Kelling Exercise 3

Claire Kelling

March 30, 2018

Read in Data

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
setwd("C:/Users/ckell/OneDrive/Penn State/2017-2018/01_Spring/SODA_501/SODA_501/Exercise_3")
dat <- read.csv("data/CentreCountyPrecinctResults2016GeneralElection.txt")
```

Calculations

Throughout the following sections, I will label individual parts of my code as either SPLIT, APPLY, or COMBINE. This will be in the left-most part of the code.

Total Vote

First, I will calculate the Total Vote for each precinct.

```
# subsetting the data to just the desired rows with the total votes, and the desired columns
#    of precname, precno, and the votes
# This part is both SPLIT and COMBINE.
#   I am subsetting my data to just total ballots cast, and combining into a df
tot_vote <- dat[which(dat$Contest == "BALLOTS CAST - TOTAL"), c(5,6,13)]
colnames(tot_vote) <- c("PrecNo", "PrecName", "Tot")
```

Two-party Vote

Now, I will calculate the share of the two-party vote for the following: * President * US Senate * Attorney General * Auditor General * State Treasurer

```
## This first part is a SPLIT- using just certain contests and parties
agg_dat <- dat %>% filter(Contest %in% c("PRESIDENTIAL ELECTORS", "UNITED STATES SENATOR",
                                         "ATTORNEY GENERAL", "AUDITOR GENERAL",
                                         "STATE TREASURER")) %>% #use only our interested contests
  filter(Party %in% c("DEMOCRATIC", "REPUBLICAN")) #use only our two-party vote
```

```
## This part is
##             SPLITTING- grouping by precinct name and contest in a dataframe
##             APPLY- applying a sum to each group independently
##
two_vote_tot <- agg_dat %>%
  group_by(PrecName, Contest) %>% #grouping by the precinct and contest
  dplyr::summarise(vote_tot = sum(Count)) #calculating the total (two-party) vote for each

## This part is SPLITTING - using just democratic party,
##                       grouping by precinct name and contest in a dataframe
##              APPLY- applying a sum to each group independently
dem_vote_tot <- agg_dat %>%
  filter(Party %in% c("DEMOCRATIC")) %>% #use only democratic vote
  group_by(PrecName, Contest) %>% #grouping by the precinct and contest
  dplyr::summarise(vote_dem = sum(Count)) #calculating the total vote

## This part is COMBINE- combining into a datastructure to perform calculation
##              APPLY - performing share calculation
share_df <- left_join(dem_vote_tot, two_vote_tot) #combining the two data frames
```

```
## Joining, by = c("PrecName", "Contest")
```

```
share_df$D2 <- round(100*share_df$vote_dem/share_df$vote_tot,2) #calculating the share

#now, I would like to make each row a precinct,
#   instead of each row a race within a precinct
share_df <- share_df[,-c(3,4)] #we just want the share as our numeric

## This part is COMBINE- combining into a datastructure.
share_df <- share_df %>% spread(Contest, D2) #each row is a precinct

#rename columns to desired format
colnames(share_df) <- c("PrecName", "D2Att", "D2Aud",
                        "D2Pre", "D2Tre", "D2Sen")

#combine with total vote data
share_df <- left_join(tot_vote, share_df)
```

```
## Joining, by = "PrecName"
```

Rolloff

Now, I will calculate the rolloff for each of these races: * President * US Senate * Attorney General * Auditor General * State Treasurer

Ballot rolloff is the percentage of voters who voted for President, but did not vote for the particular office. That is: 100 * ((1 - total votes cast for the office)/ (total votes cast for President)).

```
## This part is SPLITTING - using just presidential elections
##              SPLITTING - grouping by precinct name in a dataframe
##              APPLY- applying a sum to each group independently
tot_pres_vote <- dat %>% filter(Contest %in% c("PRESIDENTIAL ELECTORS")) %>% #only president
  group_by(PrecName) %>% #group by precinct and contest
  dplyr::summarise(pres_vote = sum(Count)) #calculating the total vote
```

```
## This part is SPLITTING - using just certain contests
##               SPLITTING - grouping by precinct name and contest
##               APPLY- applying a sum to each group independently
tot_vote <- dat %>% filter(Contest %in% c("UNITED STATES SENATOR",
                                          "ATTORNEY GENERAL", "AUDITOR GENERAL",
                                          "STATE TREASURER")) %>%  #including all other races (not president)
  group_by(PrecName, Contest) %>% #grouping by the precinct and contest
  dplyr::summarise(vote_race = sum(Count)) #calcuating the total vote

## This part is COMBINE- putting all results into a data-frame that makes sense in order to...
##               APPLY- using the rolloff calculation
ro_dat <- left_join(tot_vote, tot_pres_vote) #combining in order to perform calculation
```

```
## Joining, by = "PrecName"
```

```
ro_dat$ro <- round(100*(1-(ro_dat$vote_race/ro_dat$pres_vote)), 2) #doing rolloff calculation

#now, I would like to make each row a precinct,
#   instead of each row a race within a precinct
ro_dat <- ro_dat[,-c(3,4)] #just want to combine on rolloff
ro_dat <- ro_dat %>% spread(Contest, ro) #now each row is a precinct, as desired

colnames(ro_dat) <- c("PrecName", "ROAtt", "ROAud",
                      "ROTre", "ROSen") #rename columns as desired

full_df <- left_join(share_df, ro_dat) #join with the other full dataset
```

```
## Joining, by = "PrecName"
```

Cleaning up final dataset

Now, I would like to clean up the dataset as follows: * Make the PrecName variable without the numbers as the beginning * Order the variables correctly

```
#Make the PrecName variable without the numbers as the beginning
full_df$PrecName <- substring(full_df$PrecName, 4)

#Make proper ordering of columns
full_df <- full_df[,c(1:3, 6, 8, 4, 5, 7, 12, 9, 10, 11)]
```

Write the csv file

```
#writing to a csv
setwd("C:/Users/ckell/OneDrive/Penn State/2017-2018/01_Spring/SODA_501/SODA_501/Exercise_3")
write.csv(full_df, file = "output/Exercise3.csv")
```