SafeGraph Data Parse

Chris Hoover

7/31/2020

Goal:

Goal is to have an $N \times N \times t$ array where N is the number of counties in CA with population $> 250{,}000$ and t is the number of days of interest. Each $N \times N$ slice is a matrix where each N_{ij} entry is the proportion of devices tracked by SafeGraph that spent time in j from their home in i.

SafeGraph Data:

SafeGraph maintains a Social distancing metrics database with 1 .csv file per day where each row is a census block group in the U.S. and relevant columns include the number of devices tracked and a list of census block groups that were visited by x devices on the day of observation.

```
# Read_csv works best for the json column. fread inserts \ trying to escape quotation marks or somethin
# sfgrph <- data.table(readr::read_csv("2020-06-01-social-distancing.csv"))
#  sfgrph[, state_fips:=substr(origin_census_block_group, start = 1, stop = 2)]
#  sfgrph[, cnty_fips:=substr(origin_census_block_group, start = 3, stop = 5)]
#  sfgrph[, ct_fips:=substr(origin_census_block_group, start = 1, stop = 11)]

#  sfgrph_ca <- sfgrph[state_fips == "06"]

# saveRDS(sfgrph_ca, "safegraph_2020-06-01_CA.rds")

sfgrph_ca <- readRDS("safegraph_2020-06-01_CA.rds")

head(sfgrph_ca %>%
       dplyr::select(origin_census_block_group, date_range_start, device_count, destination_cbgs))
```

```
##    origin_census_block_group    date_range_start device_count
## 1:              060372713005 2020-06-01 07:00:00           29
## 2:              060372766041 2020-06-01 07:00:00           80
## 3:              060374334012 2020-06-01 07:00:00           35
## 4:              060375017003 2020-06-01 07:00:00           39
## 5:              060375322002 2020-06-01 07:00:00           68
## 6:              060375401013 2020-06-01 07:00:00           50
##
## 1:
## 2:
## 3:
## 4:
```

```
## 5: {"060375006005":1,"060375009001":1,"060371911202":1,"060375401022":1,"060372774002":1,"0603724270
## 6:
```

Process

Main challenge is parsing the `destinatin_cbgs` column as each entry contains a json file of the census block groups visited and the number of devices visiting them.

```
sfgrph_ca$destination_cbgs[1]
```

```
## [1] "{\"060372713005\":25,\"060372652022\":1,\"060372719011\":2,\"060372713001\":1,\"060372713004\":
```

We ca use `jsonlite::fromJSON` to convert each entry into a list or data.frame

```
jsonlite::fromJSON(sfgrph_ca$destination_cbgs[1])
```

```
## $`060372713005`
## [1] 25
##
## $`060372652022`
## [1] 1
##
## $`060372719011`
## [1] 2
##
## $`060372713001`
## [1] 1
##
## $`060372713004`
## [1] 1
##
## $`060372640002`
## [1] 1
##
## $`060372721004`
## [1] 1
##
## $`060372125021`
## [1] 1
##
## $`060379201021`
## [1] 1
```

And the census block group names have data on the state and county embedded. Specifically, first two digits is the state fips code (06 is for California) and the next three digits are for the county (e.g. 06075 is San Francisco, see here for other CA county fips codes).

Given that, we could get to the desired $N \times N \times t$ matrix by:

1) expanding the json columns `destination_cbgs` into a readable format and extracting the county info. `jsonlite::fromJSON` extracts the data into a list and `susbtr([cbg],start = 3, stop = 5)` will return the county fips code.

2) determine the proportion of devices from `origin_census_block_group` that visit each destination_cbg

3) fill entries in the $N \times N \times t$ matrix with the appropriate data

Not sure what the most efficient way to do this is, could probably speed it up by aggregating everything to the county first, but not sure how to merge the json data in a smart way. E.g. if we take all the cbgs in 06075 and meld all their `destination_cbgs` entries into one json entry, then parse that json entry. Could also maybe do some preprocessing on the json entries to only consider out-of-county trips.

Another caveat is that data from each day (e.g. each t slice) is contained in a separate csv file, so I think the function would ideally read a csv and return an $N \times N \times$ matrix