# 필요 패키지를 불러옵니다.

library(ggplot2)

library(dplyr)

library(ggmap)

library(gridExtra)

library(arules)

library(rgdal)

library(stringr)

library(sp)

library(rgdal)

library(magrittr)

library(readr)

# setwd("C:/R/RR")

# aniFile <- list.files(path = './data', pattern = 'report')

#

#

# setwd("C:/R/RR/data")

# aniFile="animal1rr.csv"

# aniStat<- read.csv("animal1rr.csv", stringsAsFactors = F, header=F,fill=T,fileEncoding = guess\_encoding(aniFile[1])[1, 1] %>% as.character)

#

# a<-read.csv("report.csv",fileEncoding = guess\_encoding(aniFile[1])[1, 1] %>% as.character)

#####보물같은csv읽기.#############################################################

setwd("C:/R/RR/data")

cat(readLines("animal1.txt"),sep="\n")

aa<-read.table("animal1.txt", header = TRUE, fill = TRUE,fileEncoding=guess\_encoding("animal1.txt")[1, 1] %>% as.character)

setwd("C:/R/RR")

##################################################################################

#필요한 데이터 생성하기.

aa1<-aa[-(1:36),-c(1,2,5:19)]

animal1<-aa1

animal2<- as.numeric(as.character(animal1[,2]))

animal3<-as.data.frame(animal2)

animal4 <- cbind(animal1[1],animal3)

colnames(animal4) <- c("분류","반려동물")

#############################################################################

# '시도명'과 '구시군명'을 합친 '지역명' 컬럼을 생성합니다.

animal4$지역명 <- str\_c("서울시", animal4$분류, sep=' ')

# 시와 구에 공백을 추가합니다. #행정경계구역 데이터와 병합할 때 기준 컬럼으로 사용하기 우히ㅏㅁ.

animal4$지역명 <- animal4$지역명 %>% str\_replace\_all(pattern = '시(?=\\w+구)', replacement = '시 ')#pattern인자에 쓰인 정규표현식은 '시'로 끝나는 부분을 찾으라는 전방탐색

result <- animal4[,-1]

# # 후보별 득표율 컬럼을 생성합니다.

# result[, 22:26] <- result[, 6:10] %>% sapply(FUN = function(x) (x / result$계 \* 100) %>% round(digits = 1L))

# colnames(x = result)[22:26] <- colnames(x = result)[6:10] %>% str\_c('R')

#

# # 지역별로 득표율이 가장 높은 후보를 확인합니다.

# result$최대R <- apply(X = result[, 22:26], MARGIN = 1, FUN = function(x) which(x == max(x)))

#

# # 최대 득표정당색을 지정합니다.

# result$색상 <- ifelse(test = result$최대R == 1, yes = 'blue', no = 'red')

# ggmap 2.7을 설치합니다. (아직 CRAN에 등록되어 있지 않습니다.)

devtools::install\_github('dkahle/ggmap')

# 필요 패키지를 불러옵니다.

library(ggmap)

# 구글 지도 API 인증키를 등록합니다.

# 만약 없다면 건너 뛰어도 무방합니다.

register\_google(key = 'AIzaSyAxgkH49KNuTqrRt0Rh2Qj4AH\_5F8GjZsU')

# 지역명으로 위경도 좌표를 얻습니다.

areaCoords <- geocode(location = enc2utf8(x = str\_c(result$지역명, '청')))

# 위경도 좌표 중 NA인 건수를 확인합니다.

(is.na(x = areaCoords) == TRUE) %>% sum()

# result에 위경도 좌표를 붙입니다.

result <- cbind(result, areaCoords)

# RDS로 저장합니다.

saveRDS(object = result, file = 'animal2.RDS')

##########################################################################

##########################################################################3

# RDS로 저장한 데이터를 불러오겠습니다.

# 데이터가 저장된 폴더로 작업경로를 변경합니다.

setwd("C:/R/RR")

# RDS 파일을 읽습니다.

result <- readRDS(file = 'animal2.RDS')

# 필요 패키지를 불러옵니다.

library(ggplot2)

library(rgdal)

# 단계구분도를 그릴 때 사용할 시/군/구 행정경계구역 데이터를 불러옵니다.

sigg <-

rgdal::readOGR(

dsn = './data/bnd\_sigungu\_00\_2018\_2018',#행정경계구역 데이터가 들어있는폴더를 지정.

layer = 'bnd\_sigungu\_00\_2018\_2018\_2Q',#공통의 파일명을 확장자 없이 지정.(shx, dbf)

encoding = 'UTF-8')

library(ggplot2)

# sigg 객체를 데이터프레임으로 변환합니다.

siggDf <- fortify(model = sigg)

## Regions defined for each Polygons

# sigg@data에서 행번호를 id 컬럼으로 만듭니다.

sigg@data$id <- rownames(x = sigg@data)

# 이제 두 데이터프레임을 병합니다.

siggDf <-

merge(

x = siggDf,

y = sigg@data[, c('id', 'base\_year', 'sigungu\_cd', 'sigungu\_nm')],

by = 'id',

all.x = TRUE)

# id와 order 기준으로 오름차순 정렬합니다.

siggDf <- siggDf[order(siggDf$id, siggDf$order), ]

# 미리보기 합니다.

head(x = siggDf, n = 10L)

# 새로 붙인 컬럼들의 이름을 변경합니다.

colnames(x = siggDf)[8:10] <- c('year', 'siggCd', 'siggNm')

siggDf1<-siggDf#여기까지의 siggDf를 siggDf1에 저장해둔다.

# 광역시도명 코드와 이름을 얻기 위해 시도 행정경계구역 데이터를 불러옵니다.

sido <-

rgdal::readOGR(

dsn = './data/bnd\_sido\_00\_2018\_2018',#행정경계구역 데이터가 들어있는폴더를 지정.

layer = 'bnd\_sido\_00\_2018\_2018\_2Q',#공통의 파일명을 확장자 없이 지정.(shx, dbf)

encoding = 'UTF-8')

# 광역시도명만 추출합니다.

sidoGb <- sido@data[, c('sido\_cd', 'sido\_nm')]

colnames(x = sidoGb) <- c('sidoCd', 'sidoNm')

# siggCd에서 앞 두 글자만 추출하여 광역시도명을 붙입니다.

siggDf$sidoCd <- str\_sub(string = siggDf$siggCd, start = 1, end = 2)

# siggDf에 sidoGb를 병합합니다.

siggDf <-

merge(x = siggDf,

y = sidoGb,

by = 'sidoCd',

all.x = TRUE)

siggDf2<-siggDf

###############################################################

###################################################################

# 컬럼 순서를 변경합니다.

siggDf <- siggDf[, c(2:11, 1, 12)]

# id와 order 기준으로 오름차순 정렬합니다.

siggDf <- siggDf[order(siggDf$id, siggDf$order), ]

####################################################################

# 광역시도명을 두 글자로 줄입니다.

siggDf$sidoNm

a<-

ifelse(test = str\_detect(string = siggDf$sidoNm, pattern = '(남|북)도'),

yes = str\_c(str\_sub(string = siggDf$sidoNm, start = 1, end = 1),

str\_sub(string = siggDf$sidoNm, start = 3, end = 3)),

no = str\_sub(string = siggDf$sidoNm, start = 1, end = 2))

aa<- str\_c(a,"시")

siggDf$sidoNm1 <- aa

##################################################################

# sidoNm과 siggNm을 합친 areaNm 컬럼을 생성합니다.

siggDf$areaNm <- str\_c(siggDf$sidoNm1, siggDf$siggNm, sep = ' ')

# 선거결과 지역명과 경계데이터 areaNm이 서로 같은지 확인합니다.

intersect(x = result$지역명, y = siggDf$areaNm %>% unique() %>% sort()) %>% length()

##이제 그려보자. 근데 데이터 넘길어 자른다.####################################

# order를 10으로 나누었을 때 나머지가 1인 행만 남깁니다.

siggDf <- siggDf[siggDf$order %% 10 == 1, ]

# 부속지역을 제외합니다.

siggDf <- subset(x = siggDf, subset = siggDf$piece == 1)

# 건수를 확인합니다.

nrow(x = siggDf)

# 지도를 깔끔하게 정리하기 위해 나만의 테마를 만듭니다.

my\_theme <-

theme(

panel.background = element\_blank(),

axis.title = element\_blank(),

axis.text = element\_blank(),

axis.ticks = element\_blank(),

plot.title = element\_text(hjust = 0.5, face = 'bold'),

legend.direction = 'horizontal',

legend.position = c(0.8, 0.1),

legend.text = element\_text(family = 'NanumGothic', size = 8, face = 'bold'),

legend.title = element\_text(family = 'NanumGothic', size = 10, face = 'bold')

)

# siggDf 데이터로 지도 이미지를 생성합니다.

ggplot(data = siggDf,

mapping = aes(x = long,

y = lat,

group = group)) +

geom\_polygon(fill = 'white',

color = 'black') +

ggtitle(label = '행경경계구역 데이터로 그린 지도') +

my\_theme +

coord\_fixed()

siggDf3<-siggDf

##############################################################3

# siggDf에 선거 결과 데이터를 붙입니다.

siggDf <-

merge(x = siggDf,

y = result,

by.x = 'areaNm',

by.y = '지역명',

all.x = TRUE)

# id와 order 기준으로 오름차순 정렬합니다.

siggDf <- siggDf[order(siggDf$id, siggDf$order), ]

siggDf5<-siggDf

########################################################################3

# 나만의 팔레트도 만들어서 적용해보겠습니다.

library(RColorBrewer)

myPal <- brewer.pal(n = 9, name = 'Reds')

# 문재인 후보의 득표율로 단계구분도를 그립니다.

ggplot(data = siggDf,

mapping = aes(x = long,

y = lat.x,

group = group)) +

geom\_polygon(mapping = aes(fill = 반려동물),

color = 'black') +

ggtitle(label = '서울시 반려동물 가구') +

my\_theme +

coord\_fixed() +

scale\_fill\_gradient(low = 'white', high = 'red')