Processing

This notebook reads in the Maryland State Bureau of Elections reports from January-September 2018, September 2014 and September 2016, which were converted from PDF to CSV using Tabula (https://tabula.technology/), an open-source tool "for liberating data tables trapped inside PDF files." The files for each month and year are processed and combined into cleaned data files for analysis (see `02_analysis.ipynb`).

The following CSV files are in the `input/` folder for each month and year:

- `*_totals.csv` : total active registration, by county and party
- `*_changes.csv` : voter registration changes, by county and change type (address or name, or changes from* a particular party)
- `*_new.csv` : new registrations, by party and method of registration
- `*_removals.csv` : removals from the registered voter list, by party and reason for removal

The cleaned files are saved in the `output/` folder.

In [1]:
```
## functions to process `totals`, `changes`, `new` and `removals` files

suppressMessages(library('tidyverse'))
suppressMessages(library('lubridate'))
suppressMessages(library('janitor'))
suppressMessages(library('data.table'))

process_totals <- function (month_yr) {
  df <- read.csv(paste0('input/', month_yr, '_totals.csv'), stringsAsFactors =
F)
  df <- df %>% mutate(date = mdy(date),
              DEM = as.numeric(str_replace(DEM,pattern = '\\,',
                                           replacement = '')),
              REP = as.numeric(str_replace(REP,pattern = '\\,',
                                           replacement = '')),
              GRN = as.numeric(str_replace(GRN,pattern = '\\,',
                                           replacement = '')),
              LIB = as.numeric(str_replace(LIB,pattern = '\\,',
                                           replacement = '')),
              UNAF = as.numeric(str_replace(UNAF,pattern = '\\,',
                                            replacement = '')),
              OTH = as.numeric(str_replace(OTH,pattern = '\\,',
                                           replacement = '')),
              TOTAL = as.numeric(str_replace(TOTAL,pattern = '\\,',
                                             replacement = '')),
              CONF.MAILING = as.numeric(str_replace(CONF.MAILING,pattern =
'\\,',
                                                    replacement = '')),
              INACTIVE = as.numeric(str_replace(INACTIVE,pattern = '\\,',
                                                replacement = ''))) %>% adorn_t
otals('row')

  colnames(df) <- tolower(colnames(df))

  df <- df %>%  mutate(date = ifelse(date == '-', lag(date), date),
           dem_perc = dem/total * 100,
           rep_perc = rep/total * 100,
           grn_perc = grn/total * 100,
           lib_perc = lib/total * 100,
           unaf_perc = unaf/total * 100,
           oth_perc = oth/total * 100)
  return(df)
}

process_changes <- function (month_yr) {
  df <- read.csv(paste0('input/', month_yr, '_changes.csv'), stringsAsFactors
= F)
  df <- df %>% mutate(date = mdy(date),
                      ADDRESS = as.numeric(str_replace(ADDRESS,pattern = '\\,'
,
                                                       replacement = '')),
                      NAME = as.numeric(str_replace(NAME,pattern = '\\,',
                                                    replacement = '')),
                      DEM = as.numeric(str_replace(DEM,pattern = '\\,',
                                                   replacement = '')),
                      REP = as.numeric(str_replace(REP,pattern = '\\,',
```

```
                                                replacement = '')),
                      GRN = as.numeric(str_replace(GRN,pattern = '\\,',
                                                replacement = '')),
                      LIB = as.numeric(str_replace(LIB,pattern = '\\,',
                                                replacement = '')),
                      UNAF = as.numeric(str_replace(UNAF,pattern = '\\,',
                                                 replacement = '')),
                      OTH = as.numeric(str_replace(OTH,pattern = '\\,',
                                                replacement = '')),
                      TOTAL = as.numeric(str_replace(TOTAL,pattern = '\\,',
                                                  replacement = ''))) %>% a
dorn_totals('row')
  colnames(df) <- tolower(colnames(df))

  df <- df %>%

    mutate(date = ifelse(date == '-', lag(date), date),
           dem_perc = dem/total * 100,
           rep_perc = rep/total * 100,
           grn_perc = grn/total * 100,
           lib_perc = lib/total * 100,
           unaf_perc = unaf/total * 100,
           oth_perc = oth/total * 100)
  return(df)
}

process_new <- function (month_yr) {
  df <- read.csv(paste0('input/', month_yr, '_new.csv'), stringsAsFactors = F)
  df <- df %>% mutate(date = mdy(date),
           DEM = as.numeric(str_replace(DEM,pattern = '\\,',
                                        replacement = '')),
           REP = as.numeric(str_replace(REP,pattern = '\\,',
                                        replacement = '')),
           GRN = as.numeric(str_replace(GRN,pattern = '\\,',
                                        replacement = '')),
           LIB = as.numeric(str_replace(LIB,pattern = '\\,',
                                        replacement = '')),
           UNAF = as.numeric(str_replace(UNAF,pattern = '\\,',
                                         replacement = '')),
           OTH = as.numeric(str_replace(OTH,pattern = '\\,',
                                        replacement = '')),
           TOTAL = as.numeric(str_replace(TOTAL,pattern = '\\,',
                                          replacement = '')),
           DUPS = as.numeric(str_replace(DUPS,pattern = '\\,',
                                         replacement = ''))) %>% adorn_totals(
'row')

  colnames(df) <- tolower(colnames(df))

  df <- df %>%
    mutate(date = ifelse(date == '-', lag(date), date),
           dem_perc = dem/total * 100,
           rep_perc = rep/total * 100,
           grn_perc = grn/total * 100,
           lib_perc = lib/total * 100,
           unaf_perc = unaf/total * 100,
           oth_perc = oth/total * 100)
```

```
  return(df)
}

process_removals <- function (month_yr) {
  df <- read.csv(paste0('input/', month_yr, '_removals.csv'), stringsAsFactors
= F)
  df <- df %>% mutate(date = mdy(date),
                      DEM = as.numeric(str_replace(DEM,pattern = '\\,',
                                                   replacement = '')),
                      REP = as.numeric(str_replace(REP,pattern = '\\,',
                                                   replacement = '')),
                      GRN = as.numeric(str_replace(GRN,pattern = '\\,',
                                                   replacement = '')),
                      LIB = as.numeric(str_replace(LIB,pattern = '\\,',
                                                   replacement = '')),
                      UNAF = as.numeric(str_replace(UNAF,pattern = '\\,',
                                                    replacement = '')),
                      OTH = as.numeric(str_replace(OTH,pattern = '\\,',
                                                   replacement = '')),
                      TOTAL = as.numeric(str_replace(TOTAL,pattern = '\\,',
                                                     replacement = ''))) %>% a
dorn_totals('row')
  colnames(df) <- tolower(colnames(df))

  df <- df %>%
    mutate(date = ifelse(date == '-', lag(date), date),
           dem_perc = dem/total * 100,
           rep_perc = rep/total * 100,
           grn_perc = grn/total * 100,
           lib_perc = lib/total * 100,
           unaf_perc = unaf/total * 100,
           oth_perc = oth/total * 100)
  return(df)
}
```

In [25]:
```
## apply functions to files

totals.List <- list()
changes.List <- list()
new.List <- list()
removals.List <- list()

for (i in c('01_2018', '02_2018', '03_2018', '04_2018', '05_2018', '06_2018',
'07_2018',
            '08_2018', '09_2018', '09_2016', '09_2014')) {
    totals.List[[i]] <- process_totals(i)
    changes.List[[i]] <- process_changes(i)
    new.List[[i]] <- process_new(i)
    removals.List[[i]] <- process_removals(i)
    }

totals <- rbindlist(totals.List)
changes <- rbindlist(changes.List)
new <- rbindlist(new.List)
removals <- rbindlist(removals.List)

totals$date <- ymd(totals$date)
changes$date <- ymd(changes$date)
new$date <- ymd(new$date)
removals$date <- ymd(removals$date)
```

In [27]:
```
## write to csv in `output/` folder

write_csv(totals, 'output/totals.csv')
write_csv(changes, 'output/changes.csv')
write_csv(new, 'output/new.csv')
write_csv(removals, 'output/removals.csv')
```