# Merge Sum Stats

*Sylvia Klosin*

*1/30/2018*

## Merge Summary Statistics

### Filter

In this case we filter the patent and infutor datasets to include only

- People whose last name starts with "DI"
- Observations more recent or equal to 2000
- People who live in the United States

```
## Warning in strptime(x, format, tz = "GMT"): unknown timezone 'default/
## America/Los_Angeles'
```

```
#=========================
# Section 2: Making new variables
#=========================
patent[, last2 := substr(name_last, start = 1, stop = 2)] # first two letters of last name
patent[, first2 := substr(name_first, start = 1, stop = 2)] # last two letters of last name
patent[, name_first_clean := gsub("([A-Za-z]+).*", "\\1", name_first)] # remove the weird middle inital

infutor[, first2 := substr(name_first, start = 1, stop = 2)] # last two letters of last name
infutor[, name_first_clean := gsub("([A-Za-z]+).*", "\\1", name_first)] # remove the weird middle inita

# since this is the test case # FIXME remove when testing is done
patent <- patent[last2 == "DI",]
```

```
#=========================
# Section 3: Removing duplicate information
#=========================
# Rows that are identical in the data are not needed in the patent data.
# Like if a person has multiple patents, but live in the same place for them,
# we dont need all that data for the merging exersize.

unique(patent[,.(name_first,
                 name_first_clean,
                 first2,
                 name_last,
                 add_city,
                 add_state,
                 unique_inventor_id)]) -> unique_patent
```

### Merge

We merge on last name, state, and the first two letters of the first name. We have 3706 unique inventors that we are trying to find matches for. The infutor data has 2955052 unique person ids.

```r
#=========================
# Section 4: Merging
#=========================
patent_infutor_merge <- merge(unique_patent, infutor,
                              all.x=TRUE, by = c("name_last", "add_state", "first2"),
                              allow.cartesian=TRUE) # doing a LEFT OUTER JOIN
# returns all the rows from the left table, filling in matched columns (or NA) from the right table
# if there are multiple rows from the right table match to a row in the left table, then new rows
# will be added to the left.
```

```r
#=========================
# Section 5: Merge stats
#=========================

patent_infutor_merge_pairs <- unique(patent_infutor_merge[,.(unique_inventor_id, pid)])
sum(is.na(patent_infutor_merge_pairs$pid))
```

```
## [1] 436
```

```r
#=========
# How many matches does one inventor get
#=========
patent_infutor_merge_pairs <- patent_infutor_merge_pairs[!is.na(pid), ]

patent_infutor_merge_pairs_inventor <- patent_infutor_merge_pairs[,.N,.(unique_inventor_id)]

sum(patent_infutor_merge_pairs_inventor$N == 1)
```

```
## [1] 718
```

```r
#=========
# People who get some matches
#=========



patent_1 <- patent_infutor_merge_pairs_inventor[N == 1,]
patent_2 <- patent_infutor_merge_pairs_inventor[N == 2,]

patent_1 <- merge(patent_1, patent_infutor_merge, by = "unique_inventor_id")
patent_2 <- merge(patent_2, patent_infutor_merge, by = "unique_inventor_id")

mean(patent_1$add_city.x == patent_1$add_city.y, na.rm = T)
```

```
## [1] 0.5955631
```

```r
mean(patent_1$name_first_clean.x == patent_1$name_first_clean.y, na.rm = T)
```

```
## [1] 0.8970421
```

```r
patent_2[, same_first := (name_first_clean.x == name_first_clean.y)]

test <- patent_2[same_first == TRUE,]
```