```r
 1   source("lib/utils.R")
 2
 3   # load the election results
 4   results <- read.csv("data/results.csv", colClasses=c(rep("character", 3),
     rep("numeric", 3)))
 5   results$OTHER <- results$TOTAL - (results$CLINTON + results$TRUMP)
 6   results$TRUMP_PERCENT   <- 100 * results$TRUMP    / results$TOTAL
 7   results$CLINTON_PERCENT <- 100 * results$CLINTON  / results$TOTAL
 8   results$TRUMP_MARGIN    <- results$TRUMP_PERCENT - results$CLINTON_PERCENT
 9   results$CLINTON_MARGIN  <- results$CLINTON_PERCENT - results$TRUMP_PERCENT
10   results$MARGIN          <- abs(results$CLINTON_MARGIN)
11
12   state_abbrs <- results$ABBR
13
14   f <- function(abbr) {
15     print(paste("Reading", abbr))
16     read_state(abbr, 2015)
17   }
18
19   names <- do.call(rbind, lapply(state_abbrs, f))
20
21   # If you're on the right OS, you can run this command line to make sure we got every
     name
22   system("grep -o ',2015,' data/states/*.TXT | wc -l")
23
24   names$HANDLE <- paste(names$NAME, "_", names$GENDER, sep="")
25
26   # tally state counts--number of states a name appears in--for each name+gender, and
     join them
27   name_counts <- as.data.frame(table(names$HANDLE))
28   colnames(name_counts) <- c("HANDLE", "STATE_COUNT")
29
30   names <- merge(names, name_counts, by="HANDLE")
31
32   # sort by state rank, with tie going to state with more of that name
33   sorted <- names[order(names$GENDER, names$HANDLE, names$RANK, -names$VALUE),]
34
35   # reduce to those in 10+ states
36   top_10_roster <- unique(names$HANDLE[names$STATE_COUNT >= 10])
37   top_10_names <- subset(sorted, sorted$STATE_COUNT >= 10)
38
39   # for each name, rank which states are highest now that we've sorted them
40   top_10_names$RANK_N <- 0
41   for (handle in top_10_roster) {
42     top_10_names$RANK_N[top_10_names$HANDLE == handle] <-
     seq(1,NROW(top_10_names$RANK[top_10_names$HANDLE == handle]))
43   }
44
45   # and filter down to top ten
46   filtered <- subset(top_10_names, top_10_names$RANK_N <= 10)
47
48   # join names with political results
49   filtered <- merge(filtered, results, by="ABBR")
50   # and re-sort because merges always mess that up
51   filtered <- filtered[order(filtered$GENDER, filtered$HANDLE, filtered$RANK_N),]
52
```

```r
53   # Thx, http://stackoverflow.com/questions/3443687/formatting-decimal-places-in-r
54   specify_decimal <- function(x, k) format(round(x, k), nsmall=k)
55
56   # reduce long decimals for csv files
57   cleaned <- filtered
58   cleaned$TRUMP_PERCENT   <- specify_decimal(filtered$TRUMP_PERCENT, 2)
59   cleaned$CLINTON_PERCENT <- specify_decimal(filtered$CLINTON_PERCENT, 2)
60   cleaned$TRUMP_MARGIN    <- specify_decimal(filtered$TRUMP_MARGIN, 2)
61   cleaned$CLINTON_MARGIN  <- specify_decimal(filtered$CLINTON_MARGIN, 2)
62   cleaned$MARGIN          <- specify_decimal(filtered$MARGIN, 2)
63
64   # eyeball this
65   for (i in 1:100) {
66     for (c in 15:19) {
67       print(paste(paste(filtered[i,c], collapse=" "), paste(cleaned[i,c], collapse=" "),
     sep=" -- "))
68     }
69   }
70
71   # let's put these values in the baby box
72   write.csv(cleaned, "csv/all_names.csv", row.names=FALSE)
73
74   f <- function(handle) {
75     print(handle)
76     write.csv(subset(cleaned, cleaned$HANDLE==handle), paste("csv/names/", handle,
     ".csv", sep=""), row.names=FALSE)
77   }
78
79   # write csvs for every name
80   lapply(top_10_roster, f)
81
82   # total votes for each name among it's top-ten states
83   names_trump <- aggregate(TRUMP ~ HANDLE, FUN=sum, data=filtered)
84   names_clinton <- aggregate(CLINTON ~ HANDLE, FUN=sum, data=filtered)
85   names_other <- aggregate(OTHER ~ HANDLE, FUN=sum, data=filtered)
86   names_total <- aggregate(TOTAL ~ HANDLE, FUN=sum, data=filtered)
87
88   # add to roster
89   roster <- merge(names_trump, names_clinton, by="HANDLE")
90   roster <- merge(roster, names_other, by="HANDLE")
91   roster <- merge(roster, names_total, by="HANDLE")
92
93   roster$TRUMP_PERCENT   <- 100 * roster$TRUMP    / roster$TOTAL
94   roster$CLINTON_PERCENT <- 100 * roster$CLINTON  / roster$TOTAL
95   roster$SPLIT           <- roster$TRUMP_PERCENT - roster$CLINTON_PERCENT
96
97   # join with some of the original info that didn't survive aggregation
98   info <- top_10_names[,c("HANDLE", "NAME", "GENDER", "STATE_COUNT")]
99   info <- info[!duplicated(info), ]
100
101  roster <- merge(roster, info, by="HANDLE")
102
103  # add nat'l data
104  national_names <- read_national(2015)
105  national_names$HANDLE <- paste(national_names$NAME, "_", national_names$GENDER, sep="")
106
107  # tally state counts
```

```r
108   roster <- merge(roster, national_names[,c("HANDLE", "VALUE", "RANK")], by="HANDLE" )
109   roster$WINNER <- ""
110   roster$WINNER[roster$TRUMP < roster$CLINTON] <- "D"
111   roster$WINNER[roster$TRUMP > roster$CLINTON] <- "R"
112
113   roster$D_COUNT <- 0
114   roster$R_COUNT <- 0
115
116   for (i in 1:NROW(roster)) {
117     handle <- roster[i,]$HANDLE
118     print(handle)
119     top_10 <- subset(filtered, filtered$HANDLE == handle)
120     roster[i,]$D_COUNT <- NROW(subset(top_10, top_10$CLINTON > top_10$TRUMP))
121     roster[i,]$R_COUNT <- NROW(subset(top_10, top_10$CLINTON < top_10$TRUMP))
122   }
123
124   # we're done!
125   write.csv(roster[,c(9:16,1:8)], "csv/roster.csv", row.names=FALSE)
```