Since the U.S. midterm elections I’ve been playing around with some Congressional Quarterly data about the composition of the House and Senate since 1945. Unfortunately I’m not allowed to share the data, but here are two or three things I had to do with it that you might find useful.

The data comes as a set of CSV files, one for each congressional session. You download the data by repeatedly querying CQ’s main database by year. In its initial form, the top of each file looks like this:

Results for 79th Congress

,

Last,First,Middle,Suffix,Nickname,Born,Death,Sex,Position,Party,State,District,Start,End,Religion,Race,Educational Attainment,JobType1,JobType2,JobType3,JobType4,JobType5,Mil1,Mil2,Mil3

Abernethy,Thomas,Gerstle,,,05/16/1903,01/23/1953,M,U.S. Representative,Democrat,MS,4,01/03/1945,01/03/1953,Methodist,White,Professional degree,Law,,,,,Did not serve,,

Adams,Sherman,,,,01/08/1899,10/27/1986,M,U.S. Representative,Republican,NH,2,01/03/1945,01/03/1947,Not specified,White,Bachelor's degree,Construction/building trades,,,,,,,

The bottom of each file looks like this:

Young,Milton,Ruben,,,12/06/1897,05/31/1983,M,U.S. Senator,Republican,ND,,03/19/1945,01/03/1981,Mormon,White,Unknown,Agriculture,,,,,Did not serve,,

Zimmerman,Orville,,,,12/31/1880,04/07/1948,M,U.S. Representative,Democrat,MO,10,01/03/1945,04/07/1948,Methodist,White,Professional degree,Education,Law,,,,Army,,

,

,

"Export list of members by biographical characteristics. Washington: CQ Press. Dynamically generated November 10, 2018, from CQ Press Electronic Library, CQ Press Congress Collection: http://library.cqpress.com/congress/export.php?which=memberbioadv&congress=198&yearlimit=0"

To make the files readable in R, the first thing we’ll want to do is strip the first two lines of each file and the last three lines of each file. (Of course I checked first to make sure each file was the same in this regard.) There are several ways to get rid of specific lines from files. The venerable sed command is one. We loop it over each CSV file, telling it to delete (d) lines 1 and 2:

## Remove first two lines from each csv file

for i in \*.csv; do

sed -i.orig '1,2d' $i

done

The -i.orig option makes a copy of the original file first, appending a .orig extension to the filename.

We do the same thing to delete the last three lines of each file. You can use some versions of head to do this quite easily, because they accept a negative number to their -n argument. Thus, while head -n 3 usually returns the first three lines of a file, head -n -3 will show you all but the last three lines. But the version of head that ships with macOS won’t do this. So I used sed again, this time taking advantage of Stack Overflow to find the following grotesque incantation:

## Remove last three lines from each csv file

for i in \*.csv; do

sed -i.orig -e :a -e '1,3!{P;N;D;};N;ba' $i

done

The -e :a is a label for the expression, and the '1,3!{P;N;D;};N;ba' is where the work gets done, streaming through the file till it locates the end, deletes that line, and then branches (b) back to the labeled script again (a) until it’s done it three times. Gross.

You could also do this using a combination of wc (to get a count of the number of lines in the file) and awk, like this:

awk -v n=$(($(wc -l < file) - 3)) 'NR file

There’s a reason people used to say sed and awk had those names because of the sounds people made when forced to use them.

Anyway, now we have a folder full of clean CSV files. Time to fire up R and get to the fun part.

Inside R, we get a vector of our filenames:

filenames <- dir(path = "data/clean",

pattern = "\*.csv",

full.names = TRUE)

filenames

#> [1] "data/clean/01\_79\_congress.csv" "data/clean/02\_80\_congress.csv"

#> [3] "data/clean/03\_81\_congress.csv" "data/clean/04\_82\_congress.csv"

#> [5] "data/clean/05\_83\_congress.csv" "data/clean/06\_84\_congress.csv"

#> [7] "data/clean/07\_85\_congress.csv" "data/clean/08\_86\_congress.csv"

#> [9] "data/clean/09\_87\_congress.csv" "data/clean/10\_88\_congress.csv"

#> [11] "data/clean/11\_89\_congress.csv" "data/clean/12\_90\_congress.csv"

#> [13] "data/clean/13\_91\_congress.csv" "data/clean/14\_92\_congress.csv"

#> [15] "data/clean/15\_93\_congress.csv" "data/clean/16\_94\_congress.csv"

#> [17] "data/clean/17\_95\_congress.csv" "data/clean/18\_96\_congress.csv"

#> [19] "data/clean/19\_97\_congress.csv" "data/clean/20\_98\_congress.csv"

#> [21] "data/clean/21\_99\_congress.csv" "data/clean/22\_100\_congress.csv"

#> [23] "data/clean/23\_101\_congress.csv" "data/clean/24\_102\_congress.csv"

#> [25] "data/clean/25\_103\_congress.csv" "data/clean/26\_104\_congress.csv"

#> [27] "data/clean/27\_105\_congress.csv" "data/clean/28\_106\_congress.csv"

#> [29] "data/clean/29\_107\_congress.csv" "data/clean/30\_108\_congress.csv"

#> [31] "data/clean/31\_109\_congress.csv" "data/clean/32\_110\_congress.csv"

#> [33] "data/clean/33\_111\_congress.csv" "data/clean/34\_112\_congress.csv"

#> [35] "data/clean/35\_113\_congress.csv" "data/clean/36\_114\_congress.csv"

#> [37] "data/clean/37\_115\_congress.csv" "data/clean/38\_116\_congress.csv"

Than, instead of writing a for loop and doing a bunch of rbind-ing, we can pipe our vector of filenames to the map\_dfr() function and we’re off to the races:

data <- filenames %>% map\_dfr(read\_csv, .id = "congress")

colnames(data) <- to\_snake\_case(colnames(data))

data

#> # A tibble: 20,111 x 26

#> congress last first middle suffix nickname born death sex position party

#>

#> 1 1 Aber… Thom… Gerst… NA NA 05/1… 01/2… M U.S. Re… Demo…

#> 2 1 Adams Sher… NA NA NA 01/0… 10/2… M U.S. Re… Repu…

#> 3 1 Aiken Geor… David NA NA 08/2… 11/1… M U.S. Se… Repu…

#> 4 1 Allen Asa Leona… NA NA 01/0… 01/0… M U.S. Re… Demo…

#> 5 1 Allen Leo Elwood NA NA 10/0… 01/1… M U.S. Re… Repu…

#> 6 1 Almo… J. Linds… Jr. NA 06/1… 04/1… M U.S. Re… Demo…

#> 7 1 Ande… Herm… Carl NA NA 01/2… 07/2… M U.S. Re… Repu…

#> 8 1 Ande… Clin… Presba NA NA 10/2… 11/1… M U.S. Re… Demo…

#> 9 1 Ande… John Zuing… NA NA 03/2… 02/0… M U.S. Re… Repu…

#> 10 1 Andr… Augu… Herman NA NA 10/1… 01/1… M U.S. Re… Repu…

#> # ... with 20,101 more rows, and 15 more variables: state ,

#> # district , start , end , religion , race ,

#> # educational\_attainment , job\_type\_1 , job\_type\_2 ,

#> # job\_type\_3 , job\_type\_4 , job\_type\_5 , mil\_1 ,

#> # mil\_2 , mil\_3

A little data-cleaning later and the congress variable is properly numbered and we’re good to go. The to\_snake\_case() function comes from the snakecase package.

The data are observed at the level of congressional terms. So, for example, we can draw a heatmap of the age distribution of U.S. representatives across the dataset:

age\_counts <- data\_all %>%

filter(position == "U.S. Representative",

party %in% c("Democrat", "Republican")) %>%

mutate(binned\_age = Hmisc::cut2(start\_age, g = 30),

binned\_age2 = Hmisc::cut2(start\_age, cuts = c(30:80))) %>%

group\_by(party, start\_year, binned\_age2) %>%

tally() %>%

mutate(freq = n / sum(n),

pct = round((freq\*100), 1))

age\_counts

#> # A tibble: 3,952 x 6

#> # Groups: party, start\_year [76]

#> party start\_year binned\_age2 n freq pct

#>

#> 1 Democrat 1945-01-03 [24,30) 0 0 0

#> 2 Democrat 1945-01-03 30 1 0.00385 0.4

#> 3 Democrat 1945-01-03 31 2 0.00769 0.8

#> 4 Democrat 1945-01-03 32 1 0.00385 0.4

#> 5 Democrat 1945-01-03 33 1 0.00385 0.4

#> 6 Democrat 1945-01-03 34 4 0.0154 1.5

#> 7 Democrat 1945-01-03 35 2 0.00769 0.8

#> 8 Democrat 1945-01-03 36 4 0.0154 1.5

#> 9 Democrat 1945-01-03 37 4 0.0154 1.5

#> 10 Democrat 1945-01-03 38 12 0.0462 4.6

#> # ... with 3,942 more rows

p <- ggplot(age\_counts, aes(x = start\_year, y = binned\_age2, fill = n))

p\_out <- p + geom\_tile() +

scale\_fill\_viridis\_c(option = "A") +

scale\_x\_date(breaks = int\_to\_year(seq(1950, 2010, by = 10), 01, 03),

date\_labels = "%Y",

limits = c(int\_to\_year(1945), int\_to\_year(2019, 2, 1))) +

scale\_y\_discrete(breaks = c(29, 30, 40, 50, 60, 70, 79, 80)) +

facet\_wrap(~ party) +

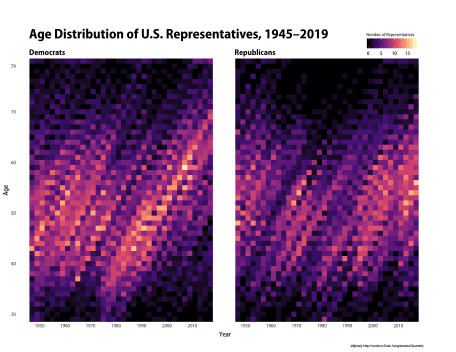
labs(title = "Age Distribution of U.S. Representatives, 1945-2019",

x = "Year", y = "Age", fill = "Number of Representatives",

caption = caption\_text) +

theme(legend.position = "top", legend.box.just = "top")

p\_out



Age distribution heatmap

Or we can look at it a different way, using the ggbeeswarm package. We layer a few different pieces here: a trend line for average age, a ribbon showing the 25th and 75th percentiles of the age distribution, the distribution itself (exlcuding its oldest and youngest 1%), and the names of the representatives in the oldest and youngest percentiles. We’ll create a separate dataset for each of these pieces.

age\_counts <- data\_all %>%

filter(position == "U.S. Representative",

party %in% c("Democrat", "Republican")) %>%

group\_by(party, start\_year, start\_age) %>%

tally() %>%

mutate(freq = n / sum(n),

pct = round((freq\*100), 1)) %>%

arrange(desc(start\_year))

mean\_age\_swarm <- data\_all %>%

filter(position == "U.S. Representative",

party %in% c("Democrat", "Republican")) %>%

group\_by(congress, party) %>%

summarize(year = first(start\_year),

mean\_age = mean(start\_age),

lo = quantile(start\_age, 0.25),

hi = quantile(start\_age, 0.75)) %>%

mutate(yr\_fac = factor(year(year)))

oldest\_group\_by\_year <- data\_all %>% filter(party %in% c("Democrat", "Republican"),

position == "U.S. Representative") %>%

group\_by(congress, party) %>% filter(start\_age > quantile(start\_age, 0.99))

youngest\_group\_by\_year <- data\_all %>% filter(party %in% c("Democrat", "Republican"),

position == "U.S. Representative") %>%

group\_by(congress, party) %>% filter(start\_age < quantile(start\_age, 0.01))

Here’s what they look like:

age\_counts

#> # A tibble: 3,410 x 6

#> # Groups: party, start\_year [76]

#> party start\_year start\_age n freq pct

#>

#> 1 Democrat 2019-01-03 29 1 0.00448 0.4

#> 2 Democrat 2019-01-03 30 1 0.00448 0.4

#> 3 Democrat 2019-01-03 31 1 0.00448 0.4

#> 4 Democrat 2019-01-03 32 2 0.00897 0.9

#> 5 Democrat 2019-01-03 34 2 0.00897 0.9

#> 6 Democrat 2019-01-03 35 2 0.00897 0.9

#> 7 Democrat 2019-01-03 37 3 0.0135 1.3

#> 8 Democrat 2019-01-03 38 5 0.0224 2.2

#> 9 Democrat 2019-01-03 39 4 0.0179 1.8

#> 10 Democrat 2019-01-03 40 5 0.0224 2.2

#> # ... with 3,400 more rows

mean\_age\_swarm

#> # A tibble: 76 x 7

#> # Groups: congress [38]

#> congress party year mean\_age lo hi yr\_fac

#>

#> 1 79 Democrat 1945-01-03 51.5 42 59 1945

#> 2 79 Republican 1945-01-03 52.8 46 59 1945

#> 3 80 Democrat 1947-01-03 50.5 43 58 1947

#> 4 80 Republican 1947-01-03 52.0 45 59 1947

#> 5 81 Democrat 1949-01-03 49.4 42 56 1949

#> 6 81 Republican 1949-01-03 53.7 47 61 1949

#> 7 82 Democrat 1951-01-03 50.8 43 57 1951

#> 8 82 Republican 1951-01-03 53.8 46.5 61 1951

#> 9 83 Democrat 1953-01-03 50.7 43 57 1953

#> 10 83 Republican 1953-01-03 52.9 46 60 1953

#> # ... with 66 more rows

oldest\_group\_by\_year

#> # A tibble: 181 x 38

#> # Groups: congress, party [76]

#> congress last first middle suffix nickname born death sex

#>

#> 1 79 Doug… Robe… Lee NA NA 1863-11-07 1954-10-01 M

#> 2 79 Mans… Jose… J. NA NA 1861-02-09 1947-07-12 M

#> 3 79 Eaton Char… Aubrey NA NA 1868-03-29 1953-01-23 M

#> 4 79 Welch Rich… Joseph NA NA 1869-02-13 1949-09-10 M

#> 5 80 Doug… Robe… Lee NA NA 1863-11-07 1954-10-01 M

#> 6 80 Mans… Jose… J. NA NA 1861-02-09 1947-07-12 M

#> 7 80 Saba… Adol… Joach… NA NA 1866-04-04 1952-11-06 M

#> 8 80 Eaton Char… Aubrey NA NA 1868-03-29 1953-01-23 M

#> 9 80 Lewis Will… NA NA NA 1868-09-22 1959-08-08 M

#> 10 81 Bloom Sol NA NA NA 1870-03-09 1949-03-07 M

#> # ... with 171 more rows, and 29 more variables: position , party ,

#> # state , district , start , end , religion ,

#> # race , educational\_attainment , job\_type\_1 ,

#> # job\_type\_2 , job\_type\_3 , job\_type\_4 , job\_type\_5 ,

#> # mil\_1 , mil\_2 , mil\_3 , start\_year , end\_year ,

#> # name\_dob , pid , start\_age , poc , days\_old ,

#> # months\_old , full\_name , end\_career , entry\_age ,

#> # yr\_fac

youngest\_group\_by\_year

#> # A tibble: 163 x 38

#> # Groups: congress, party [76]

#> congress last first middle suffix nickname born death sex

#>

#> 1 79 Beck… Lind… Gary NA NA 1913-06-30 1984-03-09 M

#> 2 79 Foga… John Edward NA NA 1913-03-23 1967-01-10 M

#> 3 79 Ryter John Franc… NA NA 1914-02-04 1978-02-05 M

#> 4 79 Benn… Mari… Tinsl… NA NA 1914-06-06 2000-09-06 M

#> 5 79 Byrn… John Willi… NA NA 1913-06-12 1985-01-12 M

#> 6 80 Bent… Lloyd Milla… Jr. NA 1921-02-11 2006-05-23 M

#> 7 80 Kenn… John Fitzg… NA NA 1917-05-29 1963-11-22 M

#> 8 80 Will… John Bell NA NA 1918-12-04 1983-03-25 M

#> 9 80 Nodar Robe… Joseph Jr. NA 1916-03-23 1974-09-11 M

#> 10 80 Pott… Char… Edward NA NA 1916-10-30 1979-11-23 M

#> # ... with 153 more rows, and 29 more variables: position , party ,

#> # state , district , start , end , religion ,

#> # race , educational\_attainment , job\_type\_1 ,

#> # job\_type\_2 , job\_type\_3 , job\_type\_4 , job\_type\_5 ,

#> # mil\_1 , mil\_2 , mil\_3 , start\_year , end\_year ,

#> # name\_dob , pid , start\_age , poc , days\_old ,

#> # months\_old , full\_name , end\_career , entry\_age ,

#> # yr\_fac

Now we can draw a graph, faceted by Party:

## Don't show points for the people we're naming

exclude\_pid <- c(oldest\_group\_by\_year$pid, youngest\_group\_by\_year$pid)

party\_names <- c(`Democrat` = "Democrats",

`Republican` = "Republicans")

p <- ggplot(data = subset(data\_all,

party %in% c("Democrat", "Republican") &

position == "U.S. Representative" &

pid %nin% exclude\_pid),

mapping = aes(x = yr\_fac, y = start\_age,

color = party,

label = last))

p\_out <- p + geom\_quasirandom(size = 0.1, alpha = 0.4,

method = "pseudorandom", dodge.width = 1) +

# mean age trend

geom\_line(data = mean\_age\_swarm,

mapping = aes(x = yr\_fac, y = mean\_age,

color = party, group = party),

inherit.aes = FALSE, size = 1, alpha = 0.5) +

# 25/75 percentile ribbon

geom\_ribbon(data = mean\_age\_swarm,

mapping = aes(x = yr\_fac, ymin = lo, ymax = hi,

color = NULL, fill = party, group = party),

inherit.aes = FALSE, alpha = 0.2) +

# Named outliers

geom\_text(data = oldest\_group\_by\_year,

size = 0.9, alpha = 1, position = position\_jitter(width = 0.4, height = 0.4)) +

geom\_text(data = youngest\_group\_by\_year,

size = 0.9, alpha = 1, position = position\_jitter(width = 0.4, height = 0.4)) +

# Hackish compromise to label years:

# we can't use a date object with the beeswarm plot, only a factor

scale\_x\_discrete(breaks = levels(data\_all$yr\_fac)[c(T, rep(F, 4))]) +

scale\_color\_manual(values = party\_colors) +

scale\_fill\_manual(values = party\_colors) +

guides(color = FALSE, fill = FALSE) +

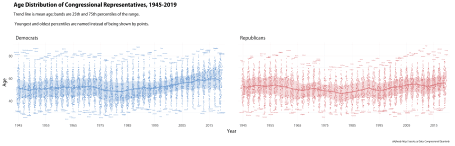
labs(x = "Year", y = "Age", title = "Age Distribution of Congressional Representatives, 1945-2019",

subtitle = "Trend line is mean age; bands are 25th and 75th percentiles of the range.\n\n Youngest and oldest percentiles are named instead of being shown by points.",

caption = caption\_text) +

facet\_wrap( ~ party, nrow = 1, labeller = as\_labeller(party\_names)) +

theme(plot.subtitle = element\_text(size = 10))



Age trends, distributions, and outliers.

That one might be easier to see [as a PDF](http://kieranhealy.org/files/misc/age_careers_line_labeled.pdf).

Finally, here’s a neat trick. One thing I was interested in was changes in the composition of the so-called “Freshman Class” of representatives over time—that is, people elected to the House for the very first time. To extract that subset, I needed to create a term\_id nested with each person’s unique identifier (their pid). I knew what Congressional session each person-term was in, but just needed to count from the first to the last. I’m sure there’s more than one way to do it, but here’s a solution:

first\_terms <- data\_all %>%

filter(position == "U.S. Representative", start > "1945-01-01") %>%

group\_by(pid) %>% nest() %>%

mutate(data = map(data, ~ mutate(.x, term\_id = 1 + congress - first(congress)))) %>%

unnest() %>% filter(term\_id == 1)

first\_terms

#> > # A tibble: 2,998 x 39

#> pid congress last first middle suffix nickname born death sex

#>

#> 1 1 79 Aber… Thom… Gerst… NA NA 1903-05-16 1953-01-23 M

#> 2 2 79 Adams Sher… NA NA NA 1899-01-08 1986-10-27 M

#> 3 4 79 Allen Asa Leona… NA NA 1891-01-05 1969-01-05 M

#> 4 5 79 Allen Leo Elwood NA NA 1898-10-05 1973-01-19 M

#> 5 6 79 Almo… J. Linds… Jr. NA 1898-06-15 1986-04-14 M

#> 6 7 79 Ande… Herm… Carl NA NA 1897-01-27 1978-07-26 M

#> 7 9 79 Ande… John Zuing… NA NA 1904-03-22 1981-02-09 M

#> 8 10 79 Andr… Augu… Herman NA NA 1890-10-11 1958-01-14 M

#> 9 13 79 Andr… Walt… Gresh… NA NA 1889-07-16 1949-03-05 M

#> 10 14 79 Ange… Homer Daniel NA NA 1875-01-12 1968-03-31 M

#> # ... with 2,988 more rows, and 29 more variables: position , party ,

#> # state , district , start , end , religion ,

#> # race , educational\_attainment , job\_type\_1 ,

#> # job\_type\_2 , job\_type\_3 , job\_type\_4 , job\_type\_5 ,

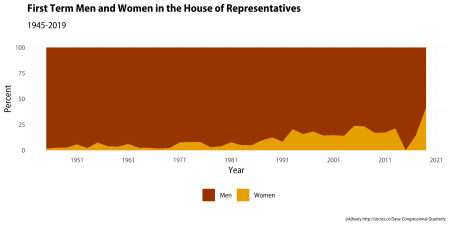
#> # mil\_1 , mil\_2 , mil\_3 , start\_year , end\_year ,

#> # name\_dob , start\_age , poc , days\_old ,

#> # months\_old , full\_name , end\_career , entry\_age ,

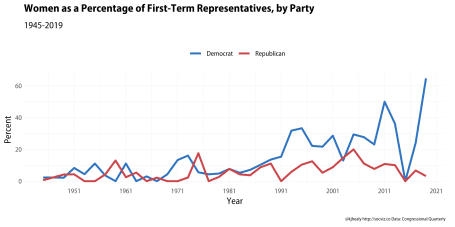
#> # yr\_fac , term\_id

The trick here is that mutate(data = map(data, ~ mutate(.x, term\_id = 1 + congress - first(congress)))) line, which nests one mutate call inside another. We group the data by pid and nest() it so it’s as if we have a separate table for each representative. Then we use map() to add a term\_id column to each subtable. Once we have a per-person term\_id, and we grab everyone’s first term, we can e.g. take a look at the breakdown of freshman representatives by gender for every session since 1945:



First-term representatives by gender, 1945-2019

And also to break that out by Party:



First-term representatives by gender and party, 1945-2019