[1] 1 2 3 4 5 6 7 8 9 10

>
> a[[1]]
[1] 1 2 3 4 5 6 7 8 9 10
```

리스트 생성 및 호출-2

```
> b <- list(id = c(1,2,3,4), names = c("김땡땡", "이땡땡", "최땡땡", "강땡땡")); b
$id
[1] 1 2 3 4

$names
[1] "김땡땡" "이땡땡" "최땡땡" "강땡땡"

>
> b[[1]]
[1] 1 2 3 4
>
> b[[2]]
[1] "김땡땡" "이땡땡" "최땡땡" "강땡땡"
```

리스트 생성 및 호출-3

```
> b[["id"]]
[1] 1 2 3 4
>
> b[["names"]]
[1] "김땡땡" "이땡땡" "최땡땡" "강땡땡"
>
> b$id
[1] 1 2 3 4
>
> b$names
[1] "김땡땡" "이땡땡" "최땡땡" "강땡땡"
>
> b[[1]][2:3]
[1] 2 3
```

리스트 수정 및 결합-1

```
> #3.2리스트 수정 및 결합
> # 수정
> b[[1]] <- c(5:8); b
$id
[1] 5 6 7 8

$names
[1] "김땡땡" "이땡땡" "최땡땡" "강땡땡"

[[3]]
[1] 1 2 3 4
```

리스트 수정 및 결합-2

```
> #결합
> b[[3]] <- 1:4; b
$id
[1] 5 6 7 8

$names
[1] "김땡땡" "이땡땡" "최땡땡" "강땡땡"

[[3]]
[1] 1 2 3 4
```

리스트 연산하기

```
> #3.3리스트 연산하기
> #lapply(): 숫자 벡터로 이루어진 리스트의 각 요소를 연산하는데 사용한다.
> a <- list(1:5,10:15,20:30); a
[[1]]
[1] 1 2 3 4 5

[[2]]
[1] 10 11 12 13 14 15

[[3]]
[1] 20 21 22 23 24 25 26 27 28 29 30

> lapply(a, mean)
[[1]]
[1] 3

[[2]]
[1] 12.5

[[3]]
[1] 25
```


데이터 프레임

데이터프레임 생성하기

```
> #4.1 데이터프레임 생성하기
> ID <- c(1:6); ID
[1] 1 2 3 4 5 6
> Gender <- c("M", "F", "M", "F", "M", "M"); Gender
[1] "M" "F" "M" "F" "M" "M"
> test <- data.frame(id=ID, gender=Gender); test #이름, 데이터
  id gender
1  1      M
2  2      F
3  3      M
4  4      F
5  5      M
6  6      M
```

데이터프레임 값 추가하기

```
> #4.2 데이터프레임 값 추가하기
> eng <- c(90,88,92,86,82,89); eng
[1] 90 88 92 86 82 89
> math <- c(98,89,87,76,85,74); math
[1] 98 89 87 76 85 74
> scores = cbind(eng, math); scores
      eng math
[1,]  90   98
[2,]  88   89
[3,]  92   87
[4,]  86   76
[5,]  82   85
[6,]  89   74
>
> test.score <- data.frame(test, scores); test.score
  id gender eng math
1  1      M  90   98
2  2      F  88   89
3  3      M  92   87
4  4      F  86   76
5  5      M  82   85
6  6      M  89   74
```

데이터프레임 요약하기

```

> #4.3 데이터프레임 요약하기
> # 자료 형 확인
> str(test.score)
'data.frame':  6 obs. of  4 variables:
 $ id      : int  1 2 3 4 5 6
 $ gender  : chr  "M" "F" "M" "F" ...
 $ eng     : num  90 88 92 86 82 89
 $ math    : num  98 89 87 76 85 74
> mode(test.score$id)
[1] "numeric"
> mode(test.score$gender)
[1] "character"
>
> #데이터 요약
> summary(test.score)
      id           gender           eng           math
Min.   :1.00   Length:6   Min.   :82.00   Min.   :74.00
1st Qu.:2.25   Class  :character 1st Qu.:86.50   1st Qu.:78.25
Median :3.50   Mode   :character  Median :88.50   Median :86.00
Mean   :3.50                Mean   :87.83   Mean   :84.83
3rd Qu.:4.75                3rd Qu.:89.75   3rd Qu.:88.50
Max.   :6.00                Max.   :92.00   Max.   :98.00

```