Data Frames and Contingency Tables

coop711

2015년 5월 5일

Data for Practice

한국 갤럽의 1987년 대선 여론조사 자료를 예제로 활용한다. 김용옥 선생이 '나는 불교를 이렇게 본다'에서 인용한 한국갤럽의 발표자료는

표 1: 1987년도 대통령선거 종교별 후보 지지도 조사(단위 %)

	노태우	김영삼	김대중	김종필	기타
불교(631)	45.8	24.4	20.0	9.7	0.2
개신교(398)	21.1	34.9	36.4	7.3	0.3
카톨릭(152)	17.1	34.2	37.5	7.2	3.3
무(1025)	35.2	28.5	28.0	7.8	0.5
계(2206)					

우선 이 자료를 입력하여 파일로 저장하자. 종교별 표본수를 먼저 벡터로 저장하고, 이름을 정한다.

```
N.Religion<-c(631, 398, 152, 1025)
names(N.Religion)<-c("Buddhism", "Protestant", "Catholic", "None")
```

`table` 구조, 사실상 `matrix` 로 읽어들인다.

```
poll.87<-matrix(c(45.8, 21.1, 17.1, 35.2, 24.4, 34.9, 34.2, 28.5, 20.0, 36.4, 3
7.5, 28.0, 9.7, 7.3, 7.2, 7.8, 0.2, 0.3, 3.3, 0.5), nrow=4, ncol=5)
poll.87
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 45.8 24.4 20.0  9.7  0.2
## [2,] 21.1 34.9 36.4  7.3  0.3
## [3,] 17.1 34.2 37.5  7.2  3.3
## [4,] 35.2 28.5 28.0  7.8  0.5
```

필요한 변수명을 넣는다.

```
poll.names<-list(Religion=names(N.Religion), Candidates=c("Roh", "YS", "DJ", "J
P", "etc"))
dimnames(poll.87)<-poll.names
poll.87
```

```
##             Candidates
## Religion      Roh   YS   DJ  JP etc
##   Buddhism   45.8 24.4 20.0 9.7 0.2
##   Protestant 21.1 34.9 36.4 7.3 0.3
##   Catholic   17.1 34.2 37.5 7.2 3.3
##   None       35.2 28.5 28.0 7.8 0.5
```

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

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

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

```
##  num [1:4, 1:5] 45.8 21.1 17.1 35.2 24.4 34.9 34.2 28.5 20 36.4 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ Religion  : chr [1:4] "Buddhism" "Protestant" "Catholic" "None"
##   ..$ Candidates: chr [1:5] "Roh" "YS" "DJ" "JP" ...
```

과연, 종교별 후보지지도의 합이 100%인지 `addmargins()` 를 이용하여 확인하되, 불필요한 column sum을 생략하기 위하여 `[1:4, ]` 적용.

```
addmargins(poll.87)[1:4, ]
```

```
##             Candidates
## Religion      Roh   YS   DJ  JP etc   Sum
##   Buddhism   45.8 24.4 20.0 9.7 0.2 100.1
##   Protestant 21.1 34.9 36.4 7.3 0.3 100.0
##   Catholic   17.1 34.2 37.5 7.2 3.3  99.3
##   None       35.2 28.5 28.0 7.8 0.5 100.0
```

카톨릭 표본의 소계가 잘 맞지 않는 것 확인. 종교별 표본수효 함께 출력.

```
options(digits=3)
cbind(poll.87, N.Religion)
```

```
##             Roh   YS   DJ  JP etc N.Religion
## Buddhism   45.8 24.4 20.0 9.7 0.2        631
## Protestant 21.1 34.9 36.4 7.3 0.3        398
## Catholic   17.1 34.2 37.5 7.2 3.3        152
## None       35.2 28.5 28.0 7.8 0.5       1025
```

`N.Religion` 의 이름을 표본 크기를 상징하는 `Size` 로 교체

```
cbind(poll.87, Size=N.Religion)
```

```
##             Roh   YS   DJ  JP etc Size
## Buddhism   45.8 24.4 20.0 9.7 0.2  631
## Protestant 21.1 34.9 36.4 7.3 0.3  398
## Catholic   17.1 34.2 37.5 7.2 3.3  152
## None       35.2 28.5 28.0 7.8 0.5 1025
```

종교별 표본에서 각 후보의 지지자수를 계산한다. 사람의 수를 세는 만큼 소숫점 이하를 나오지 않도록 한다.

```
options(digits=0)
poll.87.counts<-N.Religion*poll.87/100
poll.87.counts
```

```
##             Candidates
## Religion     Roh  YS  DJ JP etc
##   Buddhism   289 154 126 61   1
##   Protestant  84 139 145 29   1
##   Catholic    26  52  57 11   5
##   None       361 292 287 80   5
```

종교별 소계, 후보별 지지자 소계를 `addmargins()` 를 이용하여 계산한다.

```
addmargins(poll.87.counts)
```

```
##             Candidates
## Religion     Roh  YS  DJ  JP etc  Sum
##   Buddhism   289 154 126  61   1  632
##   Protestant  84 139 145  29   1  398
##   Catholic    26  52  57  11   5  151
##   None       361 292 287  80   5 1025
##   Sum        760 637 615 181  13 2206
```

반올림으로 인하여 불교와 카톨릭에서 1명씩의 차이가 나지만 전체 표본의 크기와는 잘 맞는 점을 확인.

후보별 지지율을 계산하기 위하여 `prop.table()` 을 적용.

```
options(digits=2)
prop.table(poll.87.counts)
```

```
##             Candidates
## Religion       Roh    YS    DJ    JP     etc
##   Buddhism   0.131 0.070 0.057 0.028 0.00057
##   Protestant 0.038 0.063 0.066 0.013 0.00054
##   Catholic   0.012 0.024 0.026 0.005 0.00227
##   None       0.164 0.132 0.130 0.036 0.00232
```

후보별 지지율 소계를 `addmargins()` 를 이용하여 계산

```
addmargins(prop.table(poll.87.counts))
```

```
##             Candidates
## Religion       Roh    YS    DJ    JP     etc   Sum
##   Buddhism   0.131 0.070 0.057 0.028 0.00057 0.286
##   Protestant 0.038 0.063 0.066 0.013 0.00054 0.180
##   Catholic   0.012 0.024 0.026 0.005 0.00227 0.068
##   None       0.164 0.132 0.130 0.036 0.00232 0.465
##   Sum        0.344 0.289 0.279 0.082 0.00571 1.000
```

결론은 마지막 행에 있으므로,

```
addmargins(prop.table(poll.87.counts))[5, 1:6]*100
```

```
##    Roh     YS     DJ     JP    etc    Sum
##  34.45  28.88  27.89   8.21   0.57 100.00
```

Matrix to Table

`table` 구조로 강제 변환한다.

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

```
##  table [1:4, 1:5] 289 84 26 361 154 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ Religion  : chr [1:4] "Buddhism" "Protestant" "Catholic" "None"
##   ..$ Candidates: chr [1:5] "Roh" "YS" "DJ" "JP" ...
```

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 `FALSE` 일지라도 직접 `stringsAsFactors=FALSE` 라고 명시하지 않으면 `chr` 을 `factor` 로 변환하는 경우가 있으므로 character들의 순서가 적합한지 살피고 적용하여야 한다. 순서가 맞지 않으면 세종대왕의 여론조사에서 했던 것처럼 `stringsAsFactors=FALSE` 로 하고, `factor()` 를 써서 나중에 전환해 주어야 한다.

```
default.stringsAsFactors()
```

```
## [1] FALSE
```

```
options(digits=0)
poll.87.df<-as.data.frame(poll.87.tbl)
str(poll.87.df)
```

```
## 'data.frame':    20 obs. of  3 variables:
##  $ Religion  : Factor w/ 4 levels "Buddhism","Protestant",..: 1 2 3 4 1 2 3
4 1 2 ...
##  $ Candidates: Factor w/ 5 levels "Roh","YS","DJ",..: 1 1 1 1 2 2 2 2 3 3
...
##  $ Freq      : num  289 84 26 361 154 ...
```

```
poll.87.df
```

```
##      Religion Candidates Freq
## 1    Buddhism        Roh  289
## 2  Protestant        Roh   84
## 3    Catholic        Roh   26
## 4        None        Roh  361
## 5    Buddhism         YS  154
## 6  Protestant         YS  139
## 7    Catholic         YS   52
## 8        None         YS  292
## 9    Buddhism         DJ  126
## 10 Protestant         DJ  145
## 11   Catholic         DJ   57
## 12       None         DJ  287
## 13   Buddhism         JP   61
## 14 Protestant         JP   29
## 15   Catholic         JP   11
## 16       None         JP   80
## 17   Buddhism        etc    1
## 18 Protestant        etc    1
## 19   Catholic        etc    5
## 20       None        etc    5
```

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 etc
##   Buddhism   289 154 126  61   1
##   Protestant  84 139 145  29   1
##   Catholic    26  52  57  11   5
##   None       361 292 287  80   5
```

행과 열의 총괄 명칭이 덧붙여졌음을 알 수 있다.

Data Frame with Counts to Data Frame with Cases

2206명 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':    2195 obs. of  2 variables:
##  $ Religion  : Factor w/ 4 levels "Buddhism","Protestant",..: 1 1 1 1 1 1 1
1 1 1 ...
##  $ Candidates: Factor w/ 5 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
## 19   Catholic        etc
## 19.1 Catholic        etc
## 19.2 Catholic        etc
## 19.3 Catholic        etc
## 19.4 Catholic        etc
## 20        None        etc
## 20.1      None        etc
## 20.2      None        etc
## 20.3      None        etc
## 20.4      None        etc
```

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 etc
##   Buddhism   288 153 126  61   1
##   Protestant  83 138 144  29   1
##   Catholic    25  51  57  10   5
##   None       360 292 287  79   5
```

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 etc
##   Buddhism   288 153 126  61   1
##   Protestant  83 138 144  29   1
##   Catholic    25  51  57  10   5
##   None       360 292 287  79   5
```

분할표와 data frame 간의 자료 전환은 기본적으로 위의 과정을 순환한다.

```
save(file="poll87_tbl_df.rda", list=ls())
```

```
savehistory("poll87_tbl_df.Rhistory")
```

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
```

와 같이 하여 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
```

위의 식은 다음과 같이 나타낼 수도 있다.

```
table(UCBAdmissions.cases["Admit"])
```

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

```
table(UCBAdmissions.cases[c("Admit", "Gender")])
```

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

```
ftable(table(UCBAdmissions.cases[c("Gender", "Admit","Dept")]))
```

```
##                 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
```

Why data frame with Cases?

입학허가 여부를 이항 변수로 보고, 성별과 학과를 독립변수로 보는 `glm` 모델을 생각해보자. 이 모델을 적합시키려면 case 별로 기록된 data frame이 필요하다. 입학허가에 성별 차이가 있는지 파악하기 위하여 logit을 link로 하는 binomial family에 적합시켜 보자.

```
UCB.glm.1<-glm(Admit~Gender, family=binomial(link="logit"), data=UCBAdmission
s.cases)
UCB.glm.1
```

```
##
## Call:  glm(formula = Admit ~ Gender, family = binomial(link = "logit"),
##     data = UCBAdmissions.cases)
##
## Coefficients:
##  (Intercept)  GenderFemale
##         0.22          0.61
##
## Degrees of Freedom: 4525 Total (i.e. Null);  4524 Residual
## Null Deviance:       6040
## Residual Deviance: 5950  AIC: 5950
```

```
summary(UCB.glm.1)
```

```
##
## Call:
## glm(formula = Admit ~ Gender, family = binomial(link = "logit"),
##     data = UCBAdmissions.cases)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.544  -1.272   0.851   1.085   1.085
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.2201     0.0388    5.68  1.4e-08 ***
## GenderFemale   0.6104     0.0639    9.55  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6044.3  on 4525  degrees of freedom
## Residual deviance: 5950.9  on 4524  degrees of freedom
## AIC: 5955
##
## Number of Fisher Scoring iterations: 4
```

성별의 계수가 통계적으로 매우 유의하게 나오고 있어서 남녀 간 입학허가의 가능성에 차이가 있음을 알 수 있다. 이를 회귀계수를 중심으로 보다 자세히 살펴보면

```
coef(UCB.glm.1)
```

```
##  (Intercept) GenderFemale
##         0.22         0.61
```

이 값은 바로 $\log \frac{(1-p_{female})/p_{female}}{(1-p_{male})/p_{male}}$ 에 해당한다. 이는 R의 `glm()` 에서 success를 취급하는 방식에 기인하는데 이를 `binomial()` 의 help 파일에서 인용하면 다음과 같다.

Details:

'family' is a generic function with methods for classes '"glm"' and '"lm"' (the latter returning 'gaussian()').

For the 'binomial' and 'quasibinomial' families the response can be specified in one of three ways:

1. As a factor: 'success' is interpreted as the factor not having the first level (and hence usually of having the second level).

2. As a numerical vector with values between '0' and '1', interpreted as the proportion of successful cases (with the total number of cases given by the 'weights').

3. As a two-column integer matrix: the first column gives the number of successes and the second the number of failures.

The 'quasibinomial' and 'quasipoisson' families differ from the 'binomial' and 'poisson' families only in that the dispersion parameter is not fixed at one, so they can model over-dispersion. For the binomial case see McCullagh and Nelder (1989, pp. 124-8). Although they show that there is (under some restrictions) a model with variance proportional to mean as in the quasi-binomial model, note that 'glm' does not compute maximum-likelihood estimates in that model. The behaviour of S is closer to the quasi- variants.

위에서 계산한 바 있는 남성과 여성의 전체 입학허가율을 이 식에 대입해 보면 같은 값을 얻게 된다.

```
p.gender<-prop.table(xtabs(Freq ~ Admit+Gender, data = UCBAdmissions.df), margi
n=2)
p.gender
```

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

```
log(p.gender[2,2]/p.gender[1,2]/(p.gender[2,1]/p.gender[1,1]))
```

```
## [1] 0.61
```

log(odds ratio) 로 의미 파악이 잘 안되면 `exp()` 을 취하여 살펴볼 수도 있다.

```
exp(coef(UCB.glm.1))
```

```
##  (Intercept) GenderFemale
##         1.25         1.84
```

즉, 학과를 고려하지 않았을 때 여성 불합격률의 odds ratio가 남성보다 1.8배 높다는 의미이다. 이는 남성 입학허가율의 odds ratio가 여성보다 1.8배 높다는 의미이기도 하다.

성별 차이에다 학과별 차이를 고려한 모델은

```
UCB.glm.2<-glm(Admit~Gender+Dept, family=binomial(link="logit"), data=UCBAdmiss
ions.cases)
UCB.glm.2
```

```
##
## Call:  glm(formula = Admit ~ Gender + Dept, family = binomial(link = "logi
t"),
##     data = UCBAdmissions.cases)
##
## Coefficients:
##  (Intercept)  GenderFemale         DeptB         DeptC         DeptD
##      -0.5821       -0.0999        0.0434        1.2626        1.2946
##        DeptE         DeptF
##       1.7393        3.3065
##
## Degrees of Freedom: 4525 Total (i.e. Null);  4519 Residual
## Null Deviance:       6040
## Residual Deviance: 5190  AIC: 5200
```

```
summary(UCB.glm.2)
```

```
##
## Call:
## glm(formula = Admit ~ Gender + Dept, family = binomial(link = "logit"),
##     data = UCBAdmissions.cases)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -2.361  -0.959   0.374   0.931   1.477
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.5821     0.0690   -8.44   <2e-16 ***
## GenderFemale  -0.0999     0.0808   -1.24     0.22
## DeptB          0.0434     0.1098    0.40     0.69
## DeptC          1.2626     0.1066   11.84   <2e-16 ***
## DeptD          1.2946     0.1058   12.23   <2e-16 ***
## DeptE          1.7393     0.1261   13.79   <2e-16 ***
## DeptF          3.3065     0.1700   19.45   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6044.3  on 4525  degrees of freedom
## Residual deviance: 5187.5  on 4519  degrees of freedom
## AIC: 5201
##
## Number of Fisher Scoring iterations: 5
```

이제, 성별 차이는 더 이상 유의하지 않고, 학과별 차이(C, D, E, F)가 지배적이다. 성별 분석과 마찬가지로 계수를 살펴보면,

```
coef(UCB.glm.2)
```

```
##  (Intercept) GenderFemale        DeptB        DeptC        DeptD
##      -0.5821      -0.0999       0.0434       1.2626       1.2946
##        DeptE        DeptF
##       1.7393       3.3065
```

이제 성별차이는 거의 없다는 것, 오히려 여성들이 약간 더 유리하다는 것을 다음 odds ratio로 확인할 수 있다. 남성들의 odds ratio는 여성들의 90% 수준이라는 의미이다(이는 다음 시간에 확인).

```
exp(coef(UCB.glm.2)["GenderFemale"])
```

```
## GenderFemale
##        0.905
```

학과별 입학허가 가능성(사실은 R의 binomial family 구성에 따라 불합격 가능성)은 기준인 A학과와의 비교로 이루어진다. B학과의 경우

```
exp(coef(UCB.glm.2)["DeptB"])
```

```
## DeptB
##  1.04
```

로 A학과보다 약간 높다는 것을 알 수 있는 데, 이는 성별을 고려한 입학허가율(혹은 불합격률)에 있어서 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
```

이었던 점을 감안하여 각각의 입학허가율(혹은 불합격률)을 성별 지원자를 가중치로 넣은 가중평균으로 계산하면 확인할 수 있는 일이다. 성별, 학과별 지원자수는

```
applicants.gender.dept<-table(UCBAdmissions.cases[c("Gender","Dept")])
applicants.gender.dept
```

```
##         Dept
## Gender     A   B   C   D   E   F
##   Male   825 560 325 417 191 373
##   Female 108  25 593 375 393 341
```

로부터 확인가능하다. 예를 들어서 A학과의 입학허가율은

```
A.admissions<-(male.admissions[1,1]*applicants.gender.dept[1,1]+female.admissio
ns[1,1]*applicants.gender.dept[2,1])/colSums(applicants.gender.dept)[1]
A.admissions
```

```
##     A
## 0.644
```

B학과의 입학허가율은

```
B.admissions<-(male.admissions[1,2]*applicants.gender.dept[1,2]+female.admissio
ns[1,2]*applicants.gender.dept[2,2])/colSums(applicants.gender.dept)[2]
B.admissions
```

```
##     B
## 0.632
```

로 계산되어 불합격률의 odds ratio, $\frac{1-B.admissions}{B.admissions}$ = 0.581와 비교했을 때 예상했던 대로 A학과 불합격률의 odds ratio, 0.552보다 약간 높다. 그밖에 다른 학과들의 입학허가율을 지원자수를 고려한 가중평균으로 계산하면,

```
(male.admissions[1,3]*applicants.gender.dept[1,3]+female.admissions[1,3]*applic
ants.gender.dept[2,3])/colSums(applicants.gender.dept)[3]
```

```
##     C
## 0.351
```

```
(male.admissions[1,4]*applicants.gender.dept[1,4]+female.admissions[1,4]*applic
ants.gender.dept[2,4])/colSums(applicants.gender.dept)[4]
```

```
##    D
## 0.34
```

```
(male.admissions[1,5]*applicants.gender.dept[1,5]+female.admissions[1,5]*applic
ants.gender.dept[2,5])/colSums(applicants.gender.dept)[5]
```

```
##     E
## 0.252
```

```
(male.admissions[1,6]*applicants.gender.dept[1,6]+female.admissions[1,6]*applic
ants.gender.dept[2,6])/colSums(applicants.gender.dept)[6]
```

```
##      F
## 0.0644
```

로 계산되어 학과별 odds ratio비교가 가능해진다. F학과의 경우 C학과와 비교했을 때,

```
F.C<-exp(coef(UCB.glm.2)["DeptF"])/exp(coef(UCB.glm.2)["DeptC"])
names(F.C)<-"F to C odds ratio"
F.C
```

```
## F to C odds ratio
##              7.72
```

만큼 불합격률의 odds ratio가 차이날 것으로 판단되는 데 이는 $\frac{(1-0.0644)/0.0644}{(1-0.351)/0.351} = 7.857$로 쉽게 확인된다.

anova 로 두 모델 간의 차이를 분석하면

```
anova(UCB.glm.1, UCB.glm.2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Admit ~ Gender
## Model 2: Admit ~ Gender + Dept
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      4524       5951
## 2      4519       5187  5      763   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

즉, 학과별 영향을 고려하면서 성별 영향은 사라진 모델이 적합함을 알 수 있다. 이와 같은 방식으로 제3의 변수가 주는 영향을 파악할 수 있다.

이와 같은 경우에 case별로 재구성한 data frame이 필요하다. 물론 counts로 구성한 data frame 으로도 동일한 분석이 가능하다.

뒷 마무리

```
save(file="tbl_df_poll_UCB_glm.rda", list=ls())
```

```
savehistory("tbl_df_poll_UCB_glm.Rhistory")
```