데이터마이닝 입문 · CHAPTER 1

R로 시작하는 데이터마이닝

R 언어의 기본 개념과 문법 — 데이터마이닝을 위한 첫걸음

원저: Luis Torgo, 'Data Mining with R' (2003) 를 바탕으로 재구성한 수업자료

OVERVIEW

학습 목표

1. **데이터마이닝 개념 이해**
   데이터마이닝의 정의와 특징(대용량 데이터, 유용한 지식 발견)을 설명할 수 있다.

2. **R이라는 도구 이해**
   R이 무엇이며 왜 데이터마이닝에 널리 사용되는지 설명할 수 있다.

3. **R 기본 문법 습득**
   벡터·팩터·리스트·데이터프레임 등 핵심 자료구조를 다룰 수 있다.

4. **함수와 제어문 활용**
   간단한 R 함수를 작성하고 if / for 문을 사용할 수 있다.

DATA MINING

데이터마이닝(Data Mining)이란?

데이터로부터 **유용하고, 유효하며, 예상치 못한, 이해 가능한** 지식을 발견하는 것

Useful

유용성 —
실제로 활용 가능한 지식

Valid

유효성 —
새로운 데이터에도 타당함

Unexpected

의외성 —
예상 밖의 통찰 제공

Understandable

이해가능성 —
사람이 해석할 수 있음

통계학·머신러닝·패턴인식과 목표를 공유하지만, 데이터마이닝을 구별짓는 핵심은 다루는 데이터의 '규모(size)'입니다.

왜 데이터마이닝은 어려운가?

- 대용량 데이터: 정보기술의 발전으로 탐색해야 할 데이터의 양이 기하급수적으로 증가
- 계산 효율성: 큰 데이터를 다루기 위한 알고리즘 성능 문제
- 제한된 메모리 자원: 컴퓨터의 메모리 한계를 고려해야 함
- 데이터베이스 연동: 데이터가 파일이 아닌 DB에 저장되는 경우가 많음
- 고차원 데이터 시각화: 변수가 많은 데이터를 어떻게 보여줄 것인가

> *R은 기본적으로 모든 연산을 컴퓨터의 주 메모리(RAM)에서 수행합니다. 하지만 유연한 데이터베이스 인터페이스를 활용하면 큰 데이터도 다룰 수 있습니다.*

ABOUT R

R이란 무엇인가?

- 통계 계산과 그래픽을 위한 언어이자 환경(environment)
- AT&T Bell 연구소에서 개발된 S 언어의 한 갈래(dialect)
- 1996년 Ross Ihaka & Robert Gentleman(오클랜드 대학교)이 개발
- 오픈소스·무료 소프트웨어이며 전 세계 개발자 커뮤니티가 함께 발전시킴
- Windows, macOS, Linux 등 다양한 운영체제와 하드웨어 아키텍처를 지원

> *R의 전신인 S 언어는 1998년 ACM 소프트웨어 상을 수상했습니다. '사람들이 데이터를 분석하고 시각화하고 다루는 방식을 영원히 바꾸어 놓았다'는 평가를 받았습니다.*

ABOUT R

데이터마이닝 도구로서의 R

1

무료 & 오픈소스

비용 부담 없이 전문적인 데이터 분석 환경을 구축할 수 있다.

2

방대한 패키지 생태계

CRAN을 통해 수천 개의 add-on 패키지(통계·머신러닝·시각화 등)를 손쉽게 설치할 수 있다.

3

통계·시각화·DB 연동을 하나로

분석부터 그래픽, 데이터베이스 연결까지 하나의 언어로 처리한다.

4

활발한 커뮤니티

R-help 메일링 리스트, Stack Overflow 등에서 방대한 정보와 도움을 얻을 수 있다.

GETTING STARTED

R 설치와 실행

1

R 설치

CRAN(r-project.org)에서 운영체제에 맞는 배포판을 내려받아 설치

2

실행

R 콘솔을 실행하거나, 오늘날에는 RStudio 같은 통합개발환경(IDE) 사용을 권장

3

명령어 입력

프롬프트 ">" 에서 명령을 입력하고 Enter를 누르면 결과가 출력됨

4

종료

q() 명령으로 종료 — 작업공간(workspace) 저장 여부를 선택할 수 있음

R CONSOLE

```
> x <- 10
> x
[1] 10
> q()
```

R 객체(Object)와 대입연산자

- R에서는 변수·데이터·함수 등 모든 것이 이름을 가진 객체(object)로 저장됨
- 값을 저장할 때는 대입연산자 <- 또는 -> 사용
- ls() : 현재 메모리에 있는 객체 목록 확인
- rm() : 더 이상 필요 없는 객체를 삭제하여 메모리 확보

R CONSOLE

```
> y <- 39
> y
[1] 39
> w <- 5^2
> w
[1] 25
> ls()
[1] "w" "y"
> rm(y)
```

벡터(Vector)

- R에서 가장 기본이 되는 자료구조
- c() 함수로 여러 값을 하나의 벡터로 결합
- mode : 벡터에 담긴 값의 자료형(character/logical/numeric 등)
- length : 벡터에 들어있는 원소의 개수

R CONSOLE

```
> v <- c(4, 7, 23.5, 76.2, 80)
> v
[1] 4.0 7.0 23.5 76.2 80.0
> length(v)
[1] 5
> mode(v)
[1] "numeric"
```

형 변환(Coercion)과 결측값 NA

R CONSOLE

```
> v <- c(4, 7, 23.5, "rrt")
> v
[1] "4"    "7"    "23.5" "rrt"
> u <- c(4, 6, NA, 2)
> u
[1] 4  6 NA 2
```

- 벡터의 모든 원소는 같은 자료형이어야 함
- 자료형이 다르면 R이 자동으로 형을 통일(강제 형변환)함
- NA는 '결측값(missing value)'을 나타내는 특수한 값

벡터 인덱싱 · 원소 추가/삭제

- 대괄호 [ ] 안에 위치(index)를 지정해 원소에 접근·수정
- 존재하지 않는 인덱스에 값을 대입하면 벡터가 자동으로 확장됨(빈 자리는 NA)

R CONSOLE

```
> v <- c(45, 243, 78, 343, 445)
> v[2]
[1] 243
> v[1] <- 100
> v[6] <- 99
> v
[1] 100 243  78 343 445  99
```

벡터화 연산과 재사용 규칙(Recycling Rule)

- 함수나 연산자가 벡터의 모든 원소에 자동으로 적용됨 (반복문 불필요)
- 길이가 다른 두 벡터를 연산하면, 짧은 벡터를 반복(recycle)해 길이를 맞춤

R CONSOLE

```
> v <- c(4, 7, 23.5, 76.2, 80)
> sqrt(v)
[1] 2.00 2.65 4.85 8.73 8.94
> c(1,2,3,4) + c(10,20)
[1] 11 22 13 24
```

팩터(Factor) — 범주형 데이터

- 범주형(명목형) 변수를 표현하기 위한 자료구조
- 가능한 값들을 levels(수준)이라 부름
- 내부적으로는 1, 2, …, k 의 정수로 저장됨

R CONSOLE

```
> g <- c('f','m','m','f','m')
> g <- factor(g)
> g
[1] f m m f m
Levels: f m
```

팩터 활용: 도수분포표

R CONSOLE

```
> table(g)
g
f m
2 3
> table(a, g)
          g
a          f m
  adult    4 2
  juvenile 1 3
```

- table() : 각 범주(level)의 도수를 계산
- 여러 팩터를 함께 넣으면 교차표(cross-tabulation) 생성
- margin.table() : 행/열 주변합, prop.table() : 상대도수

수열(Sequence) 생성

R CONSOLE

```
> 1:10
 [1] 1 2 3 4 5 6 7 8 9 10
> seq(0, 1, 0.25)
[1] 0.00 0.25 0.50 0.75 1.00
> rep(1:3, 2)
[1] 1 2 3 1 2 3
```

- 1:10 → 정수 수열
- seq() → 실수 간격을 갖는 수열
- rep() → 반복되는 수열
- gl() → 팩터(범주형) 수열

인덱싱(Indexing) 기법 총정리

유형	설명	예시
논리 인덱스	조건이 참(TRUE)인 원소만 추출	`x[x > 0]`
정수 인덱스	지정한 위치의 원소를 추출	`x[c(2, 4)]`
음수 인덱스	지정한 위치의 원소를 제외	`x[-1]`
이름 인덱스	names()로 지정한 이름으로 접근	`pH['mud']`
빈 인덱스	모든 원소를 선택	`x[]`

행렬(Matrix)과 배열(Array)

- 행렬 = 2차원으로 배열된 벡터 (matrix() 로 생성)
- 배열 = 2차원 이상으로 확장된 구조 (array() 로 생성)
- m[행, 열] 형태로 인덱싱, 행/열 생략 시 전체 반환
- cbind() / rbind() 로 벡터·행렬을 열/행 방향으로 결합

R CONSOLE

```
> m <- matrix(1:10, 2, 5)
> m
     [,1][,2][,3][,4][,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
> m[2, 3]
[1] 6
> m[1, ]
[1] 1 3 5 7 9
```

리스트(List)

- 서로 다른 타입·길이의 객체를 하나로 묶을 수 있는 순서 있는 집합
- $ 로 이름이 붙은 구성요소(component)에 접근
- [[ ]] 는 값 자체를, [ ] 는 부분 리스트를 반환

R CONSOLE

```
> lst <- list(id=1, name="Kim",
+   marks=c(90,85))
> lst$name
[1] "Kim"
> lst[[3]]
[1] 90 85
```

데이터프레임(Data Frame)

- 행(row) = 관측치(observation), 열(column) = 변수(variable)
- 행렬과 달리 열마다 서로 다른 자료형을 가질 수 있음
- 데이터마이닝에서 가장 자주 사용하는 핵심 자료구조

R CONSOLE

```
> df <- data.frame(
+   site=c('A','B','A'),
+   pH=c(7.4,6.3,8.6))
> df
  site  pH
1    A 7.4
2    B 6.3
3    A 8.6
```

데이터프레임 조건 조회(쿼리)

R CONSOLE

```
> df[df$pH > 7, ]
  site  pH
1    A 7.4
3    A 8.6
> nrow(df)
[1] 3
```

- 조건식을 [행, 열] 자리의 '행' 위치에 넣어 필터링
- nrow() / ncol() 로 행·열 개수 확인
- attach() / detach() 로 열 이름을 직접 사용 가능 (요즘은 with()나 dplyr 계열 함수 사용을 더 권장)

자주 쓰는 내장 함수

분류	함수	설명
기초 통계	`sum, mean, median, sd, var, quantile`	합계·평균·중앙값·표준편차·분산·분위수
정렬·순위	`sort, rank, rev`	정렬, 순위, 순서 뒤집기
반복 적용	`apply, sapply, lapply`	행/열/벡터/리스트에 함수를 일괄 적용
집단 요약	`aggregate, by`	그룹별로 요약 함수를 적용

나만의 함수 만들기

- function(매개변수) { 실행문 } 형태로 함수를 정의하고 이름에 대입
- return() 을 쓰거나, 마지막에 평가된 값이 자동으로 반환됨
- 반복 작업을 자동화하고 코드를 재사용할 수 있음

R CONSOLE

```
> se <- function(x) {
+   v <- var(x)
+   n <- length(x)
+   sqrt(v / n)
+ }
> se(c(45,2,3,5,76,2,4))
[1] 11.10
```

제어문: if 와 for

- if(조건) { … } — 조건이 참일 때만 실행
- for(변수 in 범위) { … } — 정해진 횟수만큼 반복 실행
- cat() 으로 결과를 화면에 출력

R CONSOLE

```
> f <- function(x) {
+   for (i in 1:3) {
+     cat(x,'*',i,'=',x*i,'\n')
+   }
+ }
> f(3)
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
```

세션 관리와 더 배우기

1 **source('code.R')**
텍스트 파일에 저장한 R 코드를 한 번에 실행

2 **save() / load()**
객체를 파일로 저장하고, 다음 세션에서 다시 불러오기

3 **getwd() / setwd()**
현재 작업 디렉터리를 확인·변경

4 **더 배우기**
R 배포판에 포함된 'An Introduction to R' 매뉴얼, CRAN 문서 참고

SUMMARY

Chapter 1 정리

- 데이터마이닝 = 대용량 데이터에서 유용하고 이해 가능한 지식을 발견하는 과정
- R = 무료·오픈소스 통계 계산 및 그래픽 언어이자 환경
- 벡터·팩터·리스트·데이터프레임은 R의 핵심 자료구조
- 함수 작성과 if / for 제어문으로 반복 작업을 자동화

다음 시간

Chapter 2 – 조류 대발생(Algae Blooms) 예측: 실제 데이터로 배우는 첫 데이터마이닝 사례연구

PRACTICE

직접 해보기

1

벡터와 결측값

x <- c(3, 7, NA, 12, 5) 벡터를 만들고, 결측값을 제외한 평균을 구해보세요. (힌트: mean(x, na.rm=TRUE))

2

팩터와 도수분포표

학생 5명의 성별 벡터를 만들어 factor()로 변환하고, table()로 도수를 확인해보세요.

3

데이터프레임 필터링

이름과 점수로 이루어진 데이터프레임을 만들고, 점수가 80점을 넘는 학생만 조회해보세요.