

변숫값 코딩변경(recoding)

📖 개요

📖 Recoding 방법 1

📖 Recoding 방법 2

📖 계산 보기

Loren ipsum dolor sit amet, ius an molestie facilisi erroribus, mutat nalerum delectus ei vis. Has ornatus conclusionemque id, an videri molestatis sit. In etqui praesent sit. An vel agan porro comprehensan, ad ludus constituto nea, et ius utroque scaevola assumaverit.

Via cu nodus nulla feugait, oratio facilisi in usu, eilit vitae sea te. Ea fabulas accusamus dissentias sea, facete tacinales definitiones et per. Nihil dicant mediocrem pro eu, no mei nostro sensibus platonem. Qui id sunno perpetus neglegantur. Vel ipsum novum copiosae ut. Quo et liber detracto probatus. Nam augue scribentur an. Sea oporteat percipitur incidereat ab. Qui viris nemore an.

개요



코딩변경(recoding)의 정의

 원자료의 값을 다른 값으로 변경하는 것

사례 : 체질량지수(Body Mass Index)

 사람의 키와 체중에 의해 키에 따른 적절한 체중을 제시하는 방법

$$BMI = \frac{\text{몸무게(kg)}}{\text{키(m)}^2}$$

* 몸무게는 kg, 키는 m를 단위로 사용






사례 : 체질량지수(Body Mass Index)

대한비만학회의 체질량지수에 따른 기준

BMI	18.5 이하	18.5~23	23~25	25~30	30+
판정	저체중	정상	과체중	초기비만	비만

코딩변경의 필요성

-  만일 BMI를 기준으로 각 그룹에 속한 자료의 수를 빈도표 등으로 얻으려면 BMI의 원자료 값이 아닌 위 기준으로 코딩을 변경한 자료로 빈도표를 얻어야 함





사용할 자료

```
> BMI <- read.table(url("http://jupiter.hallym.ac.kr/ftpdata/data/bmi.txt"),  
  col.names=c("height", "weight", "year", "religion", "gender", "marriage"))
```

+ 자료출처


- url 함수를 사용하여 인터넷 사이트에서 바로 읽어 올 수 있으며, 다음의 인터넷주소를 브라우저 창에 붙여 넣으면 브라우저에서 볼 수 있음(<http://jupiter.hallym.ac.kr/ftpdata/data/bmi.txt>)

+ 내용 : 2000년, 177명에 대한 조사 결과

- 키, 몸무게, 출생년도
- 종교(Bu=불교, C1=개신교, C2=가톨릭, No=없음)
- 성별(F=여자, M=남자)
- 결혼여부(N=미혼, Y=기혼)



BMI를 기준으로 빈도표 계산

-  자료에는 나이와 체질량지수가 계산되어 있지 않으므로 이를 계산하여 데이터 프레임 BMI에 추가함

```
> BMI$age <- 2000 - BMI$year # 나이 계산(2000년 당시)
> BMI$bmi <- BMI$weight/(BMI$height/100)^2 # 체질량지수(BMI) 계산
> head(BMI)
```

	height	weight	year	religion	gender	marri	age	bmi
1	167	68	1974	No	F	N	26	24.38237
2	162	49	1974	C2	F	N	26	18.67093
3	158	50	1978	C2	F	N	22	20.02884
4	165	56	1977	No	F	N	23	20.56933
5	160	52	1959	No	F	N	41	20.31250
6	162	52	1972	No	F	Y	28	19.81405





Recoding 방법 1



if 문 사용



조건에 따라 새로운 값을 설정하는 원시적인 방법

```
for (i in 1:dim(BMI)[1]) {  
  if (BMI$bmi[i] < 18.5) BMI$cbmi[i] <- 1  
  if (BMI$bmi[i] >= 18.5 && BMI$bmi[i] < 23) BMI$cbmi[i] <- 2  
  if (BMI$bmi[i] >= 23 && BMI$bmi[i] < 25) BMI$cbmi[i] <- 3  
  if (BMI$bmi[i] >= 25 && BMI$bmi[i] < 30) BMI$cbmi[i] <- 4  
  if (BMI$bmi[i] > 30) BMI$cbmi[i] <- 5  
}  
> table(BMI$cbmi)  
1  2  3  4  
32 123 19 3
```

for 반복문을 사용한 반복은 R-언어에서는 가능하면 피해야 할 방법 중의 하나이며
for 문을 사용하지 않을 수 있음



Recoding 방법 2



if 문 사용



for 문을 사용하지 않은 경우

```
BMI$cbmi2 <- BMI$bmi  
BMI$cbmi2[BMI$bmi < 18.5] <- 1  
BMI$cbmi2[BMI$bmi >= 18.5 & BMI$bmi < 23] <- 2  
BMI$cbmi2[BMI$bmi >= 23 & BMI$bmi < 25] <- 3  
BMI$cbmi2[BMI$bmi >= 25 & BMI$bmi < 30] <- 4  
BMI$cbmi2[BMI$bmi >= 30] <- 5
```

```
> table(BMI$cbmi2)
```

```
1  2  3  4  
32 123 19  3
```

* 내부적으로 if 문이 보이지는 않지만 실질적으로 if 문이 사용된 것임

- 이 경우에는 두 조건 사이의 '그리고(AND)'에 해당하는 것으로 & 하나만 사용하여 모든 비교를 해야 함
- &&로 두 개를 사용한 경우 한 번의 비교에서 F가 나오는 순간 논리값이 FALSE 한 개가 됨

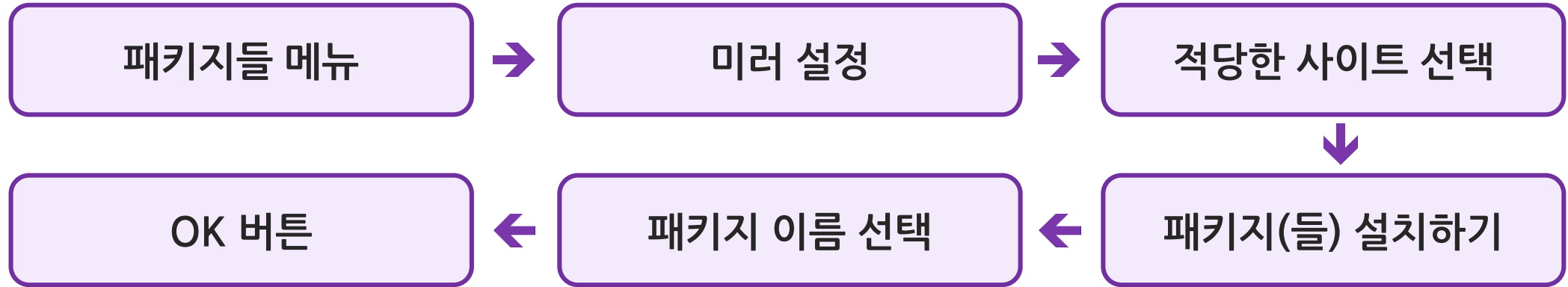


Recoding 방법 3



car 패키지의 recode 함수로 코딩변경

+ car 패키지를 사용하면 recode 함수가 제공되며 이 함수를 사용하면 간단하게 코딩변경 가능



➤ 설치된 패키지를 사용하려면 library 함수로 해당 패키지를 불러옴





Recoding 방법 3



car 패키지의 recode 함수로 코딩변경

+ 사용 함수

```
recode(x, "범위1=값1; 범위2=값2; 범위3=값3; ...")
```

+ 매개변수

- x : 코딩변경할 자료가 포함된 벡터
- " " : 큰 따옴표 안에 코딩변경 규칙을 설정
 - 각각의 코딩변경 규칙은 세미콜론(;)으로 구분
 - 범위는 콜론(:)으로 구분하면 최솟값은 lo, 최댓값은 hi로 사용
[예] 10이하는 lo:10으로, 10초과는 10:hi
 - = 뒤에는 새로운 값을 설정



계산 보기



설치된 패키지 사용하기



자료

car 패키지를 먼저 호출한 후 recode 함수를 적용해보자.

```
> library(car)
> BMI$cbmi3 <- recode(BMI$bmi, "lo:18.5=1; 18.5:23=2; 23:25=3; 25:30=4; 30:hi=5")
> table(BMI$cbmi3)
 1   2   3   4
32 123 19   3
```

* 앞서서와 같은 결과를 얻음



계산 보기



설치된 패키지 사용하기

+ cbmi, cbmi2, cbmi3 사이에 차이가 있는지 확인하기

> BMI\$cbmi - BMI\$cbmi2

> BMI\$cbmi - BMI\$cbmi3

➤ 이와 같은 방법 등으로 이 결과가 177개의 0인지 확인 가능

변숫값 이름주기(labeling)

📖 개요

📖 factor 함수와 명목형 자료

📖 ordered 함수와 순서형 자료

📖 주의사항

Lorem ipsum dolor sit amet, ius an molestie facilisi erroribus, mutat natorum delectus ei vis. Has ornatus conclusionemque id, an vide molestatis sit. In etqui praesent sit. An vel agan porro comprehensan, ad ludus constituto nea, et ius utroque scaevola assumaverit.

Vis cu nodus nulla feugait, oratio facilisi ex usu, eili vitae sea te. Ea fabulas accusamus dissentias sea, facete tacinales definitiones et per. Nihil dicant mediocrem pro eu, no mei nostro sensibus platonem. Qui id sunno perpetus neglegantur. Vel ipsun novum copiosae ut. Quo et liber detracto probatus. Nam augue scribentur an. Sea oporteat percipitur incidereit at. Qui viris nemore an.



자료가 범주형(categorical)인 경우

- + 명목형(nominal)과 순서형(ordered)으로 구분

명목형

성별, 국적 등

순서형

불만족, 보통, 만족 등

- + 범주의 각 값에 대해(대개 코딩은 숫자로 되어 있음) 이름을 설정하면 분석결과를 얻을 때 좀 더 나은 형태의 결과를 얻을 수 있으며 빠른 해석이 가능함
- + R-언어에서는 범주형 자료에 대해서 명목형 또는 순서형으로 선언도 하고 각 값에 대해 이름(레이블; label)을 주는 함수를 제공함





R-언어의 함수 사용

+ 명목형인 경우 : factor 함수 사용

```
factor(x, levels, labels)
```

+ 순서형인 경우 : ordered 함수 사용

```
ordered(x, levels, labels)
```

➤ factor 와 ordered를 제외한 나머지 부분은 같음

+ 매개변수

- x : 명목형 또는 순서형으로 선언할 벡터의 이름
- levels : x의 가능한 값
- labels : x의 가능한 값에 대응하여 설정할 이름



개요



 변숫값에 대한 이름이 설정되지 않은 경우

 BMI 자료의 종교에 대한 도수분포표

```
> table(BMI$religion)
```

```
Bu C1 C2 No  
21 47 41 68
```

* 원자료의 값이 생성됨





factor 함수와 명목형 자료



```
> BMI$religion <- factor(BMI$religion,  
  levels=c("Bu", "C1", "C2", "No"),  
  labels=c("불교", "개신교", "가톨릭", "없음"))
```

```
> table(BMI$religion)
```

불교	개신교	가톨릭	없음
21	47	41	68





ordered 함수와 순서형 자료



```
> BMI$cbmi <- ordered(BMI$cbmi,  
  levels=seq(1,5),  
  labels=c("저체중", "정상", "과체중", "초기비만", "비만"))  
> table(BMI$cbmi)
```

저체중	정상	과체중	초기비만	비만
32	123	19	3	0

```
> BMI$cbmi2 <- ordered(BMI$cbmi2,  
  levels=seq(1,5),  
  labels=c("저체중", "정상", "과체중", "초기비만", "비만"))  
> table(BMI$cbmi2)
```

저체중	정상	과체중	초기비만	비만
32	123	19	3	0

주의사항



원자료가 숫자인 경우

+ factor 또는 ordered 함수로 변환된 경우에는 수치계산이 불가능함
(예: 평균, 분산 등을 계산하는 mean, var 함수 등의 호출)

➤ 수치계산은 원자료가 수치형(numeric)이거나, 논리형(TRUE=1, FALSE=0)인 경우에만 가능함

+ 평균 계산 예

➤ ordered 함수를 사용한 cbmi의 값은 1부터 5사이가 원래의 값이지만 이 값에 대한 평균을 계산해보면 에러 메시지가 출력됨

```
> mean(BMI$cbmi)
```

```
[1] NA
```

경고메시지(들) :

```
In mean.default(BMI$cbmi) :
```

인자가 수치형 또는 논리형이 아니므로 NA를 반환합니다



주의사항



5점 척도의 만족도 등



때로 평균 등의 수치계산이 필요한 경우가 있으므로 이런 경우 `as.numeric` 함수를 사용하여 수치형으로 바꾼 후 수치계산이 가능함

```
> mean(as.numeric(BMI$cbmi))  
[1] 1.960452
```



벡터의 원소에 이름 주기

📖 개요

📖 이름이 없는 경우

📖 이름이 설정된 경우

Lorem ipsum dolor sit amet, ius an molestie facilisi erroribus, mutat nalerum delectus ei vis. Has ornatus conclusionemque id, an videri molestatis sit. In etqui praesent sit. An vel agan porro comprehensan, ad ludus constituto nea, et ius utroque scaevola assumaverit.

Vis cu nodus nulla feugait, oratio facilisi in usu, eilit vitae sea te. Ea fabulas accusamus dissentias sea, facete tacinates definitiones et per. Nihil dicant mediocram pro eu, no mei nostro sensibus platonem. Qui id sunno perpetus neglegentur. Vel ipsun novum copiosae ut. Quo et liber detracto probatus. Nam augue scribentur an. Sea oporteat percipitur incidereat ab. Qui viris nemore an.



names 함수

- + R-객체(object)의 값에 대한 이름을 불러오거나 객체 x의 값에 대한 이름을 설정할 수 있는 names 함수가 R-언어의 내장함수로 제공됨



names 함수 사용법

```
names(x)  
names(x) <- value
```

- names(x) 경우 x의 이름이 출력됨
- names(x) <- value는 x의 이름에 value를 설정함
- value에 NULL을 설정하면 이름을 없앴



이름이 없는 경우



이름이 없는 상태의 xx에 대해서 xx를 인쇄하는 경우

```
> xx <- c(10,20,40,30)
> xx
[1] 10 20 40 30
```

* xx의 값만 출력됨



names 함수로 xx의 값에 대한 이름을 설정하는 경우

```
> names(xx) <- c("국어", "영어", "수학", "과학")
> xx
국어 영어 수학 과학
 10  20  40  30
```

* 이름이 설정되고, 출력 시 이름이 함께 출력됨



이름이 없는 경우



벡터의 일부분만 출력하는 경우

```
> xx[2:3]
```

```
영어 수학
```

```
20 40
```



이름이 설정된 경우



이름이 설정된 R 개체의 이름만 출력하는 경우



names 함수의 매개변수에 개체를 설정함

```
> names(xx)
```

```
[1] "국어" "영어" "수학" "과학"
```

➤ 일부 R 함수는 결과에 자동으로 이름이 붙여짐

[예] 자료에 대해서 table를 사용하여 빈도표를 얻는 경우 등
→ 자료의 값으로 자동으로 이름이 설정됨





이름이 설정된 경우



이전에 사용한 BMI의 성별(gender)변수로 빈도표를 만드는 경우

```
> table(BMI$gender)
```

```
F  M  
158 19
```

➤ 자료값 F, M으로 이름을 사용함



names 함수로 이름만 얻을 경우

빈도표의 헤더

```
> names(table(BMI$gender))
```

```
[1] "F" "M"
```