HW1-2. 국회의원 표결 분석

조성우

2020년 5월 9일

## 문제1. csv불러오기

* #새로 데이터프레임을 만들지 않고 받아온 데이터를 그대로 사용하려고 노력했습니다.이 과정에서 부득이하게 데이터프레임의 첫열을 지우고 열에 이름을 할당하는 작업을 문제2에서 실행하였습니다.

getwd()

## [1] "C:/Users/JSW/Desktop/강의자료/R프로그래밍/R 실습 및 과제/HW/HW안내및소스/HW1/1-2"

setwd("C:\\Users\\JSW\\Desktop\\강의자료\\R프로그래밍\\R 실습 및 과제\\HW\\HW안내및소스\\HW1\\1-2")  
  
virtualData <- read.csv(header=F,"virtualData.csv",stringsAsFactors = F)  
  
str(virtualData)

## 'data.frame': 10 obs. of 10 variables:  
## $ V1 : chr "김우동" "박수박" "이족발" "홍라면" ...  
## $ V2 : chr "일식우리당" "과일천하당" "야식한국당" "일식우리당" ...  
## $ V3 : chr "찬성" "찬성" "찬성" "찬성" ...  
## $ V4 : chr "반대" "찬성" "찬성" "찬성" ...  
## $ V5 : chr "찬성" "찬성" "반대" "찬성" ...  
## $ V6 : chr "반대" "찬성" "찬성" "반대" ...  
## $ V7 : chr "반대" "" "찬성" "반대" ...  
## $ V8 : chr "찬성" "찬성" "반대" "찬성" ...  
## $ V9 : chr "찬성" "찬성" "반대" "찬성" ...  
## $ V10: chr "찬성" "찬성" "반대" "찬성" ...

## 문제2

* dataFrame 의 각 의원 별 표결 정보에서 “찬성”은 1로 “반대”는 -1로 “기권”은 0으로 는 0으로 변경

#새로 데이터프레임을 만들지 않고 받아온 데이터를 그대로 사용하려고 노력했습니다.  
#이 과정에서 문제1과 문제2의 해결 순서가 바뀌게 됐습니다.  
#감안하고 봐주시면 감사하곘습니다.  
  
  
  
#문제 2 : 전처리  
virtualData <- sapply(virtualData,function(x){gsub("찬성","1",x)})   
virtualData <- sapply(virtualData,function(x){gsub("반대","-1",x)})  
virtualData <- sapply(virtualData,function(x){gsub("기권","0",x)})  
  
virtualData0 <- as.numeric(virtualData) #숫자형 데이터셋으로 변경

## Warning: 강제형변환에 의해 생성된 NA 입니다

virtualData0[is.na(virtualData0)] <- 0 #NA값을 0으로 바꿔준다  
  
member <- virtualData[1:10] #기존의 데이터프레임 구성에 맞게 변수에 데이터를 재구성시켜줍니다.  
party <- virtualData[11:20]  
item1 <- virtualData0[21:30]  
item2 <- virtualData0[31:40]  
item3 <- virtualData0[41:50]  
item4 <- virtualData0[51:60]  
item5 <- virtualData0[61:70]  
item6 <- virtualData0[71:80]  
item7 <- virtualData0[81:90]  
item8 <- virtualData0[91:100]  
  
workedVD<- data.frame(member,party,item1,item2,item3,item4,item5,item6,item7,item8,stringsAsFactors = F) # 전처리가 완료된 데이터프레임 생성  
  
  
workedVD <- workedVD[,2:10]   
workedVD

## party item1 item2 item3 item4 item5 item6 item7 item8  
## 김우동 일식우리당 1 -1 1 -1 -1 1 1 1  
## 박수박 과일천하당 1 1 1 1 0 1 1 1  
## 이족발 야식한국당 1 1 -1 1 1 -1 -1 -1  
## 홍라면 일식우리당 1 1 1 -1 -1 1 1 1  
## 박보쌈 야식한국당 -1 1 -1 1 -1 -1 -1 -1  
## 김피자 호화양식당 -1 -1 -1 1 -1 -1 1 -1  
## 장감바스 호화양식당 0 -1 0 0 0 0 0 0  
## 백된장 야식한국당 -1 1 1 1 1 -1 -1 -1  
## 이초밥 일식우리당 0 -1 1 -1 -1 1 0 1  
## 송돈까스 일식우리당 1 0 1 -1 -1 1 1 -1

str(workedVD)

## 'data.frame': 10 obs. of 9 variables:  
## $ party: chr "일식우리당" "과일천하당" "야식한국당" "일식우리당" ...  
## $ item1: num 1 1 1 1 -1 -1 0 -1 0 1  
## $ item2: num -1 1 1 1 1 -1 -1 1 -1 0  
## $ item3: num 1 1 -1 1 -1 -1 0 1 1 1  
## $ item4: num -1 1 1 -1 1 1 0 1 -1 -1  
## $ item5: num -1 0 1 -1 -1 -1 0 1 -1 -1  
## $ item6: num 1 1 -1 1 -1 -1 0 -1 1 1  
## $ item7: num 1 1 -1 1 -1 1 0 -1 0 1  
## $ item8: num 1 1 -1 1 -1 -1 0 -1 1 -1

## 문제1

getwd()

## [1] "C:/Users/JSW/Desktop/강의자료/R프로그래밍/R 실습 및 과제/HW/HW안내및소스/HW1/1-2"

setwd("C:\\Users\\JSW\\Desktop\\강의자료\\R프로그래밍\\R 실습 및 과제\\HW\\HW안내및소스\\HW1\\1-2")  
  
virtualData <- read.csv(header=F,"virtualData.csv",stringsAsFactors = F)  
str(virtualData)

## 'data.frame': 10 obs. of 10 variables:  
## $ V1 : chr "김우동" "박수박" "이족발" "홍라면" ...  
## $ V2 : chr "일식우리당" "과일천하당" "야식한국당" "일식우리당" ...  
## $ V3 : chr "찬성" "찬성" "찬성" "찬성" ...  
## $ V4 : chr "반대" "찬성" "찬성" "찬성" ...  
## $ V5 : chr "찬성" "찬성" "반대" "찬성" ...  
## $ V6 : chr "반대" "찬성" "찬성" "반대" ...  
## $ V7 : chr "반대" "" "찬성" "반대" ...  
## $ V8 : chr "찬성" "찬성" "반대" "찬성" ...  
## $ V9 : chr "찬성" "찬성" "반대" "찬성" ...  
## $ V10: chr "찬성" "찬성" "반대" "찬성" ...

virtualData

## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10  
## 1 김우동 일식우리당 찬성 반대 찬성 반대 반대 찬성 찬성 찬성  
## 2 박수박 과일천하당 찬성 찬성 찬성 찬성 찬성 찬성 찬성  
## 3 이족발 야식한국당 찬성 찬성 반대 찬성 찬성 반대 반대 반대  
## 4 홍라면 일식우리당 찬성 찬성 찬성 반대 반대 찬성 찬성 찬성  
## 5 박보쌈 야식한국당 반대 찬성 반대 찬성 반대 반대 반대 반대  
## 6 김피자 호화양식당 반대 반대 반대 찬성 반대 반대 찬성 반대  
## 7 장감바스 호화양식당 기권 반대 기권 기권  
## 8 백된장 야식한국당 반대 찬성 찬성 찬성 찬성 반대 반대 반대  
## 9 이초밥 일식우리당 기권 반대 찬성 반대 반대 찬성 기권 찬성  
## 10 송돈까스 일식우리당 찬성 기권 찬성 반대 반대 찬성 찬성 반대

## 문제3

#문제 3------------------------------------------------------  
  
#계산의 용이성을 위해 데이터프레임 workedVD에서 행렬계산의 대상이 될 데이터만 subset하겠습니다.  
c <- c(workedVD$item1,workedVD$item2,workedVD$item3,workedVD$item4,workedVD$item5,workedVD$item6,workedVD$item7,workedVD$item8)  
c

## [1] 1 1 1 1 -1 -1 0 -1 0 1 -1 1 1 1 1 -1 -1 1 -1 0 1 1 -1 1 -1  
## [26] -1 0 1 1 1 -1 1 1 -1 1 1 0 1 -1 -1 -1 0 1 -1 -1 -1 0 1 -1 -1  
## [51] 1 1 -1 1 -1 -1 0 -1 1 1 1 1 -1 1 -1 1 0 -1 0 1 1 1 -1 1 -1  
## [76] -1 0 -1 1 -1

MatVD <- matrix(c,10,8,byrow=F) #구상한 계산을 위해 매트릭스를 열기준으로 만들겠습니다.  
  
  
  
#--------반복문으로 행렬곱 구현  
result <- matrix(NA,10,10) #반복문을 통한 내적 계산의 결과를 할당하기위한 matrix를 선언( 그 결과는 100개가 될것으로 10 X 10)  
  
#a로 행렬곱셈의 대상 행을 고정한채로 곱해줄 i에 해당하는 행을 모두 곱해주는 계산을 위해 a와 i로 두단계로 이루어진 반복문을 만들겠습니다.  
  
for (a in 1:10){   
 for (i in 1:10){  
 result[i,a] <- sum(MatVD[a,]\*MatVD[i,])}}  
  
innerVD <- data.frame(result) #결과를 다시 데이터프레임으로 변환합니다.   
  
colnames(innerVD) <-c("김우동","박수박","이족발","홍라면","박보쌈","김피자","장감바스","백된장","이초밥","송돈까스") #결과에 해당하는 이름  
row.names(innerVD) <-c("김우동","박수박","이족발","홍라면","박보쌈","김피자","장감바스","백된장","이초밥","송돈까스") #결과에 해당하는 이름  
  
innerVD

## 김우동 박수박 이족발 홍라면 박보쌈 김피자 장감바스 백된장 이초밥  
## 김우동 8 3 -6 6 -6 -2 1 -6 6  
## 박수박 3 7 -1 5 -3 -3 -1 -1 1  
## 이족발 -6 -1 8 -4 4 0 -1 4 -6  
## 홍라면 6 5 -4 8 -4 -4 -1 -4 4  
## 박보쌈 -6 -3 4 -4 8 4 -1 4 -4  
## 김피자 -2 -3 0 -4 4 8 1 0 -2  
## 장감바스 1 -1 -1 -1 -1 1 1 -1 1  
## 백된장 -6 -1 4 -4 4 0 -1 8 -4  
## 이초밥 6 1 -6 4 -4 -2 1 -4 6  
## 송돈까스 5 2 -3 5 -3 -1 0 -3 3  
## 송돈까스  
## 김우동 5  
## 박수박 2  
## 이족발 -3  
## 홍라면 5  
## 박보쌈 -3  
## 김피자 -1  
## 장감바스 0  
## 백된장 -3  
## 이초밥 3  
## 송돈까스 7

str(innerVD)

## 'data.frame': 10 obs. of 10 variables:  
## $ 김우동 : num 8 3 -6 6 -6 -2 1 -6 6 5  
## $ 박수박 : num 3 7 -1 5 -3 -3 -1 -1 1 2  
## $ 이족발 : num -6 -1 8 -4 4 0 -1 4 -6 -3  
## $ 홍라면 : num 6 5 -4 8 -4 -4 -1 -4 4 5  
## $ 박보쌈 : num -6 -3 4 -4 8 4 -1 4 -4 -3  
## $ 김피자 : num -2 -3 0 -4 4 8 1 0 -2 -1  
## $ 장감바스: num 1 -1 -1 -1 -1 1 1 -1 1 0  
## $ 백된장 : num -6 -1 4 -4 4 0 -1 8 -4 -3  
## $ 이초밥 : num 6 1 -6 4 -4 -2 1 -4 6 3  
## $ 송돈까스: num 5 2 -3 5 -3 -1 0 -3 3 7

## 문제4

* 문제3 을 통해 만들어진 데이터프레임을 분석해보세요.
* 예) 표결 결과가 가장 크게 차이나는 경우는 XX의원과 XX의원을 비교했을 때이다

### 분석 :

* 의원간 표결 결과가 가장 positive한 관계는 김우동 & 홍라면(6), 김우동 & 이초밥(6) 의 두 관계가 있다.
* -> 김우동과 홍라면 이초밥 셋 다 일식우리당 소속으로 이를 보면 주로 같은 당의 의원간 positive 한 관계를 가짐을 알 수 있고 이는 이족발&박보쌈(4) 이족발&백된장(4) 의 결과를 보아 야식한국당에서도 같은 분석결과가 나오는 것을 알 수 있다, 하지만 그중에서도 일식우리당이 당내 구성원 간 가장 높은 표결 일치도를 보여 단합이 잘 되는 모습을 보인다, 다만 김피자와 장 감바스 두 의원으로 구성된 호화양식당은 두 의원간 (1)의 관계를 가져 거대 양당에 비해 비교적 약한 단합력을 가지고 있다고 예상된다)
* 의원간 표결 결과가가장 negative한 관계는 김우동 & 이족발(-6), 김우동 & 박보쌈(-6), 이족발 & 이초밥(-6) 이다 (일식우리당 VS 야식한국당) -> 총의석이 제시된 데이터의 10석이라고 할 때 일식우리당과 야식한국당 거대 양당간 표결이 가장 낮은 표결 일치도를 보여 두 당은 서로 비협조적이고 경쟁적인 관계에 있음을 알 수 있다.
* 서로간 가장 무관한 표결을 보인 관계는 이족발&장감바스(0), 김피자&이초밥(0) 이다. -> 해당 분석의 중심에 있는 당은 장감바스,김피자를 총 구성원으로 가지고 있는 호화양식당으로 ,호화양식당은 이족발(야식한국당)과 이초밥(일식우리당)과 가장 낮은관계를 보임으로써 거대 양당에 치우치지 않는 독자적인 노선을 가지고 있음을 알 수 있다.

## 문제5

* 문제2 에서 데이터프레임을 만든 방법을 이용해 정당 별 표결 평균 및 분산을 구해보세요. 평균과 분산은 8차원 벡터형태여야 합니다.

#정당별 평균함수 Define  
mean.party <- function(b){ #첫번째 인자 a에 결과를 할당할 8차원벡터의 변수명을/두번째 인자에 평균낼 정당명을 입력  
 a <- numeric(8)  
   
 a[1] <- mean(workedVD$item1[party == b])  
 a[2] <- mean(workedVD$item2[party == b])  
 a[3] <- mean(workedVD$item3[party == b])  
 a[4] <- mean(workedVD$item4[party == b])  
 a[5] <- mean(workedVD$item5[party == b])  
 a[6] <- mean(workedVD$item6[party == b])  
 a[7] <- mean(workedVD$item7[party == b])  
 a[8] <- mean(workedVD$item8[party == b])  
   
 return(a)  
}  
  
MeanOfFruit <- mean.party("과일천하당") #정당별 평균함수 Call  
MeanOfJapan <- mean.party("일식우리당")  
MeanOfKorea <- mean.party("야식한국당")  
MeanOfWest <- mean.party("호화양식당")  
  
MeanOfFruit

## [1] 1 1 1 1 0 1 1 1

MeanOfJapan

## [1] 0.75 -0.25 1.00 -1.00 -1.00 1.00 0.75 0.50

MeanOfKorea

## [1] -0.3333333 1.0000000 -0.3333333 1.0000000 0.3333333 -1.0000000 -1.0000000  
## [8] -1.0000000

MeanOfWest

## [1] -0.5 -1.0 -0.5 0.5 -0.5 -0.5 0.5 -0.5

#정당별 분산함수 Define  
var.party <- function(b){ #첫번째 인자 a에 결과를 할당할 8차원벡터의 변수명을/두번째 인자에 평균낼 정당명을 입력  
 a <- numeric(8)  
   
 a[1] <- var(workedVD$item1[party == b])  
 a[2] <- var(workedVD$item2[party == b])  
 a[3] <- var(workedVD$item3[party == b])  
 a[4] <- var(workedVD$item4[party == b])  
 a[5] <- var(workedVD$item5[party == b])  
 a[6] <- var(workedVD$item6[party == b])  
 a[7] <- var(workedVD$item7[party == b])  
 a[8] <- var(workedVD$item8[party == b])  
   
 return(a)  
}  
  
  
VarOfFruit <- var.party("과일천하당") #정당별 분산함수 Call  
VarOfJapan <- var.party("일식우리당")  
VarOfKorea <- var.party("야식한국당")  
VarOfWest <- var.party("호화양식당")  
  
VarOfFruit # 과일천하당은 단 한명의 당원으로 구성된 당으로 분산이 존재할 수 없다.

## [1] NA NA NA NA NA NA NA NA

VarOfJapan

## [1] 0.2500000 0.9166667 0.0000000 0.0000000 0.0000000 0.0000000 0.2500000  
## [8] 1.0000000

VarOfKorea

## [1] 1.333333 0.000000 1.333333 0.000000 1.333333 0.000000 0.000000 0.000000

VarOfWest

## [1] 0.5 0.0 0.5 0.5 0.5 0.5 0.5 0.5

## 문제6

* 의안 정보 시스템 에서 원하는 표결 정보 10 개를 받아 의원 간 표결 비교표를 만들고 분석해보세요. 또, 정당 별 평균 및 분산도 구해보세요. – 의안 정보 시스템 웹페이지 주소 – 페이지에 접속하고 원하는 의안명을 클릭하신 후, 의원성명 탭을 클릭합니다.
* 검색 된 의원 명, 소속 정당, 표결 결과를 모두 긁어 하나의 의안정보 당 csv 파일 한 개를 만듭니다(드래그로 선택 후 엑셀로 가공). 즉, 총 10 개의 csv 파일이 생성되어야 합니다.
* 10 개의 표결정보 데이터(csv 파일)를 하나의 테이블로 만들어서 분석을 진행합니다. 방법은 2 가지가 있습니다. 2 가지중 한가지를 사용하시면 됩니다

1. 다음 설명은 %in% 를 사용해 문제를 해결하는 방법입니다.

* – 추가할 데이터의 각 행의 이름을 조사해 기본 데이터프레임에 있는 이름인지확인 – 기본 데이터 프레임에 있는 이름이라면, 기본 데이터프레임에 새로운 열을 만들고 해당 이름에 해당하는 행에 그 이름에 해당하는 정보(표결 정보)를 추가 – 기본 데이터 프레임에 없는 이름이라면, 기본 데이터프레임에 새로운 열을 만든 뒤, 그 이름에 해당하는 새로운 행을 만들고 이름에 해당하는 정보(당명,표결 정보)를 추가

1. merge 계열 함수를 사용하시면 문제를 해결할 수 있습니다. – merge 계열 함수를 사용하지 않고, %in% 를 사용해 문제를 해결하시면 가산점이 있습니다. – outer join 을 하기 위해서는 merge(x=a, y=b, by = c(“key1”, “key2”), all = T) 함수를 사용하시면 됩니다. x 와 y 는 결합할 데이터 프레임들, by 는 기준 열,all 은 outer join 의 유무 입니다.

#### 문제6.1 10개의 표결데이터 불러오기 및 전처리 후 데이터프레임화

work.data <- function(x){ #csv파일을 전처리 및 데이터프레임화시키는 함수를 define  
x <- unlist(x)  
  
x <- strsplit(x,split="\t") #엑셀의 csv파일에서 직접 불러와지지않아 메모장에 옮긴 후 ANSI로 인코딩하여 불러왔습니다 이 과정에서 열이 구분되지 않은 데이터셋이 되었기 때문에 CSV를 불러왔을 때 \t를 기준으로 데이터가 구분되어있습니다. 이를 다시 알맞게 구분된 원래의 데이터프레임을 만들기 위한 코드입니다.  
x <- unlist(x)  
  
x <- matrix(x,ncol=3,,byrow=T)  
x <- data.frame(x,stringsAsFactors = F)  
  
x <- sapply(x,function(x){gsub("찬성","1",x)},simplify = F) #찬성 반대 기권을 숫자형 데이터로 바꿔주는 함수를 sapply로 실행해줍니다.  
x <- sapply(x,function(x){gsub("반대","-1",x)},simplify = F)  
x <- sapply(x,function(x){gsub("기권","0",x)},simplify = F)  
  
gsub("찬성","1",x)  
  
x <- data.frame(x,stringsAsFactors = F)  
  
  
x$X3 <- as.numeric(x$X3)  
  
  
colnames(x) <- c("name","party","vote")  
return(x)  
}  
  
item1 <- read.csv(header=F,"item1.csv",stringsAsFactors = F) #csv불러오기  
item2 <- read.csv(header=F,"item2.csv",stringsAsFactors = F)  
item3 <- read.csv(header=F,"item3.csv",stringsAsFactors = F)  
item4 <- read.csv(header=F,"item4.csv",stringsAsFactors = F)  
item5 <- read.csv(header=F,"item5.csv",stringsAsFactors = F)  
item6 <- read.csv(header=F,"item6.csv",stringsAsFactors = F)  
item7 <- read.csv(header=F,"item7.csv",stringsAsFactors = F)  
item8 <- read.csv(header=F,"item8.csv",stringsAsFactors = F)  
item9 <- read.csv(header=F,"item9.csv",stringsAsFactors = F)  
item10 <- read.csv(header=F,"item10.csv",stringsAsFactors = F)  
  
item1 <- work.data(item1) # 아까만든 전처리함수를 call  
item2 <- work.data(item2)  
item3 <- work.data(item3)  
item4 <- work.data(item4)  
item5 <- work.data(item5)  
item6 <- work.data(item6)  
item7 <- work.data(item7)  
item8 <- work.data(item8)  
item9 <- work.data(item9)  
item10 <- work.data(item10)

#### 문제6.2 merge 활용 후처리

merged.item <- merge(x=item1,y=item2,by='name',all.x= TRUE) #item1과 item2를 name을 기준변수로하고 merge  
merged.item <- merge(merged.item,item3,by='name',all.x=TRUE) #같은방식으로 쭉 merge  
merged.item <- merge(merged.item,item4,by='name',all.x=TRUE)

## Warning in merge.data.frame(merged.item, item4, by = "name", all.x = TRUE):  
## column names 'party.x', 'vote.x', 'party.y', 'vote.y' are duplicated in the  
## result

merged.item <- merge(merged.item,item5,by='name',all.x=TRUE)

## Warning in merge.data.frame(merged.item, item5, by = "name", all.x = TRUE):  
## column names 'party.x', 'vote.x', 'party.y', 'vote.y' are duplicated in the  
## result

merged.item <- merge(merged.item,item6,by='name',all.x=TRUE)

## Warning in merge.data.frame(merged.item, item6, by = "name", all.x = TRUE):  
## column names 'party.x', 'vote.x', 'party.y', 'vote.y', 'party.x', 'vote.x',  
## 'party.y', 'vote.y' are duplicated in the result

merged.item <- merge(merged.item,item7,by='name',all.x=TRUE)

## Warning in merge.data.frame(merged.item, item7, by = "name", all.x = TRUE):  
## column names 'party.x', 'vote.x', 'party.y', 'vote.y', 'party.x', 'vote.x',  
## 'party.y', 'vote.y' are duplicated in the result

merged.item <- merge(merged.item,item8,by='name',all.x=TRUE)

## Warning in merge.data.frame(merged.item, item8, by = "name", all.x = TRUE):  
## column names 'party.x', 'vote.x', 'party.y', 'vote.y', 'party.x', 'vote.x',  
## 'party.y', 'vote.y', 'party.x', 'vote.x', 'party.y', 'vote.y' are duplicated in  
## the result

merged.item <- merge(merged.item,item9,by='name',all.x=TRUE)

## Warning in merge.data.frame(merged.item, item9, by = "name", all.x = TRUE):  
## column names 'party.x', 'vote.x', 'party.y', 'vote.y', 'party.x', 'vote.x',  
## 'party.y', 'vote.y', 'party.x', 'vote.x', 'party.y', 'vote.y' are duplicated in  
## the result

merged.item <- merge(merged.item,item10,by='name',all.x=TRUE)

## Warning in merge.data.frame(merged.item, item10, by = "name", all.x = TRUE):  
## column names 'party.x', 'vote.x', 'party.y', 'vote.y', 'party.x', 'vote.x',  
## 'party.y', 'vote.y', 'party.x', 'vote.x', 'party.y', 'vote.y', 'party.x',  
## 'vote.x', 'party.y', 'vote.y' are duplicated in the result

merged.item <- merged.item[,c(1,2,3,5,7,9,11,13,15,17,19,21)]  
colnames(merged.item) <- c('name','party','vote1','vote2','vote3','vote4','vote5','vote6','vote7','vote8','vote9','vote10')  
  
merged.item[is.na(merged.item)] <- 0 #na값을 0으로 만들어줍니다  
head(merged.item,20)

## name party vote1 vote2 vote3 vote4 vote5 vote6 vote7 vote8 vote9  
## 1 강길부 무소속 1 1 1 1 1 1 1 1 1  
## 2 강병원 더불어민주당 0 1 -1 1 1 1 1 1 1  
## 3 강석진 미래통합당 1 0 1 1 0 0 0 0 0  
## 4 강효상 미래통합당 -1 0 0 1 0 0 0 0 0  
## 5 강훈식 더불어민주당 1 1 0 1 1 1 1 0 1  
## 6 경대수 미래통합당 -1 -1 1 0 0 0 0 0 0  
## 7 곽상도 미래통합당 0 0 1 -1 0 0 0 1 0  
## 8 권미혁 더불어민주당 1 1 0 1 1 1 0 1 1  
## 9 권은희 국민의당 1 0 -1 0 0 0 1 1 0  
## 10 권칠승 더불어민주당 1 -1 0 0 1 1 1 1 1  
## 11 기동민 더불어민주당 1 1 0 1 1 1 1 1 1  
## 12 김경진 무소속 1 1 0 0 1 1 1 1 1  
## 13 김경협 더불어민주당 1 0 -1 1 1 1 1 0 1  
## 14 김관영 무소속 1 1 -1 1 1 1 1 1 1  
## 15 김광림 미래통합당 -1 1 0 0 0 0 0 0 0  
## 16 김광수 민생당 1 1 -1 1 1 1 1 1 1  
## 17 김규환 미래통합당 -1 1 1 -1 1 1 1 1 1  
## 18 김기선 미래한국당 1 0 1 0 1 1 1 0 1  
## 19 김두관 더불어민주당 1 1 -1 1 1 1 1 1 1  
## 20 김명연 미래통합당 -1 1 0 0 0 0 0 0 0  
## vote10  
## 1 1  
## 2 1  
## 3 0  
## 4 0  
## 5 0  
## 6 0  
## 7 1  
## 8 1  
## 9 1  
## 10 1  
## 11 1  
## 12 1  
## 13 1  
## 14 1  
## 15 0  
## 16 1  
## 17 1  
## 18 0  
## 19 1  
## 20 0

merged.item <- merged.item[,-c(1:2)] #계산을 위해 name, party 잠시 배제  
  
result <- matrix(NA,239,239) #내적곱 계산의 용이성을 위한 행렬 선언 : merged.item의 의원 수가 239명이기 때문에 239x239 행렬을 생성  
  
for (a in 1:239){   
 for (i in 1:239){  
 result[i,a] <- sum(merged.item[a,]\*merged.item[i,])}}  
  
str(result)

## num [1:239, 1:239] 10 7 3 0 7 -1 2 8 3 6 ...

df.result<- data.frame(result,stringsAsFactors = F)  
  
str(df.result)

## 'data.frame': 239 obs. of 239 variables:  
## $ X1 : num 10 7 3 0 7 -1 2 8 3 6 ...  
## $ X2 : num 7 9 0 1 6 -2 0 7 4 5 ...  
## $ X3 : num 3 0 3 0 2 0 0 2 0 1 ...  
## $ X4 : num 0 1 0 2 0 1 -1 0 -1 -1 ...  
## $ X5 : num 7 6 2 0 7 -2 -1 6 2 4 ...  
## $ X6 : num -1 -2 0 1 -2 3 1 -2 -2 0 ...  
## $ X7 : num 2 0 0 -1 -1 1 4 1 1 2 ...  
## $ X8 : num 8 7 2 0 6 -2 1 8 3 5 ...  
## $ X9 : num 3 4 0 -1 2 -2 1 3 5 4 ...  
## $ X10 : num 6 5 1 -1 4 0 2 5 4 8 ...  
## $ X11 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X12 : num 8 7 1 -1 6 -2 2 7 4 6 ...  
## $ X13 : num 6 7 1 0 6 -2 -1 6 4 6 ...  
## $ X14 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X15 : num 0 1 -1 1 0 0 0 0 -1 -2 ...  
## $ X16 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X17 : num 6 5 -1 0 3 1 4 4 1 4 ...  
## $ X18 : num 6 3 2 -1 5 0 1 4 1 5 ...  
## $ X19 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X20 : num 0 1 -1 1 0 0 0 0 -1 -2 ...  
## $ X21 : num -1 0 -1 1 -1 1 0 -1 -1 -1 ...  
## $ X22 : num 7 8 1 0 6 -2 0 7 5 7 ...  
## $ X23 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X24 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X25 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X26 : num 8 7 1 -1 6 -2 2 7 4 6 ...  
## $ X27 : num 2 2 1 1 2 -1 -1 2 0 -1 ...  
## $ X28 : num 7 5 1 0 4 1 3 5 2 6 ...  
## $ X29 : num 7 8 1 0 7 -3 -1 7 4 5 ...  
## $ X30 : num 4 3 -1 0 1 3 4 2 1 6 ...  
## $ X31 : num 7 8 1 0 6 -3 0 7 5 5 ...  
## $ X32 : num 8 5 3 0 5 1 2 6 3 8 ...  
## $ X33 : num 4 5 -1 1 2 0 2 3 2 2 ...  
## $ X34 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X35 : num 1 2 -2 0 1 1 1 1 -1 2 ...  
## $ X36 : num 5 6 1 0 4 -3 0 5 5 3 ...  
## $ X37 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X38 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X39 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X40 : num 5 6 -1 1 3 1 2 4 2 5 ...  
## $ X41 : num 6 7 1 0 5 -1 0 6 5 8 ...  
## $ X42 : num 9 6 3 0 7 -1 1 7 2 5 ...  
## $ X43 : num 7 8 1 0 6 -2 0 7 5 7 ...  
## $ X44 : num 3 0 3 0 2 0 0 2 0 1 ...  
## $ X45 : num 0 -1 1 -1 0 0 0 0 1 2 ...  
## $ X46 : num 8 7 1 -1 6 -2 2 7 4 6 ...  
## $ X47 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X48 : num 6 5 1 -1 4 0 2 5 4 8 ...  
## $ X49 : num 5 3 1 0 4 1 1 3 0 4 ...  
## $ X50 : num 8 5 2 -1 5 0 3 6 3 7 ...  
## $ X51 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X52 : num 6 5 1 2 3 1 2 4 1 2 ...  
## $ X53 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X54 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X55 : num 1 0 0 1 0 1 1 0 -2 -2 ...  
## $ X56 : num 2 1 -1 0 -1 2 4 0 1 2 ...  
## $ X57 : num 7 8 1 0 7 -3 -1 7 4 5 ...  
## $ X58 : num 8 7 1 -1 6 -2 2 7 4 6 ...  
## $ X59 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X60 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X61 : num 8 5 2 -1 5 0 3 6 3 7 ...  
## $ X62 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X63 : num 0 1 0 2 0 1 -1 0 -1 -1 ...  
## $ X64 : num 6 7 1 0 5 -1 0 6 5 8 ...  
## $ X65 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X66 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X67 : num 0 -1 0 1 -1 2 1 -1 -2 -1 ...  
## $ X68 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X69 : num 7 7 0 0 5 -1 2 6 3 5 ...  
## $ X70 : num 6 7 0 -1 5 -2 1 6 5 7 ...  
## $ X71 : num 0 1 -1 1 0 1 0 -1 0 0 ...  
## $ X72 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X73 : num 7 8 0 -1 6 -3 1 7 5 6 ...  
## $ X74 : num 7 8 1 0 6 -2 0 7 5 7 ...  
## $ X75 : num 10 7 3 0 7 -1 2 8 3 6 ...  
## $ X76 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X77 : num 7 8 1 0 6 -3 0 8 4 5 ...  
## $ X78 : num 8 5 3 0 5 -1 2 6 3 4 ...  
## $ X79 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X80 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X81 : num 7 8 1 0 7 -3 -1 7 4 5 ...  
## $ X82 : num 9 8 2 0 7 -2 1 8 4 6 ...  
## $ X83 : num 8 7 1 -1 6 -2 2 7 4 6 ...  
## $ X84 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X85 : num 7 6 2 0 7 -2 -1 6 2 4 ...  
## $ X86 : num 8 9 1 0 7 -3 0 8 5 6 ...  
## $ X87 : num 7 8 1 0 7 -3 -1 7 4 5 ...  
## $ X88 : num 6 5 0 1 3 1 3 4 1 3 ...  
## $ X89 : num 7 8 0 -1 6 -3 1 7 5 6 ...  
## $ X90 : num 7 6 2 0 6 -1 0 6 3 6 ...  
## $ X91 : num 7 6 1 -1 5 -1 2 6 4 7 ...  
## $ X92 : num 6 7 1 0 6 -2 -1 6 4 6 ...  
## $ X93 : num 7 6 1 -1 5 -1 2 6 4 7 ...  
## $ X94 : num 9 7 2 1 6 0 2 7 2 5 ...  
## $ X95 : num 2 1 1 2 1 1 0 1 -2 -2 ...  
## $ X96 : num 8 7 2 0 7 -2 0 7 3 5 ...  
## $ X97 : num 6 7 1 0 5 -1 0 6 5 8 ...  
## $ X98 : num 8 6 2 1 5 1 2 6 2 6 ...  
## $ X99 : num 7 6 1 -1 5 -1 2 6 4 7 ...  
## [list output truncated]

colnames(df.result) <- merged.item$name  
row.names(df.result) <- merged.item$name  
  
head(df.result)

##   
## 1 10 7 3 0 7 -1 2 8 3 6 9 8 6 8 0 8 6 6 8 0 -1 7 8 8 9 8  
## 2 7 9 0 1 6 -2 0 7 4 5 8 7 7 9 1 9 5 3 9 1 0 8 9 9 8 7  
## 3 3 0 3 0 2 0 0 2 0 1 2 1 1 1 -1 1 -1 2 1 -1 -1 1 1 1 2 1  
## 4 0 1 0 2 0 1 -1 0 -1 -1 0 -1 0 0 1 0 0 -1 0 1 1 0 0 0 0 -1  
## 5 7 6 2 0 7 -2 -1 6 2 4 7 6 6 7 0 7 3 5 7 0 -1 6 7 7 7 6  
## 6 -1 -2 0 1 -2 3 1 -2 -2 0 -2 -2 -2 -3 0 -3 1 0 -3 0 1 -2 -3 -3 -2 -2  
##   
## 1 2 7 7 4 7 8 4 9 1 5 9 8 8 5 6 9 7 3 0 8 8 6 5 8 8 6 9  
## 2 2 5 8 3 8 5 5 8 2 6 8 9 9 6 7 6 8 0 -1 7 9 5 3 5 9 5 8  
## 3 1 1 1 -1 1 3 -1 2 -2 1 2 1 1 -1 1 3 1 3 1 1 1 1 1 2 1 1 2  
## 4 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 -1 -1 0 -1 0 -1 0 2 0  
## 5 2 4 7 1 6 5 2 7 1 4 7 7 7 3 5 7 6 2 0 6 7 4 4 5 7 3 7  
## 6 -1 1 -3 3 -3 1 0 -2 1 -3 -2 -3 -3 1 -1 -1 -2 0 0 -2 -3 0 1 0 -3 1 -2  
##   
## 1 8 1 2 7 8 9 8 8 9 0 6 9 8 0 9 7 6 0 8 7 7 10 8 7 8 9  
## 2 9 0 1 8 7 8 9 5 8 1 7 8 9 -1 8 7 7 1 9 8 8 7 9 8 5 8  
## 3 1 0 -1 1 1 2 1 2 2 0 1 2 1 0 2 0 0 -1 1 0 1 3 1 1 3 2  
## 4 0 1 0 0 -1 0 0 -1 0 2 0 0 0 1 0 0 -1 1 0 -1 0 0 0 0 0 0  
## 5 7 0 -1 7 6 7 7 5 7 0 5 7 7 -1 7 5 5 0 7 6 6 7 7 6 5 7  
## 6 -3 1 2 -3 -2 -2 -3 0 -2 1 -1 -2 -3 2 -2 -1 -2 1 -3 -3 -2 -1 -3 -3 -1 -2  
##   
## 1 8 7 9 8 8 7 8 7 6 7 7 7 6 7 9 2 8 6 8 7 7 3 9 5 8 7 4 8  
## 2 9 8 8 7 9 6 9 8 5 8 6 6 7 6 7 1 7 7 6 6 8 4 8 4 9 8 3 9  
## 3 1 1 2 1 1 2 1 1 0 0 2 1 1 1 2 1 2 1 2 1 1 0 2 1 1 1 0 1  
## 4 0 0 0 -1 0 0 0 0 1 -1 0 -1 0 -1 1 2 0 0 1 -1 0 2 0 -1 0 0 1 0  
## 5 7 7 7 6 7 7 7 7 3 6 6 5 6 5 6 1 7 5 5 5 6 1 7 4 7 6 1 7  
## 6 -3 -3 -2 -2 -3 -2 -3 -3 1 -3 -1 -1 -2 -1 0 1 -2 -1 1 -1 -2 1 -2 0 -3 -2 1 -3  
##   
## 1 8 -1 10 6 7 6 7 8 1 8 8 8 7 1 10 3 7 7 6 9 5 8 8 9 9 5 8  
## 2 9 -2 7 7 8 3 4 9 2 7 9 7 8 0 7 0 8 6 5 8 3 9 9 8 7 5 9  
## 3 1 -1 3 1 1 1 1 1 1 1 1 2 1 1 3 3 0 1 0 2 2 1 1 2 2 0 1  
## 4 0 0 0 0 0 -2 -2 0 0 -1 0 0 0 2 0 0 -1 -1 -2 0 1 0 0 0 1 0 0  
## 5 7 -2 7 5 6 3 4 7 2 6 7 6 7 0 7 1 6 5 4 7 2 7 7 7 6 3 7  
## 6 -3 2 -1 -1 -2 1 0 -3 -2 -2 -3 -2 -3 2 -1 1 -3 -1 -1 -2 1 -3 -3 -2 0 0 -3  
##   
## 1 8 9 8 8 1 5 3 2 9 8 8 6 8 0 10 -2 7 8 0 6 5 7 8 2 1 5 7 4 6  
## 2 7 8 9 9 2 6 0 1 8 9 7 7 5 -1 7 -3 6 9 1 3 3 5 7 3 0 6 6 3 5  
## 3 1 2 1 1 0 -1 3 2 2 1 2 0 3 1 3 -1 2 1 0 3 2 0 1 0 1 1 2 1 1  
## 4 -1 0 0 0 2 1 0 0 0 0 0 2 0 2 0 0 0 0 -1 0 1 -1 2 2 -1 0 0 2 -1  
## 5 6 7 7 7 1 3 2 2 7 7 6 5 5 -1 7 -3 6 7 1 3 3 4 5 2 1 4 5 1 5  
## 6 -2 -2 -3 -3 0 1 0 -1 -2 -3 -1 0 1 3 -1 3 -1 -3 -2 -1 0 0 1 2 -1 -3 0 3 -2  
##   
## 1 7 6 3 8 7 8 6 9 8 0 8 4 8 9 9 7 0 7 0 10 2 5 9 9 7 5 10 10  
## 2 5 4 5 9 8 9 3 8 5 1 9 5 7 8 8 8 0 6 -2 7 1 4 7 7 8 6 7 7  
## 3 1 0 0 1 1 1 3 2 2 -2 1 0 2 2 2 0 0 1 0 3 -1 1 2 2 1 1 3 3  
## 4 0 -1 1 0 0 0 0 0 -1 0 0 2 0 0 0 -1 0 -1 -1 0 0 2 1 1 0 0 0 0  
## 5 4 3 2 7 7 7 3 7 5 0 7 2 6 7 7 6 0 5 -1 7 0 2 6 6 6 6 7 7  
## 6 1 1 -2 -3 -3 -3 -1 -2 0 1 -3 1 -1 -2 -2 -3 0 -1 1 -1 2 1 0 0 -2 -2 -1 -1  
##   
## 1 9 9 7 1 5 5 7 6 8 8 6 8 6 2 2 7 9 1 7 6 2 6 4 0 10 8  
## 2 6 8 6 2 6 6 8 7 9 7 5 7 5 0 1 6 8 2 8 7 1 7 5 -1 7 5  
## 3 3 2 2 1 -1 -1 0 1 1 2 2 1 1 1 2 2 2 1 0 0 2 1 1 -1 3 3  
## 4 0 0 0 0 1 1 -1 0 0 0 0 -1 -1 0 0 0 0 0 -1 -1 0 0 0 0 0 0  
## 5 6 7 7 3 3 3 6 7 7 7 7 6 6 0 2 7 7 2 6 6 3 5 5 -2 7 7  
## 6 0 -2 -2 -3 0 0 -3 -3 -3 -2 -2 -2 -2 1 -1 -2 -2 -2 -3 -3 -2 -1 -1 2 -1 -1  
##   
## 1 6 8 7 9 5 5 7 7 9 5 7 9 8 8 3 7 8 6 0 2 9 7  
## 2 7 9 6 8 6 6 8 6 8 6 8 6 9 7 4 6 9 7 -2 1 8 6  
## 3 1 1 2 2 -1 -1 1 2 2 1 1 3 1 1 0 1 1 1 0 1 2 2  
## 4 0 0 0 0 1 1 0 0 0 0 0 0 0 2 -1 -1 0 0 -1 -1 0 0  
## 5 7 7 7 7 3 3 6 7 7 6 6 7 7 5 5 5 7 7 -1 1 7 7  
## 6 -3 -3 -2 -2 1 1 -2 -2 -2 -2 -2 -1 -3 1 -3 -1 -3 -3 1 -1 -2 -2

## 문제7

* 문제 6 에서 선택한 10 개의 표결 정보들을 선택한 이유를 서술하고 분석한 내용들을 토대로 나름대로의 분석결과를 도출하세요. 10개의 표결 데이터는 데이터의 통일성을 위해 총선일인 4.15일 이후의 자료를 위주로 구성했습니다. 극소수의 반대,기권표를 가진 표결을 선택하기 보다,다른 표결들에 비해 상대적으로 반대나 기권표가 두드러진 자료들을 선택하여 유의미한 분석결과를 내고자 했습니다.