Philadelphia_Property_Tax_Data

Joy Payton

12/3/2016

```
knitr::opts_chunk$set(echo = TRUE, cache = TRUE, message = FALSE)
```

Load Libraries

```
library(dplyr)
library(jsonlite)
```

Obtain Data

We can get property tax delinquency data from Philadelphia's Open Data site:

```
csv <- "https://data.phila.gov/api/views/4v5p-qi2q/rows.csv?accessType=DOWNLOAD&bom=true"
taxDataWhole <- read.csv(csv, stringsAsFactors = FALSE)
taxData <- taxDataWhole %>% select(Parcel.Number, Location,
                                   Owner, Tax.Period, Total, Coordinates)
```

We'll also get property assessments from the Office of Property Assessments, which can tell us more about properties – most helpfully, their census tracts!

```
csv <- "https://data.phila.gov/api/views/3h6i-2hfj/rows.csv"
assessmentDataWhole <- read.csv(csv, stringsAsFactors = FALSE)
assessmentData <- assessmentDataWhole %>% select(Parcel.Number, Location,
                                                 Owner.1, Owner.2, Census.Tract,
                                                 Zip.Code, Geographic.Ward,
                                                 Market.Value, Exterior.Condition,
                                                 Coordinates)
```

Finally, we can obtain a geoJSON file reflecting the Census tracts in Philadelphia:

```
tractData <- fromJSON("http://data.phl.opendata.arcgis.com/datasets/8bc0786524a4486bb3cf0f9862ad0fbf_0.g
```

Let's combine property data to get some enriched data about delinquent properties. We have duplicated data that appears in both data frames (like street address), which works in our favor for making sure that we're matching correctly.

```
combined <- merge(x=taxData, y=assessmentData, all.x=TRUE,
                  by=c("Parcel.Number","Coordinates", "Location"))
```

Do we have census tracts on everyone? Almost certainly not, given that our assessment data had fewer rows than our tax data.

```
table(is.na(combined$Census.Tract))
```

```
##
##  FALSE   TRUE
## 506893 216803
```

We've got a lot of NA's. Let's fix that. We can imagine that a lot of the latitude and longitude coordinates probably overlap between distinct parcels (for example, condos in the same building, or next door neighbors).

So let's make a table of just the coordinates and the census tract, and then use that to populate coordinate matches!

We'll split out the complete and incomplete rows and then re-bind them.

```
completeRows <- combined[!is.na(combined$Census.Tract),]
incompleteRows <- combined[is.na(combined$Census.Tract),]
incompleteRows$Census.Tract <- NULL
tractsIKnow <- unique(completeRows[completeRows$Coordinates !="",
                                   c("Coordinates","Census.Tract")])
additionalCompleteRows <- merge(x=incompleteRows, y=tractsIKnow,
                                by="Coordinates", all.x=TRUE)
```

How many new properties did we find tracts for?

```
table(is.na(additionalCompleteRows$Census.Tract))
```

```
##
##  FALSE   TRUE
##   1881 214924
```

Only around four thousand – we still have 38K properties for which we can't find tracts! But let's add the ones we found to our completeRows and rebuild our "incompleteRows" with the ones we still don't have a tract for.

```
completeRows <- bind_rows(completeRows,
          additionalCompleteRows[!is.na(additionalCompleteRows$Census.Tract),])
```

```
incompleteRows <- additionalCompleteRows[is.na(additionalCompleteRows$Census.Tract),]
```

It turns out that a lot of our incomplete rows are lacking both the coordinates and the location – there will be no way to locate those properties in a census tract, at least not with our current data! Let's remove those entirely.

```
incompleteRows <- incompleteRows %>% filter(Coordinates!="" | Location != "")
```

That gets us down to 31K properties for which we do have some kind of location, either an address and/or some geographic coordinates, but we don't have a tract.
Let's see if we can get down to just the unique coordinates in this group.

```
coordinatesToGeocode <- as.data.frame(unique(incompleteRows$Coordinates))
names(coordinatesToGeocode) <- c("Coordinates")
```

OK! That's less than 4k, so let's use an API and find out the census tract for each of those coordinates. First we'll have to split out the latitude and longitude:

```
library(stringr)
coordinatesToGeocode$lat <- as.numeric(str_match(coordinatesToGeocode$Coordinates,
                                                 "\\(\\d+.\\d+),")[,2])
coordinatesToGeocode$long <- as.numeric(str_match(coordinatesToGeocode$Coordinates,
                                                  ",\\s+(.+)\\)")[,2])
```

Now let's create a function that will use an FCC API to get the census tract for a given latitude and longitude.

```
getTract <- function(latitude, longitude)
{
  tract <- vector(length = length(latitude))
  for (i in 1:length(latitude)) {
    lookup <- paste("http://data.fcc.gov/api/block/find?format=json&latitude=",
                    latitude[i], "&longitude=", longitude[i],
```

```
                    sep="")
    response <- fromJSON(txt=lookup)
    tract[i] <- response$Block$FIPS
  }
  return(tract)
}
```

As an aside, census tracts have state and county prefixes associated with them. The Commonwealth of Pennsylvania has a code of 42, and Philadelphia County has a code of 101. So we'll need to adjust our tracts, since the ones we have so far have been the "short" form, assuming that we know we're in Philadelphia, whereas the ones we're about to get from the FCC are full length.

This next bit takes awhile! We can do it in batches of 1000 to give a tiny bit of throttling.

```
coordinatesToGeocode$tract<-NA
coordinatesToGeocode$tract[1:1000] <-
  getTract(coordinatesToGeocode$lat[1:1000],
           coordinatesToGeocode$long[1:1000])
coordinatesToGeocode$tract[1001:2000] <-
  getTract(coordinatesToGeocode$lat[1001:2000],
           coordinatesToGeocode$long[1001:2000])
coordinatesToGeocode$tract[2001:3000] <-
  getTract(coordinatesToGeocode$lat[2001:3000],
           coordinatesToGeocode$long[2001:3000])
coordinatesToGeocode$tract[3001:length(coordinatesToGeocode$tract)] <-
  getTract(coordinatesToGeocode$lat[3001:length(coordinatesToGeocode$tract)],
           coordinatesToGeocode$long[3001:length(coordinatesToGeocode$tract)])
```