An analysis of residential home values after the Flint water crisis

Ilse Paniagua and Zeping Tao

12/02/2019

Contents

This project needs the following libraries:

```
library(ZillowR)
library(censusapi)
library(tidyverse)
library(ggmap)
library(factoextra)
library(lubridate)
library(corrplot)
library(magrittr)
library(foreign)
library(XML)
library(tigris)
library(sf)
library(GGally)
```


Abstract

This project examines the effect of the Flint water crisis on residential home values between 2010-2017. Using data from the American Community Survey (ACS), tax assessment data (Zillow), and home sale prices (Zillow), we found that only the ACS measure of a home's value showed a sharp decline after the Flint water crisis. Averaged at the tract level, our results also revealed large differences between administrative records (tax assessments and home sale prices) and the ACS question that asks respondents to estimate their home value. These differences were between 10,000 and 50,000 dollars for all census tracts in Flint. Finally, we provide visualizations showing perceived home value changes in Flint by census tract. Future research could design an experiment to casually estimate the measurement error associated with asking respondents to provide their home values under conditions of potential market change.

0. Introduction

The Flint water crisis began in 2014, after the water supply of 100,000 people was switched from the Detroit to the Flint River. Because water was not properly treated, the city's population was exposed to elevated lead levels. Previous studies have explored the effects of the water crisis on children's lead levels (Hanna-Attisha, 2016) and depopulation in the city due to perceptions of unsafe water (Morckel and Greg Rybarczyk, 2018).

This project contributes this literature by analyzing the effect of the Flint water crisis on residential home values. We use data from the American Community Survey (https://census.gov/programs-surveys/acs/) and the Zillow real estate API (https://www.zillow.com/) to examine housing trends over time at the census tract level.

The ACS asks respondents to give an estimate of their home value ("About how much do you think this house and lot, appartment or mobile home (and lot, if owned) would sell for if it were for sale?"). However, we hypothesize that this question will be affected by measurement error. We argue that an individual is unlikely to know the exact value of their home at any given time. In addition to this potential failure to encode their home values, the ACS home value question may suffer from social desirability bias if an individual does not wish to be embarassed by disclosing a low price to the interviewer. It is also possible that a respondent will overestimate how much the water crisis has affected their home value and provide a price that is considerably lower than the price that their house would sell on the market at that time.

Our motivation for gathering data from the Zillow API is to obtain an objective measure of a home's value, separate from an individual's subjective perceptions. The Zillow API provides details on the latest tax

assessment conducted on a property (used to calculate property taxes), as well as the last price for which this home was sold. We use both metrics as objective indicators of value and compare these to ACS estimates.

Our research questions are as follows:

1. How did home values change in Flint, MI between 2010 and 2017?
2. Do self-report and Zillow estimates of home values agree?
3. Was the change in home values higher in areas with lead pipes?

The first section of this report describes ACS data gathering and cleaning steps; section 2 features ACS data exploration; section 3 describes the process of gathering and cleaning Zillow data; section 4 presents our results, divided by research question; and section 5 concludes.

1. Data Gathering from the American Community Survey

To gather data from the Census API, we obtain a key for the Census bureau API and save this as an R object.

```
cs_key <- "MY KEY"
```

The listCensusApis function provides a list of the various datasets provided by the census. We are interested in the American Community Survey 5-Year Data (2009-2017).

```
apis <- listCensusApis()

View(apis)

# 5 year detailed tables

ac5_vars <- listCensusMetadata(name = "acs/acs5",
    vintage = 2017, type = "variables")

ac5_geo <- listCensusMetadata(name = "acs/acs5",
    vintage = 2017, type = "geographies")
```

We obtain the variables of interest (from acs5_vars dataframe):

B19001B_001E: income for black households B19001A_001E: income for white households B01001B_017E: total female (black) B01001B_002E: total male (black) B01001B_001E: total black B01001A_001E: total white B01001A_002E: total male (white) B01001A_017E: total female (white) B01001_027E: female under 5 years B01001_003E: male under 5 years B11016_002E: family households (note: need to calculate proportion of family HH) B11016_001E: total number of households B25107_001E: median home value

B25034_010E: built 1940-1949 B25034_009E: built 1950-1959 B25034_008E: built 1960-1969 B25034_007E: built 1970-1979 B25034_006E: built 1980-1989 B25034_005E: built 1990-1999 B25034_004E: built 2000-2009

```
myvars <- c("B19001B_001E", "B19001A_001E",
    "B01001B_017E", "B01001B_002E", "B01001B_001E",
    "B01001A_001E", "B01001A_002E", "B01001A_017E",
    "B01001_027E", "B01001_003E", "B11016_002E",
    "B11016_001E", "B25107_001E")
```

The state of Michigan is 026, and Genessee county is 049. The following code chunk retrieves variables that are only needed once (housing structure age).

```
# 5 year detailed tables

# Variables that are only needed once
# (municipality names and housing
# structure age)

flint_strage <- getCensus(name = "acs/acs5",
    vintage = 2010, vars = c("NAME", "B25034_010E",
        "B25034_009E", "B25034_008E", "B25034_007E",
        "B25034_006E", "B25034_005E", "B25034_004E"),
    region = "tract:*", regionin = "state:26+county:049",
    key = cs_key)

colnames(flint_strage) <- c("state", "county",
    "tract", "desc", "1940-1949", "1950-1959",
    "1960-1969", "1970-1979", "1980-1989",
    "1990-1999", "2000-2009")

glimpse(flint_strage)
```

Now we gather data for the years 2010-2017 for the variables that would change from year to year.

```
# Loop for years 2010-2017
flint_acs5 <- matrix()

years <- c("2010", "2011", "2012", "2013",
    "2014", "2015", "2016", "2017")

for (i in years) {

    flint_partial <- getCensus(name = "acs/acs5",
        vintage = i, vars = myvars, region = "tract:*",
        regionin = "state:26+county:049",
        key = cs_key)

    colnames(flint_partial) <- c("state",
        "county", "tract", "income_black",
        "income_white", "f_black", "m_black",
        "black", "white", "m_white", "f_white",
        "f_under5", "m_under5", "fam_hh",
        "hh", "homevalue")

    # Male under 5: B18101_003E

    flint_acs5 <- cbind(flint_acs5, flint_partial)
}
```

Census Data Cleaning

We have 129 columns, some of which repeat across years. This step will help us organize variables for reshaping and will add clarity to variable names.

```r
# Removing empty first column
flint_acs5_clean <- flint_acs5[, 2:129]

# Removing multiple cases of state and
# municipality ID
flint_acs5_clean <- flint_acs5_clean %>%
    select(-matches("state*|county*|tract*|desc*"))

# Adding years

# .1 = 2011, .2=2012.....7=2017

nums <- c(1, 2, 3, 4, 5, 6, 7)

for (col in colnames(flint_acs5_clean)) {
    for (i in nums) {
        colnames(flint_acs5_clean)[colnames(flint_acs5_clean) ==
            col] <- gsub(paste("\\.", nums[i],
            sep = ""), paste("_201", nums[i],
            sep = ""), col)
    }
}

# Changing name of 2010 (col 1:13)

colnames(flint_acs5_clean)[1:13] <- paste(colnames(flint_acs5_clean[1:13]),
    "_2010", sep = "")


# Adding structure age data
flint_acs5_clean <- cbind(flint_strage, flint_acs5_clean)
```

Now we will keep only the census tracts that are inside of Flint.

```r
# Keeping only tracts of interest
tracts <- (c(100, 1000, 1100, 1200, 11713,
    13500, 13600, 1400, 1500, 1600, 1700,
    1800, 1900, 200, 2000, 2200, 2300, 2400,
    2600, 2700, 2800, 2900, 300, 3000, 3100,
    3200, 3300, 3400, 3500, 3600, 3700, 3800,
    400, 4000, 500, 600, 700, 800, 900))

# Full data
flint_acs5_clean$tract <- as.numeric(flint_acs5_clean$tract)

flint_acs5_clean <- flint_acs5_clean %>%
    filter(tract %in% tracts)

# Structure age data
flint_strage$tract <- as.numeric(flint_strage$tract)

flint_strage <- flint_strage %>% filter(tract %in%
    tracts)
```

We can now restructure our data into tidy format, with one column per variable and one row per observation.

```
# Gathering data

# Full dataset
flint_acs5_clean <- flint_acs5_clean %>%
    gather(5:11, key = "built", value = "cases") %>%
    arrange(., tract)

# Structure age data
flint_strage <- flint_strage %>% gather(5:11,
    key = "built", value = "cases") %>% arrange(.,
    tract)

# Creating year variable
flint_acs5_clean <- flint_acs5_clean %>%
    select(noquote(order(colnames(.)))) %>%
    select(state, county, tract, desc, built,
        cases, everything())

# Reshaping data columns 7-110
acs_vars <- c("income_black", "income_white",
    "f_black", "m_black", "black", "white",
    "m_white", "f_white", "f_under5", "m_under5",
    "fam_hh", "hh", "homevalue") %>% sort(decreasing = FALSE)

flint_acs5_clean_full <- flint_acs5_clean %>%
    reshape(idvar = c("tract", "built"),
        varying = list(c(7:14), c(15:22),
            c(23:30), c(31:38), c(39:46),
            c(47:54), c(55:62), c(63:70),
            c(71:78), c(79:86), c(87:94),
            c(95:102), c(103:110)), v.names = acs_vars,
        timevar = "year", times = years,
        direction = "long") %>% arrange(.,
    tract)

glimpse(flint_acs5_clean_full)
```

Now we create a dataset with only one row per tract and year.

```
# Dataset that doesn't include structure
# age

flint_acs5_series <- flint_acs5_clean_full %>%
    select(-built, -cases) %>% unique.data.frame()

glimpse(flint_acs5_series)
```

We now have three tidy datasets:

1. Housing structure age by tract (273 rows= 39 tracts* 7 structure age)
2. Demographic variables and housing structure by tract (2,184 rows= 39 tracts* 7 structure age * 8 years)

3. Demographic variables by tract (312 rows= 39 tracts * 8 years)

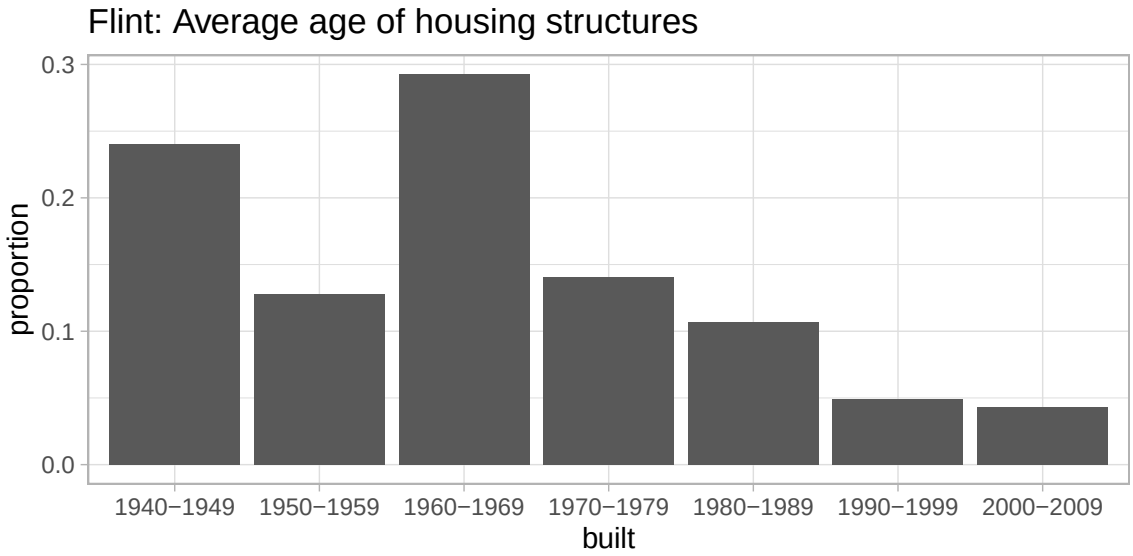
2. Census Data Exploration

This section contains graphics to help us understand demographic changes in Flint from 2010-2017. We will add a spatial component to these descriptives in the next sections.

- Age of housing structures, by tract

Calculate the total number of houses per tract. Then, calculate what percentage belong to each category, per tract. Aggregate the results in a bar chart for each age group.

```
# Barchart, number of cases in each
# category

flint_strage %>% group_by(tract) %>% mutate(numhouses = sum(cases)) %>%
    group_by(built, tract) %>% summarise(mean = cases/numhouses) %>%
    group_by(built) %>% summarise(proportion = mean(mean)) %>%
    ggplot(., aes(built, proportion)) + geom_col() +
    theme_light() + labs(title = "Flint: Average age of housing structures")
```



Flint has a large proportion of old homes built between 1940-49, followed by homes built between 1960-69.

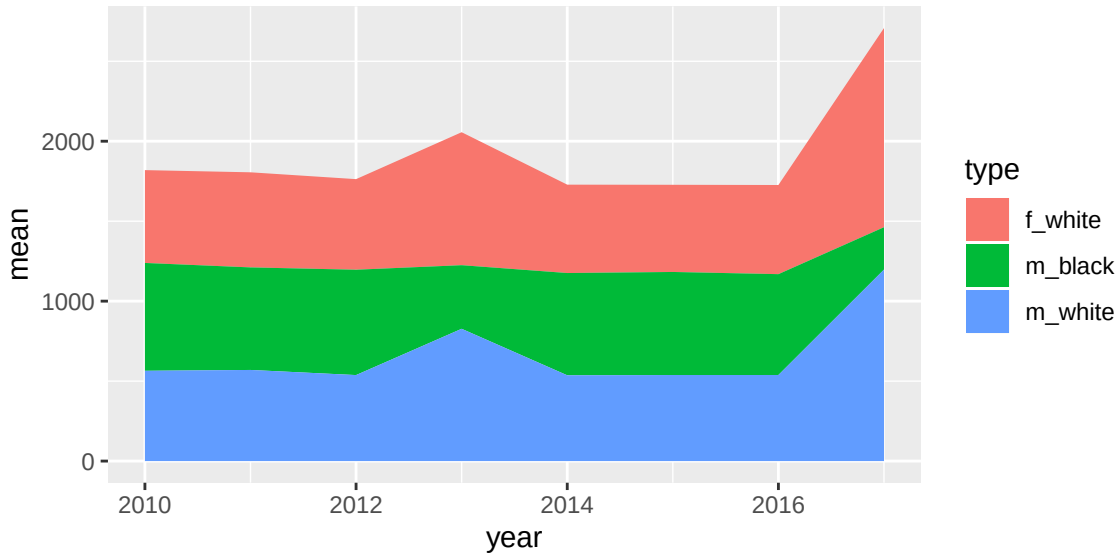
- Population trends over time

```
flint_acs5_series$year <- flint_acs5_series$year %>%
    as.numeric()
```

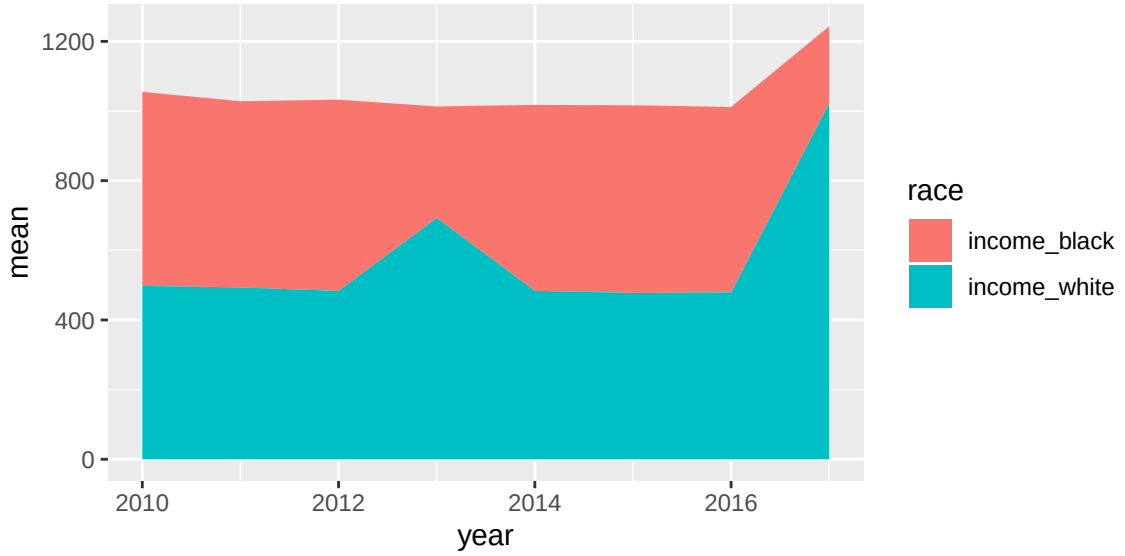
Population trends seem steady over time. The white population seems to have increased after 2017. This, combined with rising house prices in 2017, may point to some gentrification.

- Population trends over time

```
# Changing year to numeric for time
# series
flint_acs5_series$year <- flint_acs5_series$year %>%
    as.numeric()

# Adult males and females, by race
flint_acs5_series %>% gather(6, 8:10, 15:18,
    key = "type", value = "population") %>%
    arrange(., tract) %>% filter(type %in%
    c("m_white", "f_white", "f_black", "m_black",
        "type", "population")) %>% group_by(year,
    type) %>% summarise(mean = mean(population)) %>%
    ggplot(aes(x = year, y = mean)) + geom_area(aes(fill = type))
```



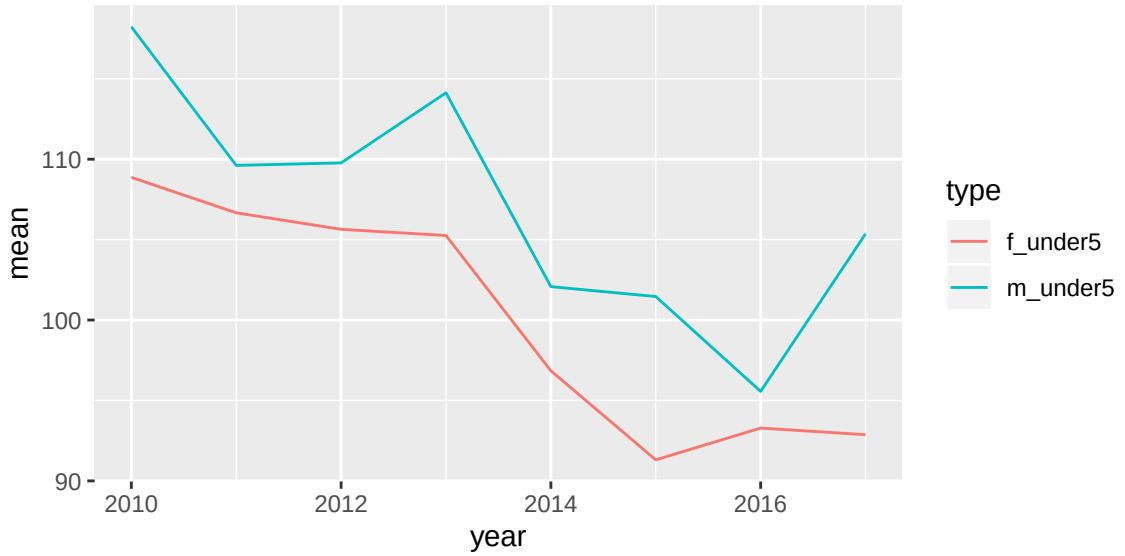

```
# Income variable
flint_acs5_series %>% gather(13:14, key = "race",
    value = "income") %>% arrange(., tract) %>%
    group_by(year, race) %>% summarise(mean = mean(income)) %>%
    ggplot(aes(x = year, y = mean)) + geom_area(aes(fill = race))
```



Income seems relatively stable from 2010-2017.

- Number of children under 5

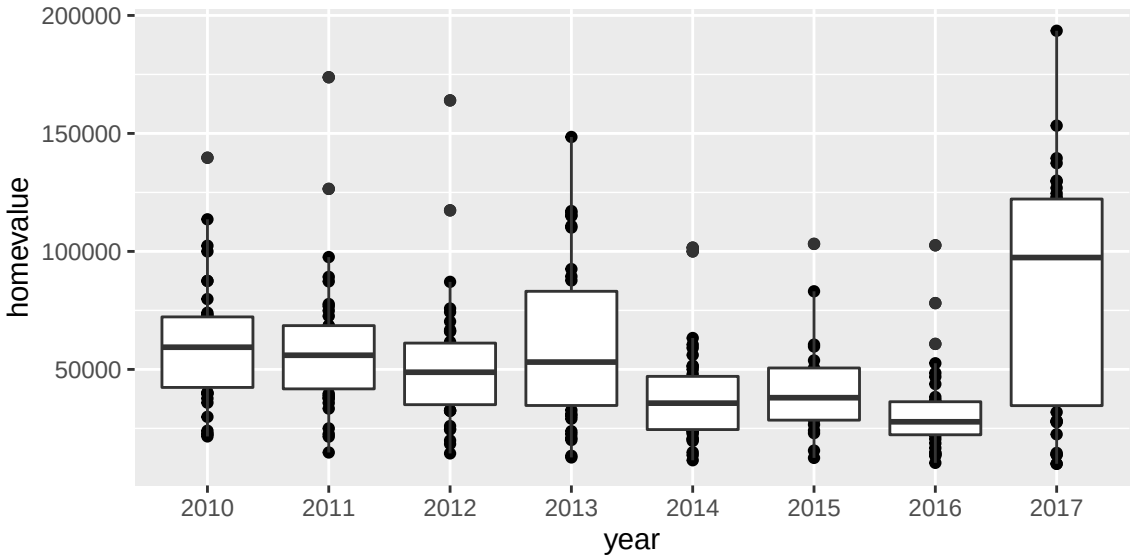
```
# Children variable

flint_acs5_series %>% gather(6, 8:10, 15:18,
    key = "type", value = "population") %>%
    arrange(., tract) %>% filter(type %in%
    c("m_under5", "f_under5", "type", "population")) %>%
    group_by(year, type) %>% summarise(mean = mean(population)) %>%
    ggplot(aes(x = year, y = mean)) + geom_line(aes(color = type))
```



The average numer of children per tract declined slightly between 2013 and 2016.

- Trends in housing values over time

```
# Housing values over time

flint_acs5_series$year <- flint_acs5_series$year %>%
    as.character()

flint_acs5_series %>% ggplot(., aes(x = year,
    y = homevalue)) + geom_point() + geom_boxplot()
```



ACS data show a perceived decline in home values from 2014-2016. After 2017, home values seem to have increased drastically.

3. Gathering Zillow data

The "ZillowR" package provides an interface to the Zillow API. However, we must specify an address as an input parameter. We use the TIGER/Line Shapefile to obtain potential addresses in Flint and try them with Zillow api.

First, we load the address ranges from "Address Ranges County-based Relationship File", which contains "Address range identifier"(ARID). Then we load the feature names (street names) from "Feature Names County-based Relationship File", which contains "Linear feature identifier"(LINEARID). Lastly, we load "Address Range-Feature Name County-based Relationship File" that describes links between ARID and LINEARID.

Then, we set up the Zillow API keys. Since one Zillow API key only allows 1000 request per day, we registered 5 keys for a more efficient data gathering. We also set up a function to turn the "character(0)" and NULL value gathered into an "NA".

We randomly sample 250 records out of our 44,502 address frame per day and iterate them for API request. Although our daily sample may have overlapping records across days, our sampling rate is 0.5% (250 out of 44,502) so the overlapping records would be small enough to ignore.

```
frame <- readRDS("Address Frame.RData")
set.seed(Sys.Date())
smp_frame <- sample_n(frame, 250)
```

Then we set up a loop to try to obtain information from each address in our sample of possible addresses.

We gathered 9,351 records in a tibble with 27 variables. After removing duplicates using the unique Zillow property ID (ZPID), We are left with 5,843 unique records.

```
length(unique(table_zlw$ZPID))
# 5843
table_zlw_nondup <- table_zlw %>% distinct(ZPID,
    .keep_all = TRUE)

saveRDS(table_zlw_nondup, "Zillow data with distinct ZPID.Rdata")
```

Cleaning Zillow Data

```
zillow <- readRDS("Zillow data with distinct ZPID.Rdata")
```

Here, we double-check that there are unique records in the dataset, covert long and lat to numeric, and convert dates from character to date format.

```
# Keeping unique records of the df
zillow <- zillow[!duplicated(zillow[, c("ZPID")]),
    ]

# Changing longitude and latitude as
# numeric.
zillow$lat %<>% as.numeric()
zillow$long %<>% as.numeric()

# Extract year from last sold date

zillow$lastSoldYear <- zillow$lastSoldDate %>%
    str_extract_all("\\d{4}$") %>% as.character()
```

We will use Zillow records in our analysis if:

1) Their most recent tax assessment is between 2010 and 2017 (use taxAssessment price)
2) Their last sale date is between 2010 and 2017 (use lastSalePrice)

When there is both tax assessment and house sales for the same property in different years, both will be used for analysis.

Using the criteria above, we have 2,066 observations.

```
# Saving records for analysis
years <- c("2010", "2011", "2012", "2013",
    "2014", "2015", "2016", "2017")

zillow_analysis <- zillow %>% filter(lastSoldYear %in%
    years | taxAssess_year %in% years)
```

Now, we link the Zillow addresses to census tracts using Zillow-provided latitude and longitude and the tigris R package (call_geolocator_latlon).

```
zillow_analysis$geo <- apply(zillow_analysis,
    1, function(row) call_geolocator_latlon(row["lat"],
        row["long"]))  #geolocator will be used for leaflet mapping later
```

```
zillow_analysis$tract <- zillow_analysis$geo %>%
  substr(6,11) %>% #subsetting string
  as.numeric()
```

We perform an inner join on the Zillow and ACS data. Because some Zillow addresses were not within the city bounds of Flint, we now have 815 records that are in our years of interest and could be matched to a Flint census tract.

```
# Inner join Zillow and ACS data Filter
# merged data just to records where there
# was a tax assessment or home sale

merged <- inner_join(zillow_analysis, flint_acs5_series,
    by = "tract") %>% filter(year == taxAssess_year |
    year == lastSoldYear)

# Which Zillow addresses are not on the
# ACS tract list?
anti_join(zillow_analysis, flint_acs5_series,
    by = "tract")
```

Now, we create buckets of categories for home values, grouped by ten thousand (10ks). We create categories for the ACS homevalue variable, the Zillow-provided tax assessment, and the Zillow-provided last sale price variable.

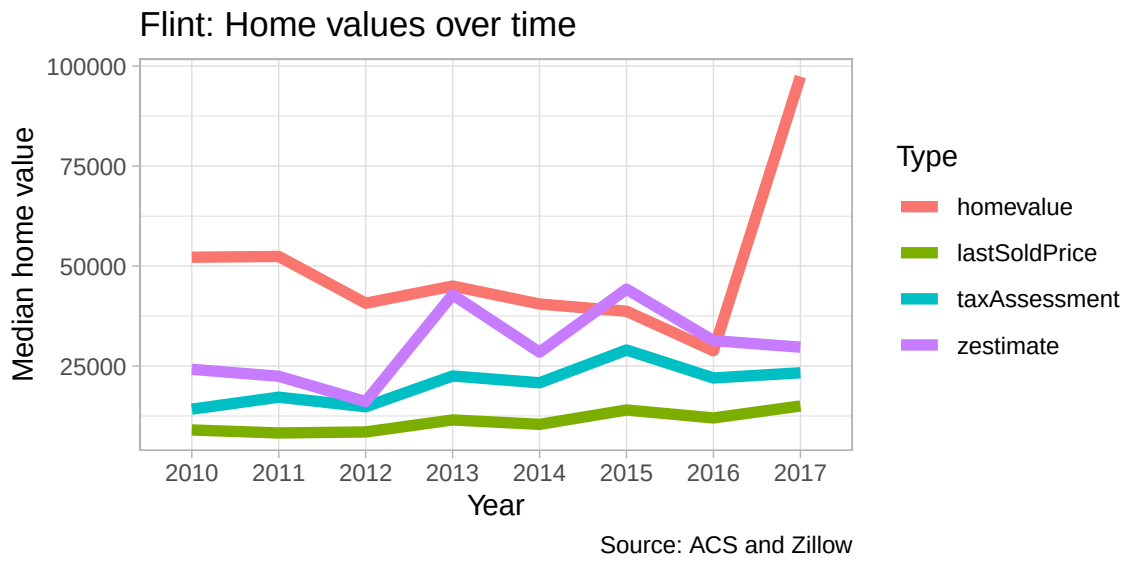
```
# Home value categorical variable
homecats <- c(-Inf, 10000, 20000, 30000,
    40000, 50000, 60000, 70000, 80000, Inf)

# ACS homevalue variable
merged %<>% mutate(homecat = cut(homevalue,
    breaks = homecats, labels = c("0-10",
        "11-20", "21-30", "31-40", "41-50",
        "51-60", "61-70", "71-80", "80andabove")))

# Zillow tax assessment variable
merged %<>% mutate(taxcat = cut(taxAssessment,
    breaks = homecats, labels = c("0-10",
        "11-20", "21-30", "31-40", "41-50",
        "51-60", "61-70", "71-80", "80andabove")))

# Zillow home sales variable
merged %<>% mutate(salevaluecat = cut(lastSoldPrice,
    breaks = homecats, labels = c("0-10",
        "11-20", "21-30", "31-40", "41-50",
        "51-60", "61-70", "71-80", "80andabove")))
```

4. Results

Research question 1: How have home prices changed over time?

We chart changes over time for all three categories (ACS value, tax value, home sale value) across all census tracts. We use the median (not mean) because of potential outliers in the data.

```
merged %>% select(year, taxAssessment, lastSoldPrice,
    homevalue, zestimate) %>% gather(key = type,
    value = value, 2:5) %>% group_by(year,
    type) %>% summarise(median = median(value,
    na.rm = TRUE)) %>% ggplot(aes(x = year,
    y = median)) + geom_line(aes(color = type,
    group = type), size = 2) + labs(title = "Flint: Home values over time",
    x = "Year", y = "Median home value",
    color = "Type", caption = "Source: ACS and Zillow") +
    theme_light()
```



We see a large difference between ACS, tax assessment, and home sale values. We will explore this further with research question 2.

Research question 2: Do Zillow and ACS prices match?

Do ACS category and Zillow categories match, on average by census tract?

```
# Match between tax assessment and ACS
merged %<>% mutate(match_tax = case_when(homecat ==
    taxcat & year == taxAssess_year ~ 1,
    homecat != taxcat & year == taxAssess_year ~
        0))

# Last home sale
merged %<>% mutate(match_sale = case_when(homecat ==
```

```
    salevaluecat & year == lastSoldYear ~
    1, homecat != salevaluecat & year ==
    lastSoldYear ~ 0))
```

What is the average discrepancy between reported values and administrative records? We also include Zillow zestimates as a comparison (although it is unclear for which year these estimates were generated).

```
merged %>% mutate(match_diff_tax = homevalue -
    taxAssessment, match_diff_sale = homevalue -
    lastSoldPrice, zest_diff = homevalue -
    zestimate) %>% group_by(tract) %>% summarise(mean(match_diff_tax,
    na.rm = TRUE), mean(match_diff_sale,
    na.rm = TRUE), mean(zest_diff, na.rm = TRUE))
```

```
## # A tibble: 37 x 4
##    tract `mean(match_diff_tax,~ `mean(match_diff_sale~ `mean(zest_diff, na~
##    <dbl>                  <dbl>                  <dbl>                <dbl>
##  1   100                 34525                  40528.               26128.
##  2   200                 32112.                 34293.               24800.
##  3   300                 31948.                 33914.               32566.
##  4   400                 25800.                 21333                16857
##  5   500                 39734.                 42480.               35286.
##  6   600                 16287.                 21895.               12556.
##  7   700                 30836.                 43321.               40076.
##  8   800                 23438.                 35622.               22227.
##  9   900                 29986.                 37987.               40810.
## 10  1000                 33935                  41970                26874.
## # ... with 27 more rows
```

The average difference between ACS home values and Zillow tax assessments and home sale prices is upwards of 20K to 50k!

It is not clear that Zestimates are more accurate. The difference is higher or lower than administrative records depending on the tract.

Now, we create a summary table for average matches by census tract and number of records for each comparison.

```
# Matches by tract
merged %>% group_by(tract) %>% summarise(mean_tax = mean(na.omit(match_tax)),
    mean_sale = mean(na.omit(match_sale)),
    count_tax = sum(!is.na(match_tax)), count_sale = sum(!is.na(match_sale)))
```

```
## # A tibble: 37 x 5
##    tract mean_tax mean_sale count_tax count_sale
##    <dbl>    <dbl>     <dbl>     <int>      <int>
##  1   100   0         0.136         10         22
##  2   200   0         0             20          4
##  3   300   0         0             27         23
##  4   400   0         1              6          1
##  5   500   0         0             22          7
##  6   600   0.0667    0             30          8
##  7   700   0         0.0833        11         12
```

```
##  8   800   0.0455    0             22          7
##  9   900   0.0333    0             30         33
## 10  1000   0         0             17          3
## # ... with 27 more rows
```

Because we do not have enough records per tract to conduct a robust analysis, we will cluster observations using both Zillow and Census data and make comparisons across clusters instead of census tracts.

Creating clusters

We will use hierarchical clustering as a way to reduce the number of tracts we have for the analysis. We use hiererchical clustering instead of k-means to avoid discarding more observations (our dataset has missing values and K-Means requires complete records).

We will cluster variables based on:

Number of bedrooms, number of bathrooms, finishedsqFt, rentzestimate, lat, long, number of black households, number of white households, and number of family households.

We need to rescale variables before the analysis.

```
clusters <- merged %>% select(bedrooms, bathrooms,
    finishedSqFt, rentzestimate, lat, long,
    black, white, fam_hh) %>% mutate_all(scale)
```

Next, create the distance matrix of the cleaned data.

```
hclust_d <- dist(clusters)
as.matrix(hclust_d)[1:10, 1:10]
```

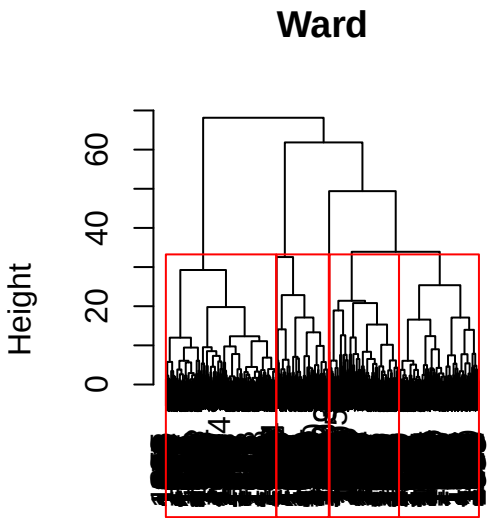
```
##           1         2        3         4        5        6        7
## 1  0.000000 1.1280527 1.335669 1.3290487 1.748808 1.812737 3.312808
## 2  1.128053 0.0000000 1.931085 0.9822091 1.271126 1.759387 4.067468
## 3  1.335669 1.9310846 0.000000 2.0113944 2.170311 1.343952 3.188236
## 4  1.329049 0.9822091 2.011394 0.0000000 2.057420 2.023810 4.634642
## 5  1.748808 1.2711259 2.170311 2.0574196 0.000000 1.396247 3.951446
## 6  1.812737 1.7593873 1.343952 2.0238098 1.396247 0.000000 3.603402
## 7  3.312808 4.0674685 3.188236 4.6346416 3.951446 3.603402 0.000000
## 8  5.017009 5.3771654 4.941492 6.3070995 5.194532 5.037954 2.622459
## 9  3.065977 3.2814527 3.791095 3.1293261 3.454157 3.728355 2.930971
## 10 3.282752 3.9681564 3.625409 4.3164605 3.885112 3.941401 1.224574
##           8        9       10
## 1  5.017009 3.065977 3.282752
## 2  5.377165 3.281453 3.968156
## 3  4.941492 3.791095 3.625409
## 4  6.307099 3.129326 4.316460
## 5  5.194532 3.454157 3.885112
## 6  5.037954 3.728355 3.941401
## 7  2.622459 2.930971 1.224574
## 8  0.000000 3.903379 2.823823
## 9  3.903379 0.000000 2.352448
## 10 2.823823 2.352448 0.000000
```

This distance matrix can be used to cluster counties, e.g. using the ward method.

```
# Using the Ward method
hc_ward <- hclust(hclust_d, method = "ward.D2")
```

Ploting the dendogram to find a reasonable number of clusters.

```
plot(hc_ward, main = "Ward", xlab = "", sub = "")

rect.hclust(hc_ward, k = 5, border = "red")
```



We will select a small number of clusters (5) to have enough records per cluster.

```
# Number of cases per cluser
merged %>% mutate(cluster = cutree(hc_ward,
    5)) %>% group_by(cluster) %>% summarise(count = n())
```

```
## # A tibble: 5 x 2
##   cluster count
##     <int> <int>
## 1       1   460
## 2       2   219
## 3       3   333
## 4       4   290
## 5       5     2
```

```
# Adding cluster column
merged %<>% mutate(cluster = cutree(hc_ward,
    5))
```

Cluster 5 only contains 2 observations and will not be used for analysis.

Now, we again calculate the differences in ACS home values and administrative records.

```
# Matches by cluster
merged %>% mutate(match_diff_tax = homevalue -
    taxAssessment, match_diff_sale = homevalue -
    lastSoldPrice) %>% filter(!cluster %in%
    c(5)) %>% group_by(cluster) %>% summarise(`Matches by Tax (%)` = mean((match_tax),
    na.rm = TRUE), `Matches by Sale Price (%)` = mean((match_sale),
    na.rm = TRUE), `Number of tax records` = sum(!is.na(match_tax)),
    `Number of sale records` = sum(!is.na(match_sale)),
    `Average difference - Tax` = mean(match_diff_tax,
        na.rm = TRUE), `Average difference - Sales` = mean(match_diff_sale,
        na.rm = TRUE))
```

```
## # A tibble: 4 x 7
##   cluster `Matches by Tax~ `Matches by Sal~ `Number of tax ~
##     <int>            <dbl>            <dbl>            <int>
## 1       1           0.0228          0.00658              307
## 2       2           0               0.0667                29
## 3       3           0.306           0.0769               134
## 4       4           0               0.0452               127
## # ... with 3 more variables: `Number of sale records` <int>, `Average
## #   difference - Tax` <dbl>, `Average difference - Sales` <dbl>
```

We still observe very large differences between randomly sampled addresses that returned a Zillow record and ACS data!

These differences vary by tract but are again in the range of 10,000-30,000 different. Cluster four seems to have performed the best in terms of matches between ACS and Zillow data.

Research question 3: Change in home values in neighborhoods with lead tracts

References: https://rpubs.com/ben_bellman/sf_tigris

Mapping census tracts

Getting census tract files from tigris package and changing to a simpler sf format.

```
flint_geo <- tracts(state = "MI", county = "Genesee")

flint_geo %<>% st_as_sf(ri)
```

Keep only polygons that are in Flint.

```
flint_geo %<>% mutate(tract = as.numeric(TRACTCE))

test <- flint_geo %>% filter(tract %in% merged$tract) %>%
    select(tract, geometry)
```

Merging geometry with ACS dataset, as well as the Zillow-ACS dataset.

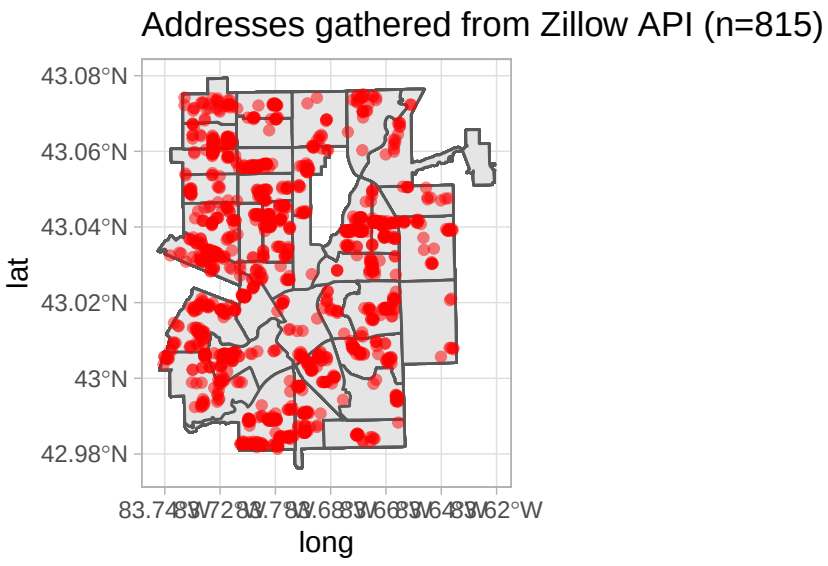
```
acs_geo <- full_join(flint_acs5_clean, test,
    by = "tract")

merged_geo <- full_join(merged, test, by = "tract")

# Every tract merged!
anti_join(test, flint_acs5_clean, by = "tract")
```

```
## Simple feature collection with 0 features and 1 field
## bbox:           xmin: NA ymin: NA xmax: NA ymax: NA
## epsg (SRID):    NA
## proj4string:    +proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +no_defs
## [1] tract    geometry
## <0 rows> (or 0-length row.names)
```

```
# Converting to a simpler sf format for
# visualization
acs_geo <- st_as_sf(acs_geo)

merged_geo <- st_as_sf(merged_geo)
```

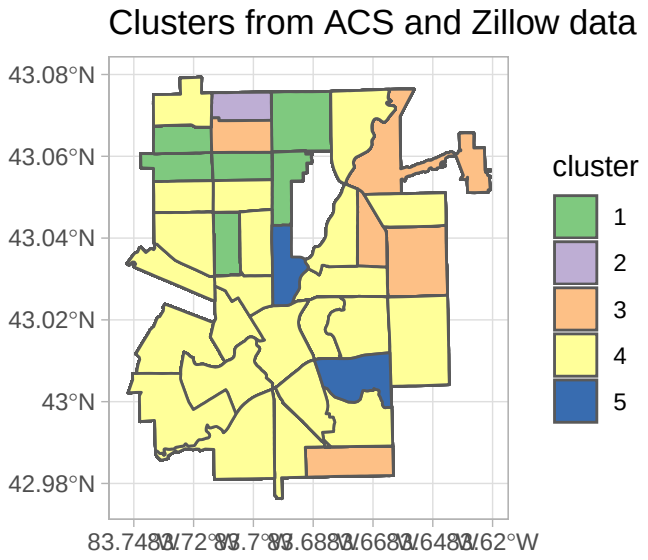
Now visualizations! Here, we map the property addresses from Zillow that we gathered.

```
# Gathered obs.
ggplot(data = merged_geo) + geom_sf() + geom_point(data = merged_geo,
    aes(x = long, y = lat), size = 1.5, alpha = 0.5,
    color = "red") + theme_light() + labs(title = "Addresses gathered from Zillow API (n=815)")
```



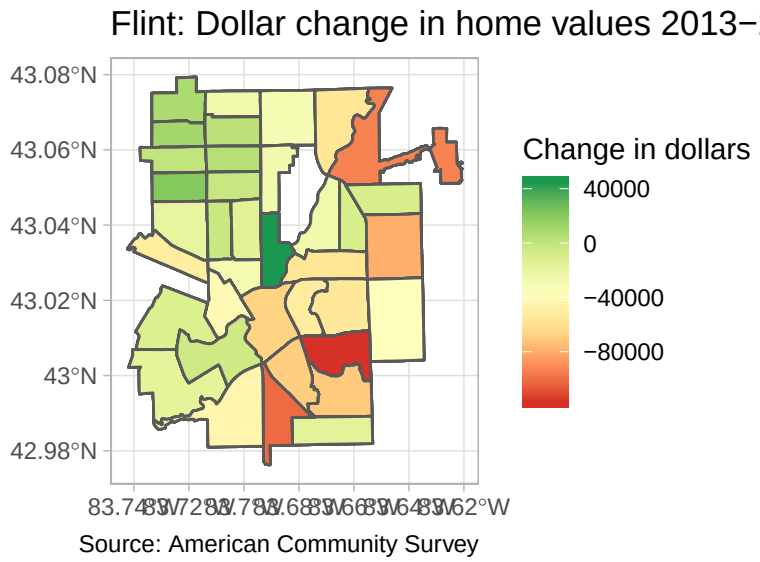
These are the clusters that we used for research question 2:

```
merged_geo %>% mutate(cluster = as.factor(cluster)) %>%
    ggplot() + geom_sf(aes(fill = cluster)) +
    scale_fill_brewer(palette = "Accent") +
    theme_light() + labs(title = "Clusters from ACS and Zillow data")
```

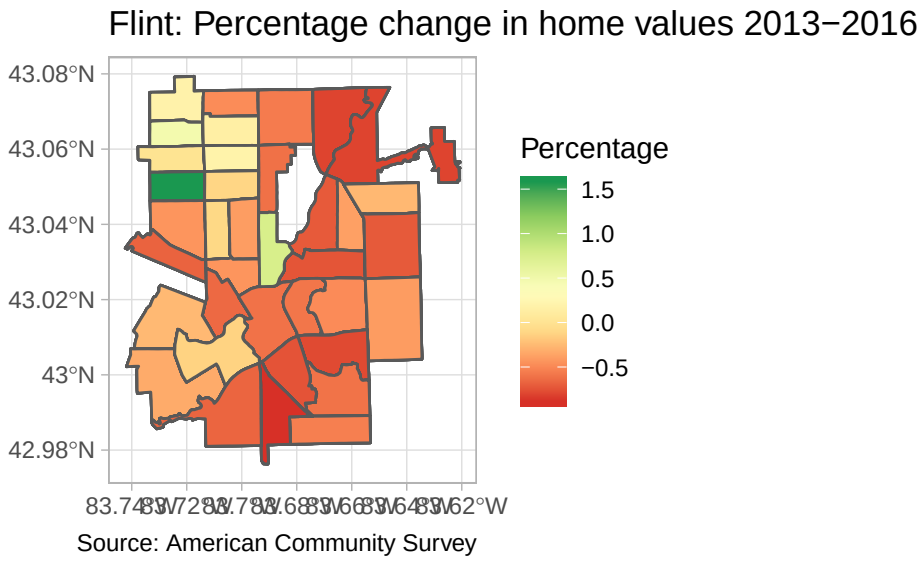


We visualize the change in home values from 2013 to 2016, roughly when the Flint water crisis became public. For both figures, a positive value means an increase and a negative value means a decrease in home values after these 3 years.

```
# Map of change in home values (dollars)
acs_geo %>% mutate(home_change = homevalue_2016 -
    homevalue_2013) %>% select(home_change,
    geometry) %>% ggplot() + geom_sf(aes(fill = home_change)) +
    scale_fill_distiller(palette = "RdYlGn",
        direction = 1) + theme_light() +
    labs(title = "Flint: Dollar change in home values 2013-2016",
        caption = "Source: American Community Survey",
        fill = "Change in dollars")
```




```
# Map of change in home values (percent)

acs_geo %>% mutate(home_change_pct = (homevalue_2016 -
    homevalue_2013)/homevalue_2013) %>% ggplot() +
    geom_sf(aes(fill = home_change_pct)) +
    scale_fill_distiller(palette = "RdYlGn",
        direction = 1) + theme_light() +
    labs(title = "Flint: Percentage change in home values 2013-2016",
        caption = "Source: American Community Survey",
        fill = "Percentage")
```



The map above shows that the northwest part of Flint in particular saw a decline in home values between 2013 and 2016. Some areas of the city appear to have increased their home values.

We saw that cluster 4 was able to match the ACS subjective home value variable better than the other clusters. Perhaps this is because cluster 4 also saw a net decrease in home values, and this matched the expectations of those residents.

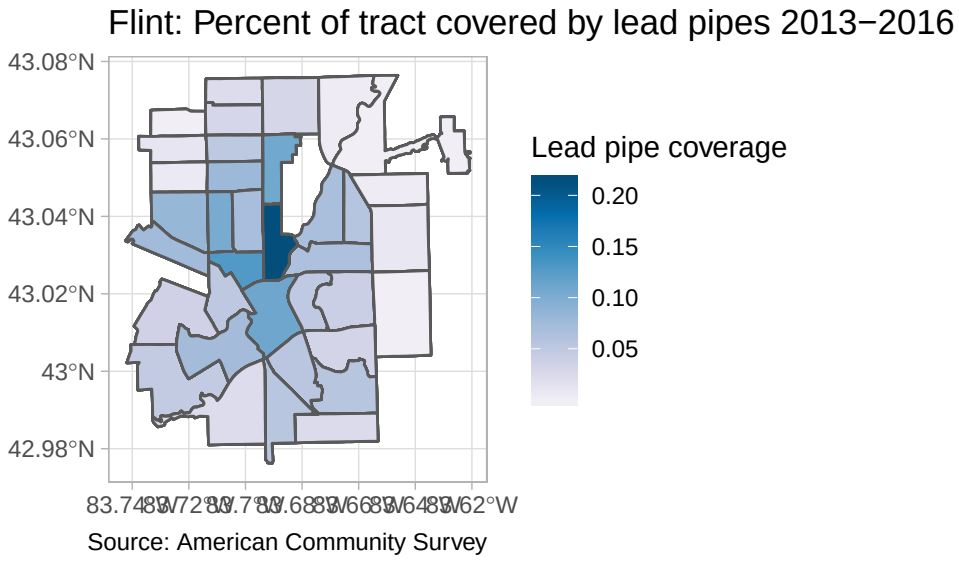
Lead pipe data

We use ArcGIS to calculate the lead-pipe-influenced area and census tract area and then spatially join them. Then we load the data from ArcGIS from a ".dbf" file. The measure we use is the coverage rate, which is the sum of lead-pipe-influenced area in a tract divided by the area of that census tract. Our hypothesis is that the higher the coverage rate, the more possible that the housing prices in that census tract are affected.

```
pipe_raw <- read.dbf("Lea_pipes_for_area_cal.dbf",
    as.is = TRUE)

# Calculate percentage of tract that is
# covered by lead pipes
pipe_coverage <- pipe_raw %>% group_by(TRACTCE) %>%
    mutate(sum_pipe_area = sum(Area), tract = as.numeric(TRACTCE)) %>%
    group_by(tract) %>% mutate(coverage = sum_pipe_area/Tract_area) %>%
    summarise(coverage = mean(coverage))
```

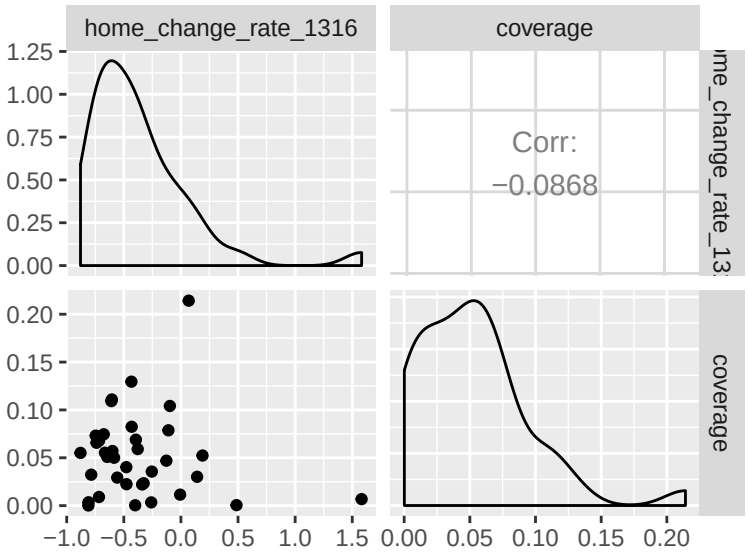
We calculate how much area of each tract is covered by lead pipes. As much as 20% of one tract has lead pipes!

```
acs_geo %>% merge(pipe_coverage, by = "tract") %>%
    ggplot() + geom_sf(aes(fill = coverage)) +
    scale_fill_distiller(palette = "PuBu",
        direction = 1) + theme_light() +
    labs(title = "Flint: Percent of tract covered by lead pipes 2013-2016",
        caption = "Source: American Community Survey",
        fill = "Lead pipe coverage")
```



Now we obtain a correlation to answer our research question.

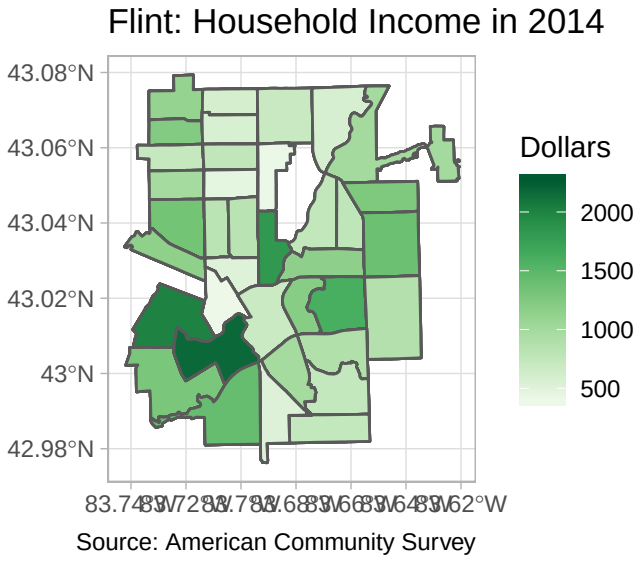
```
flint_acs5_clean %>% mutate(home_change_rate_1316 = (homevalue_2016 -
    homevalue_2013)/homevalue_2013) %>% merge(pipe_coverage,
    by = "tract") %>% select(home_change_rate_1316,
    coverage) %>% unique() %>% ggpairs()
```



There is a very low correlation between lead tract coverage and percentage change in home values.

Here, we explore other spatial trends to help us understand the demographic characteristics of southeast Flint.

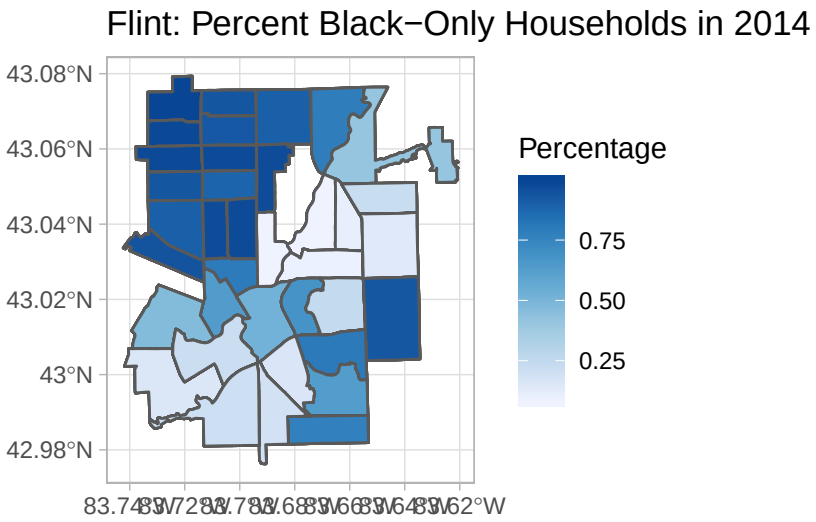
```
# Map of household income
acs_geo %>% mutate(totalinc = income_black_2014 +
    income_white_2014) %>% ggplot() + geom_sf(aes(fill = totalinc)) +
    scale_fill_distiller(palette = "Greens",
        direction = 1) + theme_light() +
    labs(title = "Flint: Household Income in 2014",
        caption = "Source: American Community Survey",
        fill = "Dollars")
```



Flint is overall a poor city. There is no obvious spatial pattern for income.

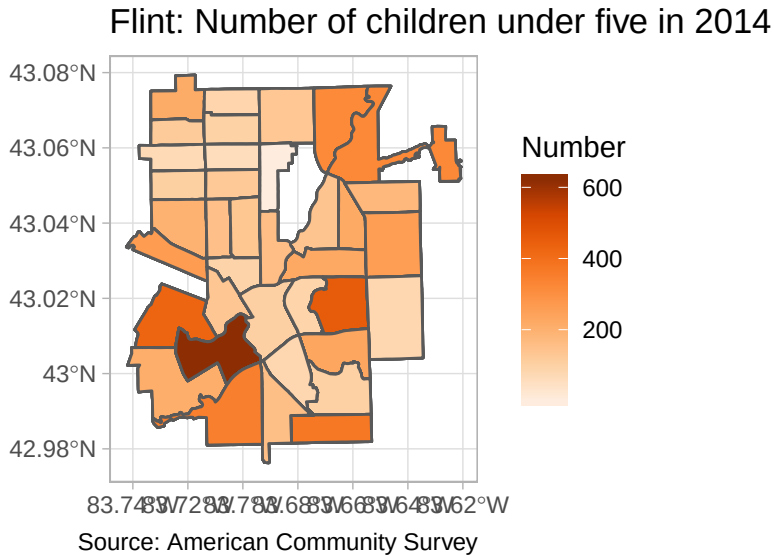
Now we examine the racial composition of the city. We know that Flint is roughly a half white, half black city in terms of its population. The northwest part of the city has a higher proportion of black-only households.

```
acs_geo %>% mutate(pctblack = black_2014/(black_2014 +
    white_2014)) %>% ggplot() + geom_sf(aes(fill = pctblack)) +
    scale_fill_distiller(palette = "Blues",
        direction = 1) + theme_light() +
    labs(title = "Flint: Percent Black-Only Households in 2014",
        caption = "Source: American Community Survey",
        fill = "Percentage")
```



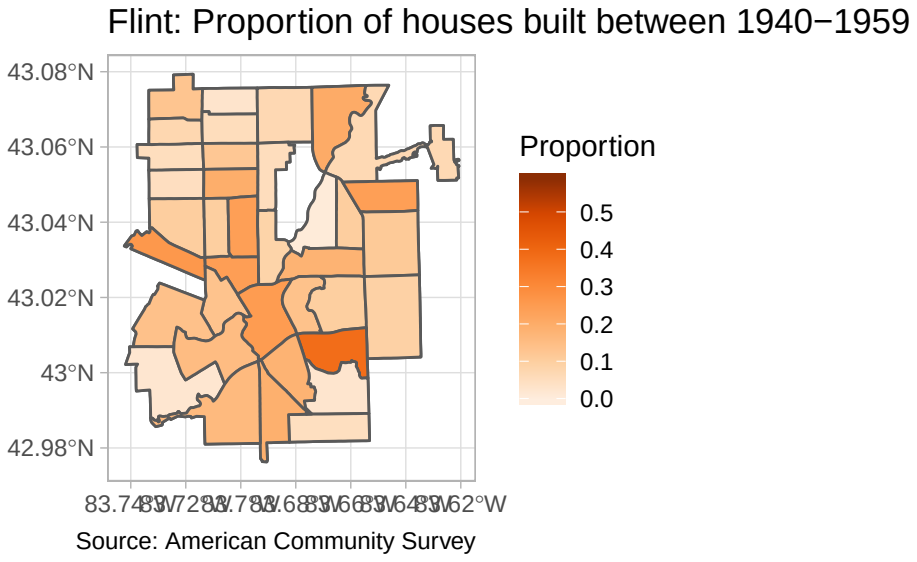
Do some areas have more children under the age of five? This population is especially affected by lead in their system.

```
acs_geo %>% mutate(under5 = f_under5_2014 +
    m_under5_2014) %>% ggplot() + geom_sf(aes(fill = under5)) +
    scale_fill_distiller(palette = "Oranges",
        direction = 1) + theme_light() +
    labs(title = "Flint: Number of children under five in 2014",
        caption = "Source: American Community Survey",
        fill = "Number")
```



Which areas of Flint have the oldest houses? Houses built between 1940 and 1959 appear to be about evenly spread across the city.

```
acs_geo %>% group_by(tract) %>% mutate(numhouses = sum(cases)) %>%
    group_by(built, tract) %>% summarise(mean = cases/numhouses) %>%
    filter(built %in% c("1940-1949", "1950-1959")) %>%
    ggplot() + geom_sf(aes(fill = mean)) +
    scale_fill_distiller(palette = "Oranges",
        direction = 1) + theme_light() +
    labs(title = "Flint: Proportion of houses built between 1940-1959",
        caption = "Source: American Community Survey",
        fill = "Proportion")
```



5. Conclusions

We found that only the ACS measure of a home's value showed a sharp decline after the Flint water crisis. Tax assessment and recorded home sales did not appear to have a substantial decline between 2013 and 2016. However, the design of our study is observational and based only on individuals that chose to sell their home or had a tax assessment in these years. Based on the perception that their home price had declined, homeowners may have purposefully kept their homes off the market. This would mean that Zillow data is not representative of the city as a whole.

It is also possible that the city did not perform as many tax assessments in these years to avoid lowering the tax base for the city, or did not sufficiently take the water crisis into account when estimating home values. Another possible source of error comes from the fact that only the most up-to-date record of tax assessment and sold price on Zillow can be extracted. We are not able to follow prices estimates for the same house over time.

Our results also showed very large differences (10,000 - 50,000 dollars) between administrative records (tax assessments and home sale prices) and the ACS home value question averaged at the tract level. Future studies can adopt an experimental research design, where a respondent's records are matched to their tax assessment or last sale price, to better estimate measurement error associated with the ACS home value question.

Finally, we provide visualizations showing perceived home value changes in Flint by census tract using ACS data. We observed that home values decreased more in the southeast and downtown Flint area. We did not find a strong correlation between lead coverage per tract and changes in home prices between 2013 and 2016.