Data Frames and Contingency Tables

coop711

2015년 5월 5일

Data for Practice

한국 갤럽의 1987년 대선 여론조사 자료를 예제로 활용한다.

`table` 구조를 가지므로 우선 `matrix` 로 읽어들인다. 변수명을 사전에 입력하는 방법에 유의하라.

```
dim.names<-list(Religion=c("Buddhism","Protestant","Catholic","None"), Candidat
es=c("Roh","YS","DJ","JP"))
poll.87<-matrix(c(289, 84, 26, 361, 154, 139, 53, 292, 126, 145, 57, 287, 61, 2
9, 11, 80), nrow=4, ncol=4, dimnames=dim.names)
poll.87
```

```
##             Candidates
## Religion     Roh  YS  DJ JP
##   Buddhism   289 154 126 61
##   Protestant  84 139 145 29
##   Catholic    26  53  57 11
##   None       361 292 287 80
```

```
class(poll.87)
```

```
## [1] "matrix"
```

```
str(poll.87)
```

```
##  num [1:4, 1:4] 289 84 26 361 154 139 53 292 126 145 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ Religion  : chr [1:4] "Buddhism" "Protestant" "Catholic" "None"
##   ..$ Candidates: chr [1:4] "Roh" "YS" "DJ" "JP"
```

`table` 구조로 강제 변환한다.

```
poll.87.tbl<-as.table(poll.87)
str(poll.87.tbl)
```

```
##  table [1:4, 1:4] 289 84 26 361 154 139 53 292 126 145 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ Religion  : chr [1:4] "Buddhism" "Protestant" "Catholic" "None"
##   ..$ Candidates: chr [1:4] "Roh" "YS" "DJ" "JP"
```

총 몇명이 참여하고 각각의 소계는 얼마인지 살피려면 `addmargins()` 를 적용한다.

```
addmargins(poll.87.tbl)
```

```
##             Candidates
## Religion       Roh   YS   DJ   JP  Sum
##   Buddhism     289  154  126   61  630
##   Protestant    84  139  145   29  397
##   Catholic      26   53   57   11  147
##   None         361  292  287   80 1020
##   Sum          760  638  615  181 2194
```

후보별 지지도를 알아내려면 `prop.table()` 와 `addmargins()` 를 복합적으로 활용한다.

```
options(digits=3)
addmargins(prop.table(poll.87.tbl))
```

```
##             Candidates
## Religion          Roh      YS      DJ      JP     Sum
##   Buddhism    0.13172 0.07019 0.05743 0.02780 0.28715
##   Protestant  0.03829 0.06335 0.06609 0.01322 0.18095
##   Catholic    0.01185 0.02416 0.02598 0.00501 0.06700
##   None        0.16454 0.13309 0.13081 0.03646 0.46490
##   Sum         0.34640 0.29079 0.28031 0.08250 1.00000
```

각 후보의 종교별 지지도를 알고 싶다면 `margin = 1` 을 적용한다.

```
options(digits=3)
prop.table(poll.87.tbl, margin=1)
```

```
##             Candidates
## Religion         Roh     YS     DJ     JP
##   Buddhism    0.4587 0.2444 0.2000 0.0968
##   Protestant  0.2116 0.3501 0.3652 0.0730
##   Catholic    0.1769 0.3605 0.3878 0.0748
##   None        0.3539 0.2863 0.2814 0.0784
```

Contingency Table to Data Frame with Counts

종교와 후보의 각 조합에 대하여 `Counts` 를 한 변수로 갖는 data frame으로 전환하려면 `as.data.frame()` 을 사용한다. 이때 `default.stringsAsFactors()` 가 `TRUE` 일 경우가 대부분이므로 character들의 순서가 적합한지 살피고 적용하여야 한다. 순서가 맞지 않으면 세종대왕의 여론조사에서 했던 것처럼 `stringsAsFactors=FALSE` 로 하고, `factor()` 를 써서 나중에 전환해 주어야 한다.

```
poll.87.df<-as.data.frame(poll.87.tbl)
str(poll.87.df)
```

```
## 'data.frame':    16 obs. of  3 variables:
##  $ Religion  : Factor w/ 4 levels "Buddhism","Protestant",..: 1 2 3 4 1 2 3
4 1 2 ...
##  $ Candidates: Factor w/ 4 levels "Roh","YS","DJ",..: 1 1 1 1 2 2 2 2 3 3
...
##  $ Freq      : num  289 84 26 361 154 139 53 292 126 145 ...
```

구조에서 살필 수 있다시피 각 변수의 속성이 잘 보전되고 있음을 알 수 있다.

Data Frame with Counts to Contingency Table

이 data frame 으로부터 분할표(contingency table)을 구하는 것은 `xtabs()` 활용.

```
poll.87.tbl.2<-xtabs(Freq ~ Religion + Candidates, data = poll.87.df)
poll.87.tbl.2
```

```
##             Candidates
## Religion     Roh  YS  DJ  JP
##   Buddhism   289 154 126  61
##   Protestant  84 139 145  29
##   Catholic    26  53  57  11
##   None       361 292 287  80
```

행과 열의 총괄 명칭이 덧붙여졌음을 알 수 있다.

Data Frame with Counts to Data Frame with Cases

2194명 각각에 대한 case가 주어지는 data frame 으로 전환하려면 `poll.87.df` 의 각 행을 그 행의 `Counts` 갯수만큼 반복하면 되므로 먼저 각 갯수만큼의 index를 확보한다.

```
index.cases<-rep(1:nrow(poll.87.df), poll.87.df[, "Freq"])
```

`poll.87.df` 의 1, 2열의 각 행을 `Freq` 만큼 반복하고 세번째 열은 필요하지 않으므로 제외하면 된다. 이 과정이 `crimtab` 테이블을 long format으로 밖는 과정에서 `apply()` 를 사용한 것보다 나은 이유는 `class` 를 보전하기 때문이다. 여기서 `Religion` 과 `Candidates` 가 갖고 있는 `factor` 가 그대로 이어진다.

```
poll.87.cases<-poll.87.df[index.cases, 1:2]
str(poll.87.cases)
```

```
## 'data.frame':    2194 obs. of  2 variables:
##  $ Religion  : Factor w/ 4 levels "Buddhism","Protestant",..: 1 1 1 1 1 1 1
1 1 1 ...
##  $ Candidates: Factor w/ 4 levels "Roh","YS","DJ",..: 1 1 1 1 1 1 1 1 1 1
...
```

```
head(poll.87.cases, n=10)
```

```
##     Religion Candidates
## 1   Buddhism        Roh
## 1.1 Buddhism        Roh
## 1.2 Buddhism        Roh
## 1.3 Buddhism        Roh
## 1.4 Buddhism        Roh
## 1.5 Buddhism        Roh
## 1.6 Buddhism        Roh
## 1.7 Buddhism        Roh
## 1.8 Buddhism        Roh
## 1.9 Buddhism        Roh
```

```
tail(poll.87.cases, n=10)
```

```
##       Religion Candidates
## 16.70     None         JP
## 16.71     None         JP
## 16.72     None         JP
## 16.73     None         JP
## 16.74     None         JP
## 16.75     None         JP
## 16.76     None         JP
## 16.77     None         JP
## 16.78     None         JP
## 16.79     None         JP
```

From Cases to Table

각 Case를 모아 분할표로 만드는 과정은 `table()` 의 본래 기능이다. `poll.87.cases` 의 두 변수가 모두 `factor` 속성을 보전하고 있기 때문에 가능한 일이다.

```
poll.87.tbl.3<-table(poll.87.cases$Religion, poll.87.cases$Candidates)
poll.87.tbl.3
```

```
##
##              Roh  YS  DJ  JP
##   Buddhism   289 154 126  61
##   Protestant  84 139 145  29
##   Catholic    26  53  57  11
##   None       361 292 287  80
```

테이블로 만들면서 Religion과 Candidates가 사라진 것을 다시 채우려면,

```
poll.87.tbl.4<-table(Religion=poll.87.cases$Religion, Candidates=poll.87.case
s$Candidates)
poll.87.tbl.4
```

```
##             Candidates
## Religion     Roh  YS  DJ  JP
##   Buddhism   289 154 126  61
##   Protestant  84 139 145  29
##   Catholic    26  53  57  11
##   None       361 292 287  80
```

분할표와 data frame 간의 자료 전환은 기본적으로 위의 과정을 순환한다.

Exercise with UCBAdmissions

```
str(UCBAdmissions)
```

```
##  table [1:2, 1:2, 1:6] 512 313 89 19 353 207 17 8 120 205 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ Admit : chr [1:2] "Admitted" "Rejected"
##   ..$ Gender: chr [1:2] "Male" "Female"
##   ..$ Dept  : chr [1:6] "A" "B" "C" "D" ...
```

```
ftable(UCBAdmissions)
```

```
##                 Dept   A   B   C   D   E   F
## Admit    Gender
## Admitted Male        512 353 120 138  53  22
##          Female       89  17 202 131  94  24
## Rejected Male        313 207 205 279 138 351
##          Female       19   8 391 244 299 317
```

3차원 array 구조를 갖고 있는 자료구조이므로 Counts를 갖는 data frame 으로 전환하려면,

```
UCBAdmissions.df<-as.data.frame(UCBAdmissions)
str(UCBAdmissions.df)
```

```
## 'data.frame':    24 obs. of  4 variables:
##  $ Admit : Factor w/ 2 levels "Admitted","Rejected": 1 2 1 2 1 2 1 2 1 2 ...
##  $ Gender: Factor w/ 2 levels "Male","Female": 1 1 2 2 1 1 2 2 1 1 ...
##  $ Dept  : Factor w/ 6 levels "A","B","C","D",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ Freq  : num  512 313 89 19 353 207 17 8 120 205 ...
```

```
UCBAdmissions.df
```

```
##       Admit Gender Dept Freq
## 1  Admitted   Male    A  512
## 2  Rejected   Male    A  313
## 3  Admitted Female    A   89
## 4  Rejected Female    A   19
## 5  Admitted   Male    B  353
## 6  Rejected   Male    B  207
## 7  Admitted Female    B   17
## 8  Rejected Female    B    8
## 9  Admitted   Male    C  120
## 10 Rejected   Male    C  205
## 11 Admitted Female    C  202
## 12 Rejected Female    C  391
## 13 Admitted   Male    D  138
## 14 Rejected   Male    D  279
## 15 Admitted Female    D  131
## 16 Rejected Female    D  244
## 17 Admitted   Male    E   53
## 18 Rejected   Male    E  138
## 19 Admitted Female    E   94
## 20 Rejected Female    E  299
## 21 Admitted   Male    F   22
## 22 Rejected   Male    F  351
## 23 Admitted Female    F   24
## 24 Rejected Female    F  317
```

`xtabs` 를 활용하여 몇 가지 사실을 파악하면,

```
xtabs(Freq ~ Admit, data = UCBAdmissions.df)
```

```
## Admit
## Admitted Rejected
##     1755     2771
```

```
options(digits=3)
prop.table(xtabs(Freq ~ Admit, data = UCBAdmissions.df))
```

```
## Admit
## Admitted Rejected
##    0.388    0.612
```

전체적인 입학허가율은 38.8%이었다. 남녀별 합격율을 비교하려면,

```
xtabs(Freq ~ Admit+Gender, data = UCBAdmissions.df)
```

```
##           Gender
## Admit      Male Female
##   Admitted 1198    557
##   Rejected 1493   1278
```

```
prop.table(xtabs(Freq ~ Admit+Gender, data = UCBAdmissions.df), margin=2)
```

```
##           Gender
## Admit       Male Female
##   Admitted 0.445  0.304
##   Rejected 0.555  0.696
```

남성들의 입학허가율이 높게 나타난다. 소송의 근거가 된 사실이다.

`ftable()` 이 근본적으로 매트릭스 구조임을 상기하면서 주요 학과별로 입학허가율을 비교하면,

```
ftable(xtabs(Freq ~ Gender+Admit+Dept, data = UCBAdmissions.df))
```

```
##                 Dept   A   B   C   D   E   F
## Gender Admit
## Male   Admitted      512 353 120 138  53  22
##        Rejected      313 207 205 279 138 351
## Female Admitted       89  17 202 131  94  24
##        Rejected       19   8 391 244 299 317
```

```
prop.table(ftable(xtabs(Freq ~ Gender+Admit+Dept, data = UCBAdmissions.df))
[1:2,], margin=2)
```

```
##       [,1] [,2]  [,3]  [,4]  [,5]  [,6]
## [1,] 0.621 0.63 0.369 0.331 0.277 0.059
## [2,] 0.379 0.37 0.631 0.669 0.723 0.941
```

```
prop.table(ftable(xtabs(Freq ~ Gender+Admit+Dept, data = UCBAdmissions.df))
[3:4,], margin=2)
```

```
##       [,1] [,2]  [,3]  [,4]  [,5]   [,6]
## [1,] 0.824 0.68 0.341 0.349 0.239 0.0704
## [2,] 0.176 0.32 0.659 0.651 0.761 0.9296
```

학과별로 볼 때는 여성들의 입학허가율이 더 높거나 최소한 비슷함을 알 수 있다. `prop.table` 을 사용하는 과정에서 빠진 변수명을 굳이 살리고 싶다면,

```
dim.names.UCB<-dimnames(UCBAdmissions)[c("Admit", "Dept")]
dim.names.UCB
```

```
## $Admit
## [1] "Admitted" "Rejected"
##
## $Dept
## [1] "A" "B" "C" "D" "E" "F"
```

```
male.admissions<-prop.table(ftable(xtabs(Freq ~ Gender+Admit+Dept, data = UCBAd
missions.df))[1:2,], margin=2)
female.admissions<-prop.table(ftable(xtabs(Freq ~ Gender+Admit+Dept, data = UCB
Admissions.df))[3:4,], margin=2)
```

남자들의 경우

```
matrix(data=male.admissions, nrow=2, ncol=6, dimnames=dim.names.UCB)
```

```
##           Dept
## Admit          A    B     C     D     E     F
##   Admitted 0.621 0.63 0.369 0.331 0.277 0.059
##   Rejected 0.379 0.37 0.631 0.669 0.723 0.941
```

여자들의 경우

```
matrix(data=female.admissions, nrow=2, ncol=6, dimnames=dim.names.UCB)
```

```
##           Dept
## Admit          A    B     C     D     E      F
##   Admitted 0.824 0.68 0.341 0.349 0.239 0.0704
##   Rejected 0.176 0.32 0.659 0.651 0.761 0.9296
```

앞에서 파악한 사실을 확인할 수 있다.

이 자료를 long format data frame으로 바꾸려면,

```
index.UCB<-rep(1:nrow(UCBAdmissions.df), UCBAdmissions.df[, "Freq"])
UCBAdmissions.cases<-UCBAdmissions.df[index.UCB, 1:3]
str(UCBAdmissions.cases)
```

```
## 'data.frame':    4526 obs. of  3 variables:
##  $ Admit : Factor w/ 2 levels "Admitted","Rejected": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Gender: Factor w/ 2 levels "Male","Female": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Dept  : Factor w/ 6 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
head(UCBAdmissions.cases)
```

```
##        Admit Gender Dept
## 1   Admitted   Male    A
## 1.1 Admitted   Male    A
## 1.2 Admitted   Male    A
## 1.3 Admitted   Male    A
## 1.4 Admitted   Male    A
## 1.5 Admitted   Male    A
```

```
tail(UCBAdmissions.cases)
```

```
##           Admit Gender Dept
## 24.311 Rejected Female    F
## 24.312 Rejected Female    F
## 24.313 Rejected Female    F
## 24.314 Rejected Female    F
## 24.315 Rejected Female    F
## 24.316 Rejected Female    F
```

여기서 다시 분할표를 만들고, data frame으로 전환하는 일을 할 수 있다.

```
table(UCBAdmissions.cases$Admit)
```

```
##
## Admitted Rejected
##     1755     2771
```

```
table(UCBAdmissions.cases$Admit, UCBAdmissions.cases$Gender)
```

```
##
##             Male Female
##   Admitted 1198    557
##   Rejected 1493   1278
```

```
ftable(table(UCBAdmissions.cases$Gender, UCBAdmissions.cases$Admit, UCBAdmissio
ns.cases$Dept))
```

```
##                   A   B   C   D   E   F
##
## Male   Admitted  512 353 120 138  53  22
##        Rejected  313 207 205 279 138 351
## Female Admitted   89  17 202 131  94  24
##        Rejected   19   8 391 244 299 317
```

뒷 마무리

```
save(file="tbl_df.rda", list=ls())
savehistory("tbl_df.Rhistory")
```