강

03 R 데이터 분석

R 데이터 처리 III

통계·데이터과학과 박서영 교수

한국방송통신대학교
Korea National Open University

학습목차

1. 변수값 전체 변환
2. 여러 단계 작업
3. 실전 예제

01 R 데이터 처리 III

2강 실습 코드

```
setwd("C:\\Users\\KNOU_stat\\Dropbox\\KNOU_강의개편\\고급R활용\\R_exercise")

library(readxl)
dat1<-read_excel("patients_2sheets.xlsx", sheet = "whole")

dat.base1<-dat1
colnames(dat1)
colnames(dat.base1)[4]<-"TX"
colnames(dat.base1)[5]<-"CA19.9"
dat.base2<-dat.base1[, c("age", "sex", "TX", "CA19.9", "Stage")]
dat.base3<-dat.base2[dat.base2$age>=40, ]
dat.base4<-dat.base3
dat.base4$CA19.9[dat.base4$CA19.9=="<1.0"]<-"1"
```

01 R 데이터 처리 III

2강 실습 코드

```
library(dplyr)

dat.dplyr1 <- dat1 %>% rename(TX=treatment, CA19.9='CA19-9')
dat.dplyr2<-dat.dplyr1 %>% select(age, sex, TX, CA19.9, Stage)
dat.dplyr3<-dat.dplyr2 %>% filter(age>=40)
dat.dplyr4<-dat.dplyr3 %>% mutate(CA19.9=replace(CA19.9, CA19.9=="<1.0", "1"))
```

더 간단히 줄일 수 있다.(2장 여러 단계 작업)

R 데이터 분석

1 변수값 전체 변환

01 R 데이터 처리 III

수치형 변수를 이분형 변수로 변환 - base R

(범주 2개로 나눌 때)

- ifelse( ) 함수
  - 어떤 조건을 만족하는지 여부에 따라 다른 값 부여

ifelse(*조건문, 참일 경우의 값, 거짓일 경우의 값*)

01 R 데이터 처리 III

수치형 변수를 범주형 변수로 변환 – base R

(범주 3개 이상으로 나눌 때)

- cut( ) 함수
  - 범주를 구분하는 cutoff 값을 breaks 옵션으로 넣는다.
  - 양 끝단 범주의 상한과 하한도 다 넣어야 한다.

cut(*수치형 변수 벡터*, breaks=*컷오프 값들의 벡터*)

- ifelse( ) 함수를 여러 개 겹쳐서 나눌 수도 있다.

01 R 데이터 처리 III

수치형 변수를 범주형 변수로 변환 - base R

```
dat.base5<-dat.base4
dat.base5$age50<-ifelse(dat.base5$age>=50, 1, 0)
dat.base5$age.grp<-cut(dat.base5$age,
                       breaks=c(0, 50, 60, 70, Inf))
```

01 R 데이터 처리 III

수치형 변수를 범주형 변수로 변환 – dplyr

- 범주 2개로 나눌 때
  - mutate( ) 함수 내에서 ifelse( ) 함수 사용

- 범주 3개 이상으로 나눌 때
  - mutate( ) 함수 내에서 cut( ) 함수 사용

01 R 데이터 처리 III

수치형 변수를 범주형 변수로 변환 - dplyr

```
dat.dplyr5<-dat.dplyr4 %>%
          mutate(age50=ifelse(age>=50, 1, 0))
dat.dplyr6<-dat.dplyr5 %>%
          mutate(age.grp=cut(age,
                    breaks=c(0, 50, 60, 70, Inf)))
```

01 R 데이터 처리 III

데이터 형태 바꾸기 – base R

- as.double( ), as.factor( ) 등의 함수 이용

```
summary(dat.base5)
dat.base5$CA19.9<-as.double(dat.base5$CA19.9)

dat.base5$sex<-as.factor(dat.base5$sex)
dat.base5$TX<-as.factor(dat.base5$TX)
dat.base5$Stage<-as.factor(dat.base5$Stage)
dat.base5$age50<-as.factor(dat.base5$age50)
```

01 R 데이터 처리 III

데이터 형태 바꾸기 – dplyr

- as.double( ), as.factor( ) 등의 함수 이용
- 변수 여러 개의 형태를 한꺼번에 바꿀 때는 mutate_at( ) 함수 이용

mutate_at(vars(*변수1, 변수2, ...*), *함수*)

```
summary(dat.dplyr6)
dat.dplyr7<-dat.dplyr6 %>%
          mutate(CA19.9=as.double(CA19.9))
dat.dplyr8<-dat.dplyr7 %>%
          mutate_at(vars(sex, TX, Stage, age50),
                    as.factor)
```

01 R 데이터 처리 III

summary 함수를 통한 점검

- 데이터 프레임을 summary( ) 함수 안에 넣으면
  - double형 변수: 다섯수치요약과 평균
  - factor형 변수: 각 범주별 데이터 개수
  - character형 변수: 벡터 길이 및 class, mode
- summary 함수의 출력결과가 변수의 형태를 올바르게 반영해야 데이터가 잘 처리되었다고 볼 수 있다.

```
summary(dat.base5)
summary(dat.dplyr8)
```

01 R 데이터 처리 III

summary 함수를 통한 점검

```
##       age         sex    TX          CA19.9          Stage  age50      age.grp
##  Min.   :41.00   0:16   0:22   Min.   :    1.00   1: 2   0: 5   (0,50]  : 7
##  1st Qu.:54.00   1:28   1:22   1st Qu.:   10.32   2: 6   1:39   (50,60] :13
##  Median :61.00                 Median :   20.10   3:12          (60,70] :18
##  Mean   :60.57                 Mean   : 1303.25   4:19          (70,Inf]: 6
##  3rd Qu.:67.00                 3rd Qu.:   53.73   5: 5
##  Max.   :79.00                 Max.   :48600.00
```

R 데이터 분석

2 여러 단계 작업

01 R 데이터 처리 III

여러 단계 작업 – base R

```
dat.base1<-dat1
colnames(dat.base1)[4]<-"TX"
colnames(dat.base1)[5]<-"CA19.9"
dat.base2<-dat.base1[, c("age", "sex", "TX", "CA19.9", "Stage")]
dat.base<-dat.base2[dat.base2$age>=40, ]
dat.base$CA19.9[dat.base$CA19.9=="<1.0"]<-"1"
dat.base$age50<-ifelse(dat.base$age>=50, 1, 0)
dat.base$age.grp<-cut(dat.base$age, breaks=c(0, 50, 60, 70, Inf))
dat.base$CA19.9<-as.double(dat.base$CA19.9)
dat.base$sex<-as.factor(dat.base$sex)
dat.base$TX<-as.factor(dat.base$TX)
dat.base$Stage<-as.factor(dat.base$Stage)
dat.base$age50<-as.factor(dat.base$age50)
```

01 R 데이터 처리 III

여러 단계 작업 - dplyr

- dplyr의 동사와 파이프 오퍼레이터를 이용하면 명령문이 짧아지고 가독성이 높아진다.
- 정확성, 재현성이 높은 분석을 할 수 있다.

```
데이터 프레임 %>% rename( ... ) %>%
                  select( ... ) %>%
                  filter( ... ) %>%
                  mutate( ... ) %>% ...
```

01 R 데이터 처리 III

여러 단계 작업 – dplyr

```
dat.dplyr<-dat1 %>% rename(TX=treatment,
                           CA19.9='CA19-9') %>%
                    select(age, sex, TX, CA19.9, Stage) %>%
                    filter(age>=40) %>%
                    mutate(CA19.9=replace(CA19.9, CA19.9=="<1.0", "1"),
                           age50=ifelse(age>=50, 1, 0),
                           age.grp=cut(age, breaks=c(0, 50, 60, 70, Inf)),
                           CA19.9=as.double(CA19.9)) %>%
                    mutate_at(vars(sex, TX, Stage, age50), as.factor)
```

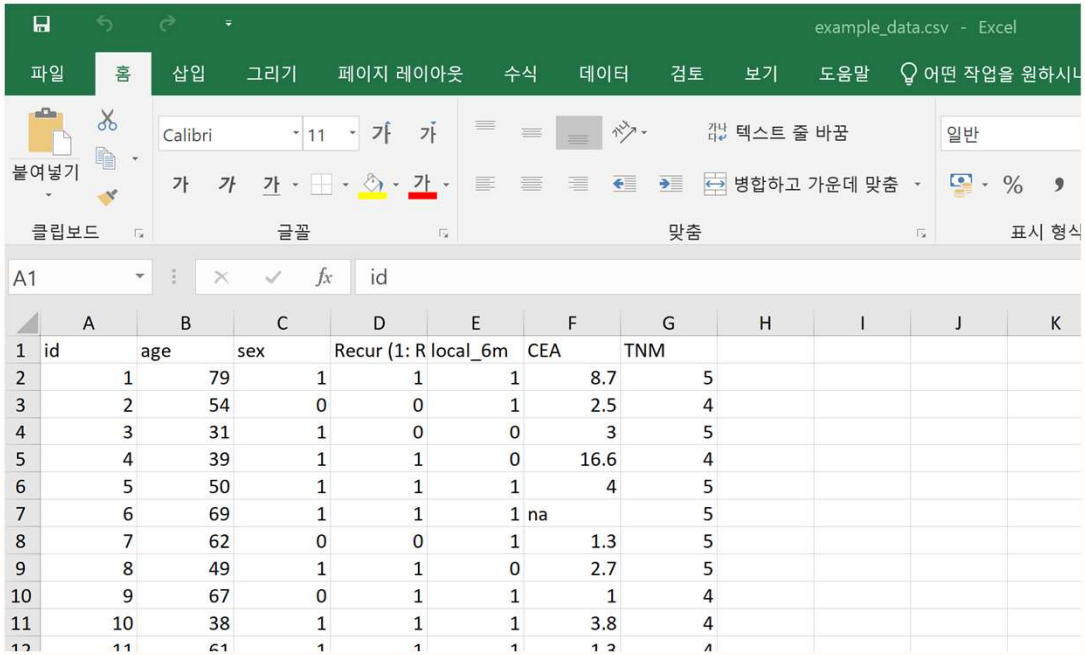
R 데이터 분석

3 실전 예제

01 R 데이터 처리 III

실전 예제 데이터

- example_data.csv
- https://github.com/biostat81/rda 에서 다운로드



	A	B	C	D	E	F	G
1	id	age	sex	Recur (1: R	local_6m	CEA	TNM
2	1	79	1	1	1	8.7	5
3	2	54	0	0	1	2.5	4
4	3	31	1	0	0	3	5
5	4	39	1	1	0	16.6	4
6	5	50	1	1	1	4	5
7	6	69	1	1	1	na	5
8	7	62	0	0	1	1.3	5
9	8	49	1	1	0	2.7	5
10	9	67	0	1	1	1	4
11	10	38	1	1	1	3.8	4

01 R 데이터 처리 III

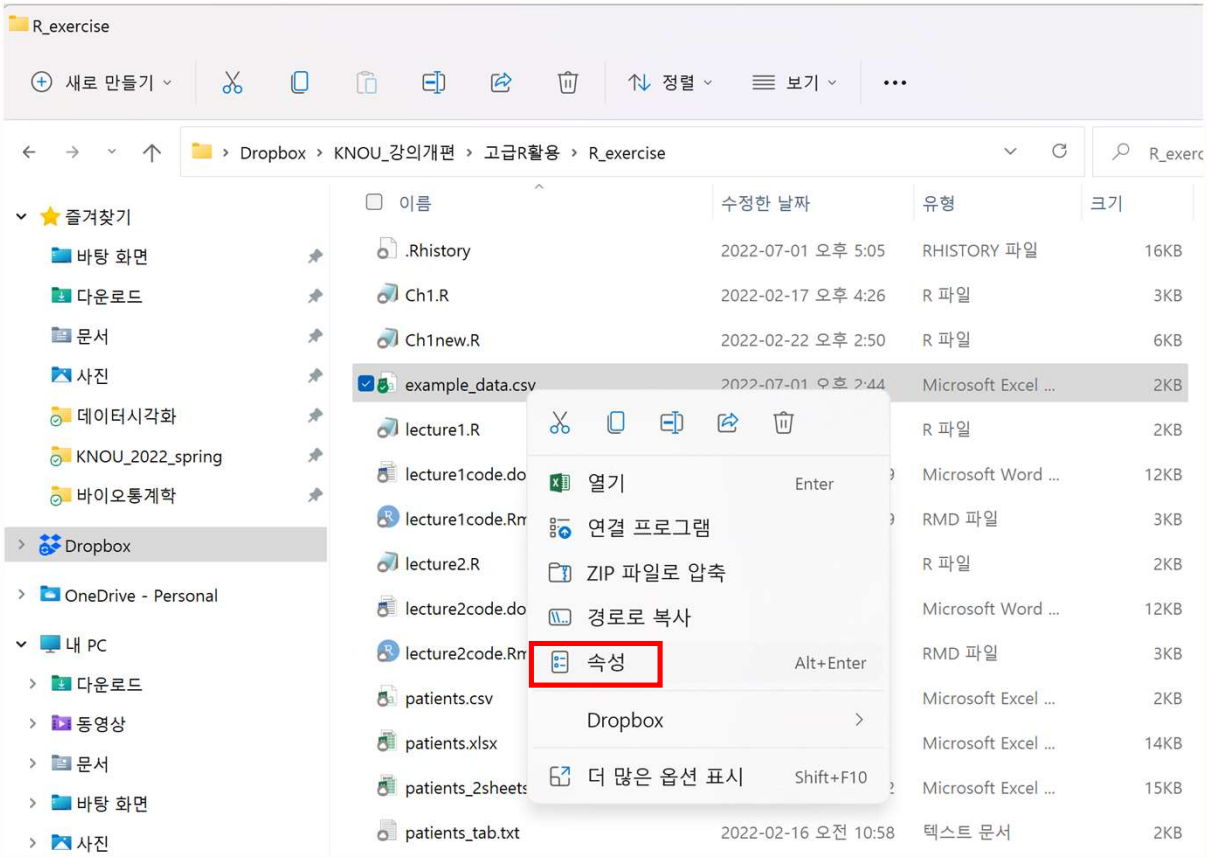
데이터 처리 실전 팁

- 엑셀에서 데이터를 다루는 것을 되도록 피한다.
  - 실수로 열이나 행을 뒤바꾸거나 잘못된 값을 입력하기 쉽다.
  - 데이터 처리는 가능한 R에서 수행한다.

- 데이터를 받자마자 읽기 전용으로 바꾼다.
  - 탐색기에서 파일 이름 우클릭>속성>읽기 전용 체크

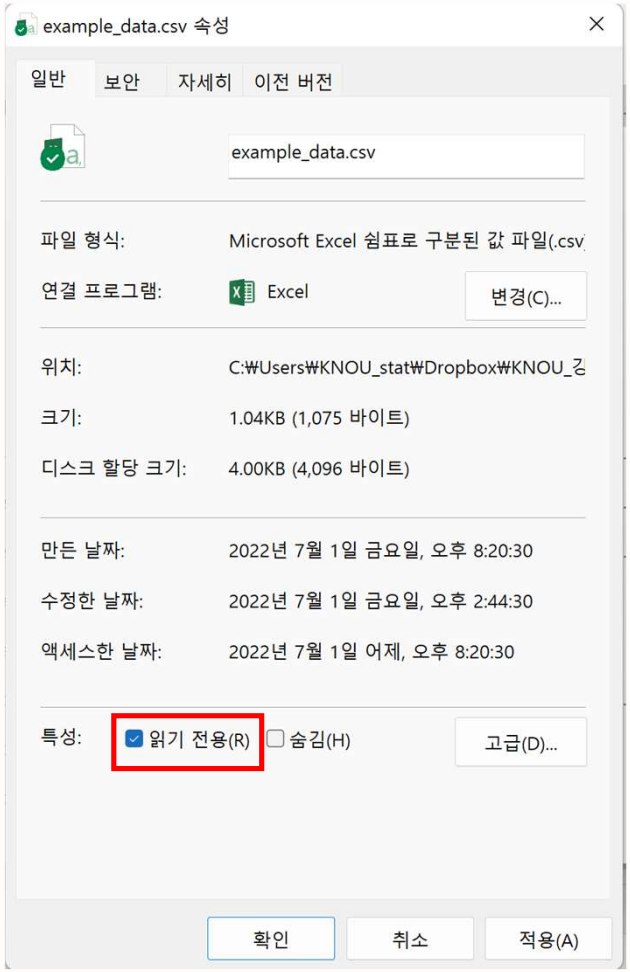
01 R 데이터 처리 III

데이터 처리 실전 팁



01 R 데이터 처리 III

데이터 처리 실전 팁



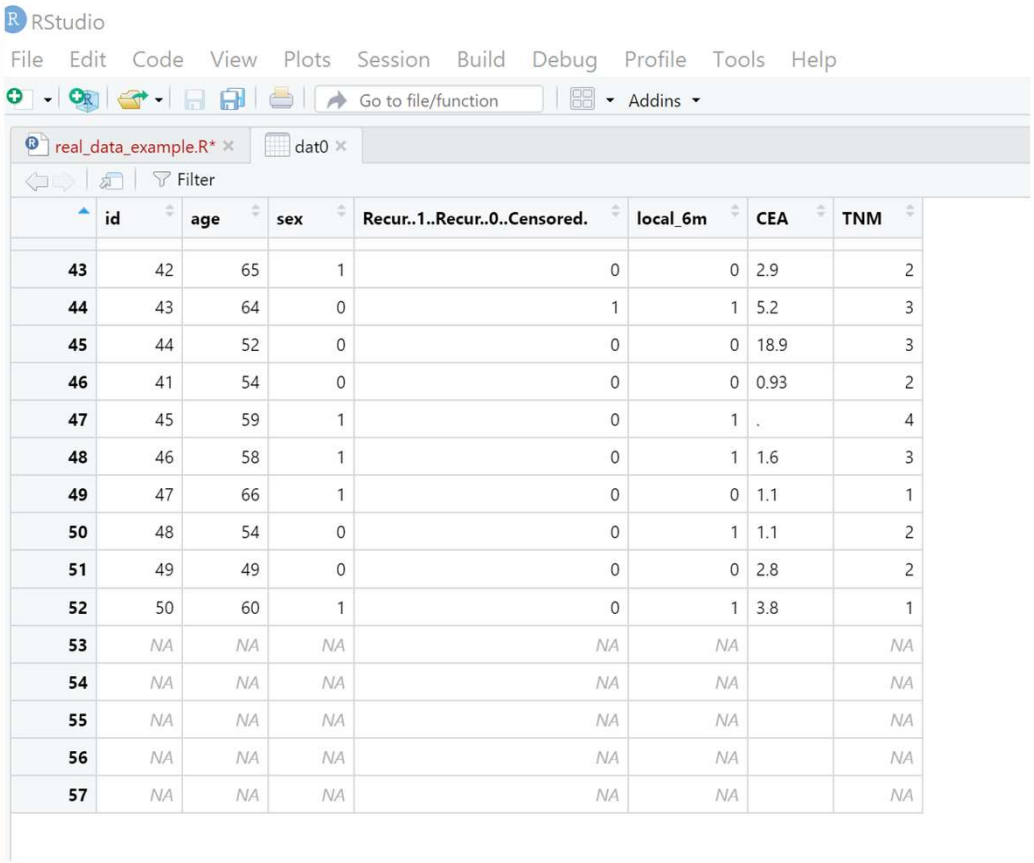
01 R 데이터 처리 III

데이터 처리 실전 팁

- R에 읽어 들인 후, 원 데이터와 반드시 비교한다.
- 한글이 포함된 데이터의 경우 인코딩 문제
- 변수 이름이 제대로 들어가 있는가
- 반드시 '제일 오른쪽 열' 확인
- 반드시 '제일 아래쪽 행' 확인

01 R 데이터 처리 III

데이터 처리 실전 팁



	id	age	sex	Recur..1..Recur..0..Censored.	local_6m	CEA	TNM
43	42	65	1	0	0	2.9	2
44	43	64	0	1	1	5.2	3
45	44	52	0	0	0	18.9	3
46	41	54	0	0	0	0.93	2
47	45	59	1	0	1	.	4
48	46	58	1	0	1	1.6	3
49	47	66	1	0	0	1.1	1
50	48	54	0	0	1	1.1	2
51	49	49	0	0	0	2.8	2
52	50	60	1	0	1	3.8	1
53	NA	NA	NA	NA	NA		NA
54	NA	NA	NA	NA	NA		NA
55	NA	NA	NA	NA	NA		NA
56	NA	NA	NA	NA	NA		NA
57	NA	NA	NA	NA	NA		NA

01 R 데이터 처리 III

데이터 처리 실전 팁

- 각 케이스가 하나의 행에 대응되는지 확인한다.
  - 케이스 ID에 해당하는 변수의 값이 모두 unique한지 확인
    - ✓ unique( ) 함수, length( ) 함수 이용
    - ✓ (하나의 케이스를 의도적으로 여러 개의 행에 입력한 경우도 있을 수 있다)
  - ID가 unique하지 않은 경우, 중복된 행을 지워준다.
    - ✓ 중복된 행을 찾을 때 duplicated( ) 함수 이용
    - ✓ 모든 변수에 대해 완벽히 같은 값이 들어간 행: distinct( ) 함수 이용
    - ✓ ID는 같으나 다른 변수들 값은 다른 경우: 경위를 알아보고 오류에 의한 경우 잘못된 행을 지워준다.

01 R 데이터 처리 III

데이터 처리 실전 팁

```
setwd("C:\\Users\\KNOU_stat\\Dropbox\\KNOU_강의개편\\고급R활용\\R_exercise")

dat0<-read.csv("example_data.csv")
dat0<-read.csv("example_data.csv", nrow=52)

library(dplyr)

length(unique(dat0$id))
which(duplicated(dat0$id))

dat1<-dat0 %>% distinct()
length(unique(dat1$id))
which(duplicated(dat1$id))
dat1$id[which(duplicated(dat1$id))]
dat1[dat1$id==41, ]

dat2<-dat1[-41, ]
length(unique(dat2$id))
```

01 R 데이터 처리 III

데이터 처리 실전 팁

❖ 결측값이 입력된 방식을 확인하고 NA로 수정한다.

- 문자나 특수 문자로 입력된 경우: 비교적 쉽게 확인 가능
- 999, -1 등의 숫자로 입력된 경우: 지나치기 쉬우므로 반드시 육안으로 확인, 요약통계량 확인
- replace( ) 등의 함수로 결측은 NA로 바꾸어준다.

01 R 데이터 처리 III

데이터 처리 실전 팁

- 범주형 변수는 반드시 factor 형으로 변환한다.
  - 범주형 변수를 double 형으로 저장한 채로 사용할 경우 잘못된 분석 결과를 얻을 가능성이 높다.

- R에서 분석할 때 알 수 없는 이유로 오류가 나오는 경우: 함수의 input으로 넣는 객체의 형태가 올바르게 설정되어 있는지 확인해본다.

01 R 데이터 처리 III

데이터 처리 실전 팁

- 반드시 요약통계량을 확인한다.
  - 이 단계를 건너 뛰고 본격적으로 분석을 할 경우 뒤늦게 데이터 오류를 발견하여 많은 시간을 낭비할 수 있다.
  - 최소값, 최대값 등이 예상 범위 내인지 확인
  - 각 변수의 형태가 올바르게 설정되었는지 확인
  - summary( ) 함수 이용
  - 요약통계량을 편리하게 출력해주는 패키지: tableone

01 R 데이터 처리 III

데이터 처리 실전 팁

```
summary(dat2)

dat3<-dat2 %>% rename(Recur=Recur..1..Recur..0..Censored.) %>%
  mutate(CEA=as.double(replace(CEA, CEA=="na" | CEA==".", NA)),
         abnormal=ifelse(CEA>=100, 1, 0),
         CEA.group=cut(CEA, breaks=c(0, 5, 10, 100, Inf))
  ) %>%
  mutate_at(vars(sex, Recur, local_6m, TNM, abnormal),
            as.factor)

summary(dat3)
```

정리하기

- 수치형 변수를 범주형 변수로 변환할 때에는 ifelse( ), cut( ) 함수를 사용할 수 있다.
- 여러 개의 변수에 대해 같은 작업을 수행할 때에는 dplyr패키지의 mutate_at( ) 함수를 사용할 수 있다.
- dplyr의 동사와 파이프 연산자를 이용하면 명령문이 짧아지고 가독성이 높아진다.
- 데이터를 처리할 때에는, 정확성과 재현성을 높이기 위해 최선을 다하고 반드시 처리 결과의 요약통계량을 확인한다.

04강

다음시간 안내

통계 계산 I

수고하셨습니다!