The American Community Survey (ACS), which is conducted on a regular basis, and takes place every 10 years, are the major source of demographic data for the United States Census Bureau. This information is often combined into several geographic regions, such as census block groups, tracts, and counties, to provide an extensive view of the features and requirements of various regions across the nation.

Although ACS data is available for census geographies such as tracts and block groups, it is not available for blocks. However, there are communities and neighborhoods that require specific information about their geographic boundaries, which may not always align with the boundaries drawn by the Bureau of Census. To address this issue, we are using a weighting scheme to convert ACS data from one geography, such as (tract, or block group) and converting them into the different relevant boundaries for the community or neighborhood. This is done by allocating the data based on how much the tract or block group falls inside or outside the specific community using a number of housing units or people in various blocks. This approach ensures that the data is appropriately allocated, providing a more accurate representation of the community’s population.

R programming provides an advanced set of tools for geospatial data analysis and visualization, including the distribution of ACS data from census geographies to neighborhood and community districts. Transforming the ACS data in order to reallocate data at the neighbourhood and community levels by utilizing R packages like "sf" and "dplyr." Once the data has been reallocated, geographical visualizations displaying the distribution of the data among neighbourhoods may be made using the "ggplot2" and "sf" functions.

Data\_Dictionary: Neighborhoods

Authors: Satya Golla and Steve Simon

copyright:

Uncertain.

creation date: Unknown

Description:

Shapefiles for neighborhoods in the

Kansas City metropolitan area.

format:

proprietary: Shapefile (zip)

missing-value-code: not needed

Rows: 375

Columns: 10

Source:

See doc/email-wilson-2022-03-03.pdf

Vars:

NID:

Label: Numeric id

Area Name:

Label: Name of neighborhood

nbr\_id:

Label: Sequential id

CommunityD:

Label: Which community district this belongs to

id:

Label: Unknown

Unique id field, inde variable gi

label\_long:

label: Unknown

long name of nbd

shid:

label: Unknown

id for Shapefile

Shape\_Leng:

label: Perimeter of neighborhood

unit: Unknown

Shape\_Area:

label: Area of neighborhood

unit: Unknown

---

CODE

+ bg.RData

+ bl.RData

+ nbd.RData

+ nbd-intersections.RData

+ red.RData

### General information

For information about various files, refer to the files that stored the relevant data (e.g., store\_block\_information.Rmd).

It helps to know the FIPS (Federal Information Processing System) codes **for** Kansas (20) and Missouri (29).

The important counties **for** this program are

Cass County, MO (29037)

Clay County, MO (29047)

Jackson County, MO (29095)

Johnson County, KS (20091)

Leavenworth County, KS (20103)

Wyandotte County, KS (20209)

### Setup

```{r setup}

library(sf)

library(tidyverse)

path\_name <- "../Data/"

```

### Load key files

Block groups

```{r load-bg}

load(paste0(path\_name, "bg", ".RData"))

glimpse(bg)

```

Blocks

```{r load-bl}

load(paste0(path\_name, "bl", ".RData"))

glimpse(bl)

```

neighborhoods

```{r load-nbd}

load(paste0(path\_name, "nbd", ".RData"))

glimpse(nbd)

```

neighborhoods intersections

```{r load-nbd-intersections}

load(paste0(path\_name, "nbd-intersections", ".RData"))

glimpse(bg\_nbd\_intersection)

glimpse(bl\_nbd\_intersection)

```

```{r load-redistricting}

load(paste0(path\_name, "red", ".RData"))

glimpse(red)

```

### Counts

```{r display-nice-counts}

# This code will compute the count of a vector

# and display it with commas if it is greater

# than 999.

nice\_count <- function(x) {

x %>%

unique %>%

length %>%

format(big.mark=",")

}

```

There are 7 counties **in** the Kansas City metropolitan area.

There are `r nice\_count(nbd$NID)` neighborhoods **in** the 7 counties.

There are

`r nice\_count(bg$bg\_id)`

block groups **in** the 7 counties.

`r nice\_count(bg\_nbd\_intersection$bg\_id)`

of these intersect partially or completely with one or more neighborhoods.

There are

`r nice\_count(bl$bl\_id)`

blocks **in** the 7 counties.

`r nice\_count(bl\_nbd\_intersection$bl\_id)`

of these intersect partially or completely with one or more neighborhoods.

### Get block group list

Identify the list of all block groups that partially or completely intersect with any neighborhoods.

```{r intersect-bg}

bg\_nbd\_intersection %>%

pull(bg\_id) %>%

unique -> bg\_list

glimpse(bg\_list)

```

### Get block list

Now get a list of all census blocks that lie inside any of these block groups. Merge with red to get population and housing unit counts.

```{r pull-blocks}

bl %>%

pull(bl\_id) %>%

str\_sub(1, 12) %>%

bind\_cols(bl$bl\_id) %>%

set\_names(c("bg\_id", "bl\_id")) %>%

filter(bg\_id %in% bg\_list) %>%

inner\_join(red,

by=c("bg\_id", "bl\_id")) -> bl\_list

glimpse(bl\_list)

```

### Calculate counts and areas of intersections

```{r calculate-proportions}

bl\_nbd\_intersection %>%

full\_join(bl\_list,

by=c("bg\_id", "bl\_id")) %>%

replace\_na(list(bl\_prop\_in=0)) %>%

replace\_na(list(NID=0)) %>%

mutate(

people\_in=

round(people\*bl\_prop\_in)) %>%

mutate(

units\_in=

round(units\*bl\_prop\_in)) %>%

select(

bg\_id,

bl\_id,

NID,

bl\_prop\_in,

people,

people\_in,

units,

units\_in) -> bl\_counts

glimpse(bl\_counts)

```

### Calculate counts for block groups

```{r count-bl}

red %>%

group\_by(bg\_id) %>%

summarize(

people=sum(people),

units=sum(units)) -> count\_total

glimpse(count\_total)

bl\_counts %>%

group\_by(bg\_id, NID) %>%

summarize(

people\_in=sum(people\_in),

units\_in=sum(units\_in)) %>%

filter(NID > 0) -> count\_in

glimpse(count\_in)

count\_in %>%

inner\_join(count\_total, by="bg\_id") %>%

filter(NID > 0) -> bg\_counts

glimpse(bg\_counts)

```

###sum across blockgroups to get tracts counts

```{r}

bg\_counts %>%

mutate(tr\_id=str\_sub(bg\_id,1,11)) %>%

group\_by(NID,tr\_id) %>%

summarize(

people=sum(people),

people\_in=sum(people\_in),

units=sum(units),

units\_in=sum(units\_in)) -> tr\_counts

glimpse(tr\_counts)

```

### Save the information in an RData file.

```{r save}

save(

bg\_counts,

bg\_list,

bl\_counts,

bl\_list,

tr\_counts,

file="../Data/nbd-weights.RData")

```