1. c() 함수를 이용한 자료 입력과 출력

> x<-c(1,2,3) // x=c(1,2,3)도 이와 결과값이 동일

> x

[1] 1 2 3 // 출력 중 [1]은 자료가 벡터 형태임을 의미함

> x<-y<-c(1,2,3) // x=y=c(1,2,3)

> x

[1] 1 2 3

> y

[1] 1 2 3

> x=c("서울","수원","대전") // 텍스트를 행렬값에 넣을땐 “”필수

> x

[1] "서울" "수원" "대전”

2. 특정한 위치에 있는 자료값 출력

> x=c(2,5,4,7,8,1) // 2, 3, 4, 6번의 예시는 전부 왼쪽의 x를 기준으로 함,

아래 명령어는 출력될 때만 유효 ↴

Ex) x[-3] 명령어가 출력됐다고 해서 x에서 3열의 원소가 삭제된 것은 아님

> x[3] // x의 3열의 원소

[1] 4

> x[1:4] // x의 1부터 4열까지의 원소

[1] 2 5 4 7

> x[-3] // x에서 3열을 제외한 나머지 원소

[1] 2 5 7 8 1

> x[c(2,5)] // x의 2, 5열의 원소

[1] 5 8

> x[c(-2,-5)] // x에서 2, 5열의 원소를 제외한 나머지 원소

[1] 2 4 7 1

3. 저장된 자료값의 일치와 위치를 알아봄 // x=c(2,5,4,7,8,1)

> x==4 // x의 값이 4와 같은가?

[1] FALSE FALSE TRUE FALSE FALSE FALSE

> x!=4 // x의 값이 4와 다른가?

[1] TRUE TRUE FALSE TRUE TRUE TRUE

> which(x==4) // x의 값이 4가 되는 곳은 어디인가?

[1] 3

> which(x==10) // 10열은 존재하지 않음

integer(0)

> which.min(x) // x의 최소값의 위치는 어디인가?

[1] 6

> which.max(x) // x의 최대값의 위치는 어디인가?

[1] 5

4. 기타 // x=c(2,5,4,7,8,1)

> max(x) // x의 최대값은 무엇인가?

[1] 8

> length(x) // x의 열의 개수는 몇인가?

[1] 6

> y=1:15 // y에 1~15까지로 구성된 정수 행벡터를 대입함

> y

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

> z=seq(1,20,2) // z에 1~20까지로 구성된 정수 헹벡터에서 2의 배수를 제외한 값을 대입함

> z

[1] 1 3 5 7 9 11 13 15 17 19

5. 참고

NA (not available) : 결측값을 나타내는 길이 1인 논리상수

Inf (infinite) : 무한대 // Ex) > 15/0

[1] Inf

NaN (not a number) : 부정형을 나타내는 길이 1인 수치상수 // Ex) > 15/0-27/0

ㅈ[1] NaN

6. 기타 // x=c(2,5,4,7,8,1)

> x1=c(x,10,24,21) // x에 10, 24, 21을 더해서 x1에 저장

> x1

[1] 2 5 4 7 8 1 10 24 21

> x2=x1[c(-1,-4)] // x1에서 1, 4열을 제외하고 x2에 저장

> x2

[1] 5 4 8 1 10 24 21

> x2[10:12]=c(31,21,40) // x2의 10~12열의 원소를 31, 21, 40으로 변경 (8, 9열의 원소가 존재하지 않으므로 NA상태가 됨

> x2

[1] 5 4 8 1 10 24 21 NA NA 31 21 40

> x3=x2[!is.na(x2)] // x2의 결측치 (NA)를 제거함 Ex) x3=x2[!is.na(x2)]

> x3

[1] 5 4 8 1 10 24 21 31 21 40

> x4=x3[x3>20] // x3의 원소중에 20을 초과하는 것만 x4에 저장

> x4

[1] 24 21 31 21 40

7. R편집기

파일 → 새 스크립트 → R편집기 // R편집기에서 저장 시 확장자가 R로 저장됨

R편집기 → 명령어 입력 → 줄 또는 선택영역 실행 → 선택한 명령어가 Console에서 실행됨

Console클릭 → 파일 → 파일에 저장 // Console에서 저장 시 확장자가 txt로 저장됨

Boxplot 명령어 실행 → 우클릭 또는 파일 → 원하는 형식으로 저장

Ex 1) x[-3]의 결과를 새로운 변수 xx에 저장하라.

> xx=x[-3]

> xx

[1] 2 5 7 8 1

Ex 2) x의 첫 번째 값과 네 번째 값을 없애라.

> x[c(-1-4)]

[1] 2 5 4 7 1

Ex 3) 자료값 6,2,7,4,1,3을 변수 y에 입력하라.

> y=c(6,2,7,4,1,3)

> y

[1] 6 2 7 4 1 3

Ex 4) 변수 y의 자료 중 최소값이 있는 위치를 구하라.

> which.min(y)

[1] 5

Ex 5) 변수 y의 자료 최소값을 y1에 저장하라

> y1=min(y)

> y1

[1] 1

Ex 6) 변수 y의 자료 중 5보다 작거나 같은 값만 선택해서 변수 y5d에 저장 하라.

> y5d=y[y<=5]

> y5d

[1] 2 4 1 3

6. 데이터 프레임

- 자료를 Spreadsheet와 유사한 표의 형식으로 표현함

- 열(column)들은 여러 변수들, 행(row)은 같은 사람이나 사물에 대한 다른 측정값 또는 변수 값을 표현함

- R 내부에서 만들기도 하고, 다른 프로그램에서 만든 자료를 불러오기도 함

- 하나의 list로서 원소(변수)들은 동일한 길이의 열벡터들이 됨

7. data.frame 함수

> 학번=c(1601,1602,1603,1604,1605,1606,1607) // 수치형 벡터

> 성별=c("m","f","m","m","f","m","f") // 문자형 벡터

> 점수=c(70,80,NA,89,75,50,55)

> exam=data.frame(학번,성별,점수)

> exam1=data.frame(code=학번,sex=성별,score=점수)

|  |  |
| --- | --- |
| > exam  학번 성별 점수  1 1601 m 70  2 1602 f 80  3 1603 m NA  4 1604 m 89  5 1605 f 75  6 1606 m 50  7 1607 f 55 | > exam1  code sex score  1 1601 m 70  2 1602 f 80  3 1603 m NA  4 1604 m 89  5 1605 f 75  6 1606 m 50  7 1607 f 55 |

|  |  |
| --- | --- |
| > head(exam1) // 앞 6행까지만 출력  code sex score  1 1601 m 70  2 1602 f 80  3 1603 m NA  4 1604 m 89  5 1605 f 75  6 1606 m 50 | > tail(exam1) // 뒤 6행까지만 출력  code sex score  2 1602 f 80  3 1603 m NA  4 1604 m 89  5 1605 f 75  6 1606 m 50  7 1607 f 55 |

8. data.frame안에 있는 자료접근

> exam1$code // code열의 요소들을 나타냄

[1] 1601 1602 1603 1604 1605 1606 1607

> exam1$sex

[1] "m" "f" "m" "m" "f" "m" "f"

> mean(exam1$score) // NA가 있어서 평균이 구해지지않음

[1] NA

> mean(exam1$score, na.rm=T) // mean: 평균을 구함, na.rm=T: NA를 삭제함

[1] 69.83333

> exam1[ ,2] // 앞은 행 뒤는 열 Ex) 공백, 2는 모든행에 대한 2열을 나타냄

[1] "m" "f" "m" "m" "f" "m" "f"

> exam1[5,2]

[1] "f"

> exam1[2:4,1:2] // 2~4행, 1~2열

code sex

2 1602 f

3 1603 m

4 1604 m

> exam2=exam1[-3, ] // exam1에서 2행을 제거 후 exam2에 저장함

> exam2

code sex score

1 1601 m 70

2 1602 f 80

4 1604 m 89

5 1605 f 75

6 1606 m 50

7 1607 f 55

Ex 7) exam1의 세 번째, 네 번째 행을 삭제하라

> exam2=exam1[c(-3,-4), ]

> exam2

code sex score

1 1601 m 70

2 1602 f 80

5 1605 f 75

6 1606 m 50

7 1607 f 55

Ex 8) exam1의 두 번째 열을 삭제하라

> exam2=exam1[ ,-2]

> exam2

code score

1 1601 70

2 1602 80

3 1603 NA

4 1604 89

5 1605 75

6 1606 50

7 1607 55

Ex 9) exam1의 세 번째 행과, 두 번째 열을 삭제하여 새로운 데이터 프레임 hw에 저장하라 / Ex 10) hw 내용을 출력하라

> hw=exam1[-3,-2]

> hw

code score

1 1601 70

2 1602 80

4 1604 89

5 1605 75

6 1606 50

7 1607 55

9. 데이터 편집기

> test=edit(data.frame()) // var1, var2…를 클릭후 자신이 원하는 변수명으로 편집

|  |  |  |
| --- | --- | --- |
| Code | Score | 성별 |
| 12 | 90 | 여 |
| 34 | 90 | 남 |

편집후 나가고

> test

Code score 성별

1 12 90 여

2 34 90 남

> newtest=edit(test)

|  |  |  |
| --- | --- | --- |
| Code | Score | 성별 |
| 12 | 90 | 여 |
| 34 | 90 | 남 |
| 1 | 60 | 여 |

> newtest

code score 성별

1 12 90 여

2 34 90 남

3 1 60 여

|  |  |  |
| --- | --- | --- |
| Code | Score | 성별 |
| 12 | 90 | 여 |
| 34 | 90 | 남 |
|  |  |  |

code score 성별

1 12 90 여

1. 34 90 남
2. NA NA <NA>=

> newtest[-3, ]

code score 성별

1 12 90 여

2 34 90 남

10. 참고

- 데이터 편집기는 작은 규모의 자료 입력시 사용하지만 대량의 자료를 입력하기에는 불편하다

- 우리는 엑셀에서 자료를 입력한 후에 R로 불러 사용할 것이다

Ex) > read.csv(“test.csv”)

11. 엑셀 자료 파일을 R로 불러오는법

- 엑셀 자료 파일 → 다른 이름으로 저장 → 확장자 csv로 (utf-8 X) → R Console에서 작업 디렉토리 변경 (csv파일을 저장한 경로로) → > 하고싶은 함수명=read.csv("csv파일의 이름.csv") → 하고싶은 함수명

12. scan 함수, sort 함수, rank 함수

|  |  |
| --- | --- |
| > x=scan()  1: 1 2 3 4 5 6  7:  Read 6 items  > x  [1] 1 2 3 4 5 6 | > z=scan(what="character")  1: 서울 대전 청주  4:  Read 3 items  > z  [1] "서울" "대전" "청주” |

> x=c(15,7,9,23,8)

> sort(x)

[1] 7 8 9 15 23 // 단순한 크기 순 정렬

> rank(x) // 크기가 큰 것의 위치 순 정렬

[1] 4 1 3 5 2

13. order 함수

> x=c(6,3,4,2,5,3) // 올림차순으로 최소값부터 최대값이 있는 위치를 출력함

> order(x)

[1] 4 2 6 3 5 1

> x=c(6,3,4,2,5,3) // 내림차순으로 최대값부터 최소값이 있는 위치를 출력함

> order(x, decreasing=T)

[1] 1 5 3 2 6 4

> x=c(5,1,7,8,2,5)

> y=c(21,12,34,5,7,10)

> order(x)

[1] 2 5 1 6 3 4

> order(y)

[1] 4 5 6 2 1 3

> y[order(x)] // order(x)의 기준으로 y를 정렬함

[1] 12 7 21 10 34 5

14. 데이터 프레임 형성 (아래의 데이터 프레임으로 Ex를 수행)

|  |
| --- |
| > x=c(2,4,1,5,6,7,10,3,16,17)  > y=c(13,2,21,4,6,13,6,31,6,7)  > group=c(2,1,2,1,1,1,2,1,2,2)  > score=c("h","h","l","l","h","l","l","l","h","h")  > ex=data.frame(x,y,group,score)  > ex  x y group score  1 2 13 2 h  2 4 2 1 h  3 1 21 2 l  4 5 4 1 l  5 6 6 1 h  6 7 13 1 l  7 10 6 2 l  8 3 31 1 l  9 16 6 2 h  10 17 7 2 h |

|  |  |
| --- | --- |
| 정렬 올림차순 | 정렬 내림차순 |
| > exi=ex[order(ex$x), ]  > exi  x y group score  3 1 21 2 l  1 2 13 2 h  8 3 31 1 l  2 4 2 1 h  4 5 4 1 l  5 6 6 1 h  6 7 13 1 l  7 10 6 2 l  9 16 6 2 h  10 17 7 2 h | > exd=ex[order(ex$x,decreasing = T), ]  > exd  x y group score  10 17 7 2 h  9 16 6 2 h  7 10 6 2 l  6 7 13 1 l  5 6 6 1 h  4 5 4 1 l  2 4 2 1 h  8 3 31 1 l  1 2 13 2 h  3 1 21 2 l |

Ex 11) 데이터 프레임 ex의 변수 y에 대해 내림차순으로 정렬하여 newex에 저장하라

> newex=ex[order(ex$y,decreasing = T), ]

> newex

x y group score

8 3 31 1 l

3 1 21 2 l

1 2 13 2 h

6 7 13 1 l

10 17 7 2 h

5 6 6 1 h

7 10 6 2 l

9 16 6 2 h

4 5 4 1 l

2 4 2 1 h