King Sejong's National Referendum on Tax Reform : Data (Piping)

coop711

2018-05-20

Reading Data

Original data came from intenet version of Sejong silok, summarized by Oh, Ki-Soo.

〈표 1〉 공법관련 여론조사의 찬성과 반대 분석

(단위 : 명, %)

	대신 · 관찰사 · 도사 등				수령				품관 · 촌민				합계			
	찬성		반대		찬성		반대		찬성		반대		찬성		반대	
	인수	%	인수	%	인수	%	인수	%	인수	%	인수	%	인수	%	인수	%
대신 등	21*1	9.8	194*2	90.2									21	9.8	194	90.2
3품 이하 현직	259	39.7	393	60.3									259	39.7	393	60.3
3품 이하 전직	443	79.1	117	20.9									443	79.1	117	20.9
유후사[23]									1,123	94.1	71	5.9	1,123	94.1	71	5.9
경기도					29	85.3	5	14.7	17,076	98.6	236	1.4	17,105	98.6	241	1.4
평안도			1		6	14.6	35	85.4	1,326	4.4	28,474	95.6	1,332	4.5	28,510	95.5
황해도					17	50.0	17	50.0	4,454	22.2	15,601	77.8	4,471	22.3	15,618	77.7
충청도			2		35	57.4	26	42.6	6,982	33.3	14,013	66.7	7,017	33.3	14,041	66.7
강원도					5	33.3	10	66.7	939	12.0	6,888	88.0	944	12.0	6,898	88.0
함길도			1		3	17.6	14	82.4	75	1.0	7,387	99.0	78	1.0	7,402	99.0
경상도					55	77.5	16	22.5	36,262	99.0	377	1.0	36,317	98.9	393	1.1
전라도			2		42	77.8	12	22.2	29,505	99.1	257	0.9	29,547	99.1	271	0.9
	723	50.5	710	49.5	192	58.7	135	41.3	97,742	57.1	73,304	42.9	98,657	57.1	74,149	42.9

출처 : 「세종실록」 12년(1430) 8월 10일

```
sejong_poll <- "../data/sejong_poll.txt" %>%
  read.table(header = TRUE, stringsAsFactors = FALSE)
sejong_poll %>% str
```

```
## 'data.frame':    44 obs. of  4 variables:
##  $ counts: int  21 194 259 393 443 117 1123 71 29 5 ...
##  $ vote  : chr  "yes" "no" "yes" "no" ...
##  $ class : chr  "high" "high" "third.current" "third.current" ...
##  $ region: chr  "Seoul" "Seoul" "Seoul" "Seoul" ...
```

```
sejong_poll %>%
  `[`(., 4:1) %>%
  kable(col.names = c("Region", "Class", "Vote", "Counts"))
```

Region	Class	Vote	Counts
Seoul	high	yes	21
Seoul	high	no	194
Seoul	third.current	yes	259
Seoul	third.current	no	393
Seoul	third.ex	yes	443
Seoul	third.ex	no	117
yuhu	ordinary	yes	1123
yuhu	ordinary	no	71
gyunggi	chief	yes	29
gyunggi	chief	no	5
gyunggi	ordinary	yes	17076
gyunggi	ordinary	no	236
pyungan	high	no	1
pyungan	chief	yes	6
pyungan	chief	no	35
pyungan	ordinary	yes	1326
pyungan	ordinary	no	28474
hwanghae	chief	yes	17
hwanghae	chief	no	17
hwanghae	ordinary	yes	4454
hwanghae	ordinary	no	15601
chungcheong	high	no	2
chungcheong	chief	yes	35
chungcheong	chief	no	26
chungcheong	ordinary	yes	6982
chungcheong	ordinary	no	14013
kangwon	chief	yes	5
kangwon	chief	no	10
kangwon	ordinary	yes	939
kangwon	ordinary	no	6888

Region	Class	Vote	Counts
hamgil	high	no	1
hamgil	chief	yes	3
hamgil	chief	no	14
hamgil	ordinary	yes	75
hamgil	ordinary	no	7387
gyungsang	chief	yes	55
gyungsang	chief	no	16
gyungsang	ordinary	yes	36262
gyungsang	ordinary	no	377
jeolla	high	no	2
jeolla	chief	yes	42
jeolla	chief	no	12
jeolla	ordinary	yes	29505
jeolla	ordinary	no	257

Factor conversion

We need vote, class, region as `factor` s. If you leave them as `chr` , it will be coerced to `factor` when you tabulate it according to alphabetical order, which is not what you want. So, use `factor()` to convert them manually.

```
sejong_poll$vote %<>%
  factor(levels = c("yes","no"), labels = c("Yes", "No"))
```

You can check that `labels =` is not necessary if same as levels. Continue with class and region_

```
class_levels <- c("high","third.current", "third.ex", "chief", "ordinary")
class_labels <- c("High","3rd_current", "3rd_former", "Chief", "Commons")
sejong_poll$class %<>%
  factor(levels = class_levels, labels = class_labels)
```

```
region_levels <- c("Seoul", "yuhu", "gyunggi", "pyungan", "hwanghae", "chungcheong",
"kangwon", "hamgil", "gyungsang", "jeolla")
# region_labels <- c("Seoul", "Yuhu", "Gyunggi", "Pyungan", "Hwanghae", "Chungcheon
g", "Kangwon", "Hamgil", "Gyungsang", "Jeolla")
region_labels <- c("SL","YH", "GG", "PA", "HH", "CC", "KW", "HG", "GS", "JL")
sejong_poll$region %<>%
  factor(levels = region_levels, labels = region_labels)
```

```
sejong_poll %>% str
```

```
## 'data.frame':    44 obs. of  4 variables:
##  $ counts: int  21 194 259 393 443 117 1123 71 29 5 ...
##  $ vote  : Factor w/ 2 levels "Yes","No": 1 2 1 2 1 2 1 2 1 2 ...
##  $ class : Factor w/ 5 levels "High","3rd_current",..: 1 1 2 2 3 3 5 5 4 4 ...
##  $ region: Factor w/ 10 levels "SL","YH","GG",..: 1 1 1 1 1 1 2 2 3 3 ...
```

Array

We can set up the data as an array

```
sejong_poll_array <- sejong_poll %>%
  xtabs(counts ~ vote + class + region, data = .)
sejong_poll_array %>% str
```

```
##  'xtabs' int [1:2, 1:5, 1:10] 21 194 259 393 443 117 0 0 0 0 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ vote  : chr [1:2] "Yes" "No"
##   ..$ class : chr [1:5] "High" "3rd_current" "3rd_former" "Chief" ...
##   ..$ region: chr [1:10] "SL" "YH" "GG" "PA" ...
##  - attr(*, "call")= language xtabs(formula = counts ~ vote + class + region, data
= .)
```

```
sejong_poll_array %>% ftable
```

```
##                region    SL    YH    GG    PA    HH    CC    KW    HG    GS    J
L
## vote class
##
## Yes  High                 21     0     0     0     0     0     0     0     0
0
##      3rd_current         259     0     0     0     0     0     0     0     0
0
##      3rd_former          443     0     0     0     0     0     0     0     0
0
##      Chief                 0     0    29     6    17    35     5     3    55    4
2
##      Commons               0  1123 17076  1326  4454  6982   939    75 36262 2950
5
## No   High                194     0     0     1     0     2     0     1     0
2
##      3rd_current         393     0     0     0     0     0     0     0     0
0
##      3rd_former          117     0     0     0     0     0     0     0     0
0
##      Chief                 0     0     5    35    17    26    10    14    16    1
2
##      Commons               0    71   236 28474 15601 14013  6888  7387   377   25
7
```

Votes

Total and Percentage

Check the total vote with `xtabs()`. Compute total and proportions.

```
vote_total <- sejong_poll %>%
  xtabs(counts ~ vote, data = .)
vote_total %>% str
```

```
##  'xtabs' int [1:2(1d)] 98657 74149
##  - attr(*, "dimnames")=List of 1
##   ..$ vote: chr [1:2] "Yes" "No"
##  - attr(*, "call")= language xtabs(formula = counts ~ vote, data = .)
```

```
vote_total %>%
#> vote_total 을 행렬로 변환
  as.matrix %>%
#> 1-column matrix 를 1-row matrix 로 변환
  t %>%
#> 천 단위에 `,` 사용
  format(big.mark = ",") %>%
#> `kable`에 가운데 정렬
  kable(align = c("c", "c"))
```

Yes	No
98,657	74,149

```
vote_total %>%
#> 백분율을 계산할 수 있도록 `prop.table()` 사용
  prop.table %>%
#> 백분율 계산
  `*`(100) %>%
#> 소수 첫째자리까지 표시
  format(digits = 3, nsmall = 1) %>%
#> 행렬로 변환
  as.matrix %>%
#> 1-column matrix 를 1-row matrix 로 변환
  t %>%
#> 가운데 정렬
  kable(align = c("c", "c"))
```

Yes	No
57.1	42.9

```
#> `array` 에는 `apply()` 적용
vote_total_a <- sejong_poll_array %>%
#> `MARGIN = 1` 이 `vote` 임을 상기하고 집계
  apply(MARGIN = 1, FUN = sum)
vote_total_a %>% str
```

```
##  Named int [1:2] 98657 74149
##  - attr(*, "names")= chr [1:2] "Yes" "No"
```

```
vote_total_a %>%
  as.matrix %>%
  t %>%
  format(big.mark = ",") %>%
  kable(align = c("c", "c"))
```

Yes	No
98,657	74,149

Vote by class

```
vote_class <- sejong_poll %>%
  xtabs(counts ~ vote + class, data = .)
vote_class %>%
  format(big.mark = ",") %>%
#> `format()` 결과는 글자이므로 왼쪽 정렬이 기본값임. 오른쪽으로 정렬하기 위하여 `align = "r"`.
  kable(align = "r", caption = "By Class")
```

By Class

	High	3rd_current	3rd_former	Chief	Commons
Yes	21	259	443	192	97,742
No	200	393	117	135	73,304

```
vote_class %>%
  prop.table(margin = 2) %>%
  `*`(100) %>%
  format(digits = 3, nsmall = 1) %>%
  kable(align = "r", caption = "By Class(%)")
```

By Class(%)

	High	3rd_current	3rd_former	Chief	Commons
Yes	9.5	39.7	79.1	58.7	57.1
No	90.5	60.3	20.9	41.3	42.9

```
#> `array`로부터 계급별 찬반 분할표 구하기.
vote_class_a <- sejong_poll_array %>%
#> `sejong_poll_array`에서 `MARGIN = 1:2`가 `vote`와 `class`임을 상기.
  apply(MARGIN = 1:2, FUN = sum)
vote_class_a %>%
  format(big.mark = ",") %>%
  kable(align = "r", caption = "By Class(%)")
```

By Class(%)

	High	3rd_current	3rd_former	Chief	Commons
Yes	21	259	443	192	97,742
No	200	393	117	135	73,304

Commons vs Bureaucrats

We need to analyse Commons separately. `class_2` factor.

```
#> 고관, 3품이하현직, 3품이하전직을 모두 `관료`로 재분류하고 `factor()` 변환. `ifelse()`를 piping하는 요령에 유의. 논리식 부분을 함수로 처리하되, 어떤 요소가 이어지는지 관찰. 단계별로 `%>%`를 없애고 결과값을 살펴볼 수 있음.
sejong_poll$class_2 <- sejong_poll$class %>%
#> `"Commons"`이면 `TRUE`, 아니면 `FALSE` 인 논리벡터 생성
  `==`("Commons") %>%
#> `TRUE`인 자리에는 `"Commons"`, `FALSE`인 자리에는 `"Bureaus"`.
  ifelse("Commons", "Bureaus") %>%
#> 다음은 `factor` 만으로도 되는데 그 이유를 생각해 볼 것.
  factor(levels = c("Bureaus", "Commons"))
sejong_poll %>%
#> 지역, 계급, 신분, 찬반, 집계 의 순서로 출력
  `[`(., c(4, 3, 5, 2, 1)) %>%
  head(n = 10) %>%
  kable(col.names = c("Region", "Class", "Class_2", "Vote", "Counts"))
```

Region	Class	Class_2	Vote	Counts
SL	High	Bureaus	Yes	21
SL	High	Bureaus	No	194
SL	3rd_current	Bureaus	Yes	259
SL	3rd_current	Bureaus	No	393
SL	3rd_former	Bureaus	Yes	443
SL	3rd_former	Bureaus	No	117
YH	Commons	Commons	Yes	1123
YH	Commons	Commons	No	71
GG	Chief	Bureaus	Yes	29
GG	Chief	Bureaus	No	5

```
sejong_poll %>% str
```

```
## 'data.frame':    44 obs. of  5 variables:
##  $ counts : int  21 194 259 393 443 117 1123 71 29 5 ...
##  $ vote   : Factor w/ 2 levels "Yes","No": 1 2 1 2 1 2 1 2 1 2 ...
##  $ class  : Factor w/ 5 levels "High","3rd_current",..: 1 1 2 2 3 3 5 5 4 4 ...
##  $ region : Factor w/ 10 levels "SL","YH","GG",..: 1 1 1 1 1 1 2 2 3 3 ...
##  $ class_2: Factor w/ 2 levels "Bureaus","Commons": 1 1 1 1 1 1 2 2 1 1 ...
```

Compare the votes by `class_2` , (Bureaucrats vs Commons)

```
#> 찬반(row), 신분(column) 구조의 분할표 생성
vote_class_2 <- sejong_poll %>%
  xtabs(counts ~ vote + class_2, data = .)
#> 집계 분할표 출력
vote_class_2 %>%
  format(big.mark = ",") %>%
  kable(align = "r", caption = "By Bureaus and Commons")
```

By Bureaus and Commons

	Bureaus	Commons
Yes	915	97,742
No	845	73,304

```
#> 신분별 찬반 백분율 출력
vote_class_2 %>%
  prop.table(margin = 2) %>%
  `*`(100) %>%
  format(digits = 3, nsmall = 1) %>%
  kable(caption = "By Bureaus and Commons(%)", align = "r")
```

By Bureaus and Commons(%)

	Bureaus	Commons
Yes	52.0	57.1
No	48.0	42.9

Compute totals and percentages from `array`

```
#> `array` 로부터 `관료`와 `품관촌민`의 찬반 집계를 구하려면, `rowSums()`와 `cbind()`를 활용
vote_class_2_a <- vote_class_a %>%
#> `"Commons"` 제외한 나머지 선택
  `[`(, -5) %>%
#> 찬반 집계
  rowSums %>%
#> 신분별 집계 과정. 관료 신분의 찬반 집계를 흘려 보냄(`.`로 표시). 품관촌민 선택 방식에 유의.
  cbind("Bureaus" = ., "Commons" = vote_class_a %>% `[`(, 5))
#> 신분별 찬반 집계 결과 출력
vote_class_2_a %>%
  format(big.mark = ",") %>%
  kable(align = "r", caption = "By Bureaus and Commons")
```

By Bureaus and Commons

	Bureaus	Commons
Yes	915	97,742
No	845	73,304

Add subtotals to the margins,

```
vote_class_2 %>%
  addmargins %>%
  kable
```

	Bureaus	Commons	Sum
Yes	915	97742	98657
No	845	73304	74149
Sum	1760	171046	172806

Percentages

```
#> 신분별 백분율 집계
vote_class_2_a %>%
  prop.table(margin = 2) %>%
  `*`(100) %>%
#> 소수 첫째자리 표시
  format(digits = 3, nsmall = 1) %>%
#> 수치임을 나타내기 위하여 오른쪽 정렬
  kable(caption = "By Bureaus and Commons(%)", align = "r")
```

By Bureaus and Commons(%)

	Bureaus	Commons
Yes	52.0	57.1
No	48.0	42.9

Votes by region with respect to `class_2`

Count the vote by region class_2 wise.

```
#> 지역별 관료들의 찬반을 집계한 vote_region_bureaus` 생성
vote_region_bureaus <- sejong_poll %>%
#> `subset()` 활용하여 관료만 선택
  subset(.$class_2 == "Bureaus") %>%
#> `.` 은 데이터 프레임에서 관료 부분만 선택한 것임.
  xtabs(counts ~ vote + region,
        data = ., drop = TRUE)
#> 관료들의 지역별 찬반 집계
vote_region_bureaus %>%
  kable(caption = "Bureaus by Region")
```

Bureaus by Region

	SL	GG	PA	HH	CC	KW	HG	GS	JL
Yes	723	29	6	17	35	5	3	55	42
No	704	5	36	17	28	10	15	16	14

```
#> 관료들이 지역별 찬반 백분율 계산
vote_region_bureaus %>%
  prop.table(margin = 2) %>%
  `*`(100) %>%
  format(digits = 3, nsmall = 1) %>%
  kable(caption = "Bureaus by Region(%)", align = "r")
```

Bureaus by Region(%)

	SL	GG	PA	HH	CC	KW	HG	GS	JL
Yes	50.7	85.3	14.3	50.0	55.6	33.3	16.7	77.5	75.0
No	49.3	14.7	85.7	50.0	44.4	66.7	83.3	22.5	25.0

```
#> 품관촌민들의 지역별 찬반을 집계하여 `vote_region_commons` 생성
vote_region_commons <- sejong_poll %>%
  subset(.$class_2 == "Commons") %>%
  xtabs(counts ~ vote + region,
        data = ., drop = TRUE)
#> 집계 결과 출력
vote_region_commons %>%
  format(big.mark = ",") %>%
  kable(caption = "Commons by Region", align = "r")
```

Commons by Region

	YH	GG	PA	HH	CC	KW	HG	GS	JL
Yes	1,123	17,076	1,326	4,454	6,982	939	75	36,262	29,505
No	71	236	28,474	15,601	14,013	6,888	7,387	377	257

```
#> 백분율 출력
vote_region_commons %>%
  prop.table(margin = 2) %>%
  `*`(100) %>%
  format(digits = 1, nsmall = 1) %>%
  kable(caption = "Commons by Region(%)", align = "r")
```

Commons by Region(%)

	YH	GG	PA	HH	CC	KW	HG	GS	JL
Yes	94.1	98.6	4.4	22.2	33.3	12.0	1.0	99.0	99.1
No	5.9	1.4	95.6	77.8	66.7	88.0	99.0	1.0	0.9

Seoul

Seoul has three times more Bureaucrats than other regions, so analyse further.

```
#> 서울의 계급별 집계 결과를 vote_seoul_class`로 생성
vote_seoul_class <- sejong_poll %>%
  subset(.$region == "SL") %>%
  xtabs(counts ~ vote + class,
        data = ., drop = TRUE)
#> 집계 결과 출력
vote_seoul_class %>%
  kable(caption = "Seoul")
```

Seoul

	High	3rd_current	3rd_former
Yes	21	259	443
No	194	393	117

```
#> 백분율 출력
vote_seoul_class %>%
  prop.table(margin = 2) %>%
  `*`(100) %>%
  format(digits = 2, nsmall = 1) %>%
  kable(caption = "Seoul(%)", align = "r")
```

Seoul(%)

	High	3rd_current	3rd_former
Yes	9.8	39.7	79.1
No	90.2	60.3	20.9

Chungcheong's case.

In Chungcheong, the burearocrats and Commons are different in the support of new tax rule.

```
#> 계급별 찬반 집계 분할표를 `vote_chung_class`로 생성
vote_chung_class <- sejong_poll %>%
  subset(.$region == "CC", drop = TRUE) %>%
  xtabs(counts ~ vote + class, data = ., drop = TRUE)
#> 계급별 찬반 집계 출력
vote_chung_class %>%
  format(big.mark = ",") %>%
  kable(caption = "Chungcheong", align = "r")
```

Chungcheong

	High	Chief	Commons
Yes	0	35	6,982
No	2	26	14,013

```
#> 계급별 찬반 백분율 출력
vote_chung_class %>%
  prop.table(margin = 2) %>%
  `*`(100) %>%
  format(digits = 3, nsmall = 1) %>%
  kable(caption = "Chuncheong", align = "r")
```

Chuncheong

	High	Chief	Commons
Yes	0.0	57.4	33.3
No	100.0	42.6	66.7

Save

```
save.image(file = "sejong_poll_data_v2.RData")
```