

|  |
| --- |
| **NYC Commuter Data** |
| Matt Beneventi, Courtney Buchanan, Tom Shaheen, Emily Coffield |
| Socioeconomic Commuting Patterns of NYC population based off of zip codes illustrating commuter patterns, commute distance, and income stratification. |
|  |

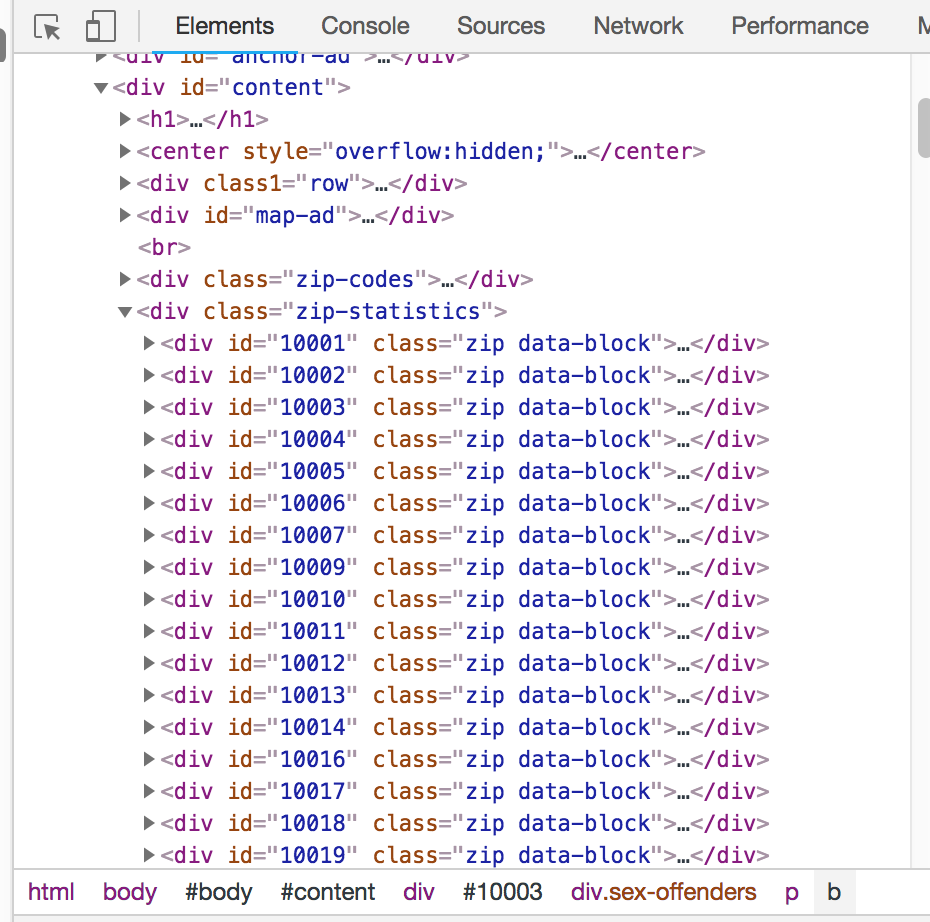




|  |  |  |
| --- | --- | --- |
| Commuting Patterns of NYC |  |  |
| **Resources**  * **City of NYC**   [Housing Data](https://data.cityofnewyork.us/Housing-Development/Housing-New-York-Units-by-Building/hg8x-zxpr)  2011  [ZIP Code to ZCTA Crosswalk for NYC](https://opendata.cityofnewyork.us/)  2010   * **Census.gov**   [Earnings in the past 12 months (in 2011 inflation adjusted dollars)](https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_11_5YR_S2001&prodType=table)  [Means of transportation to work by selected characteristics](https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_17_5YR_S0802&prodType=table)   * **City-Data.com**   [Median Income](http://www.city-data.com/zipmaps/New-York-New-York.html)  2016   * **WNYC**   [Commuter Time](https://project.wnyc.org/transit-time/#40.72293,-73.95481,12,748)  2011 |
| ETL Database Assignment |
| Extract First, we thought about the kind of data we wanted. Instead of thinking about topics we were interested in and going from there, we thought about what kinds of data sets could be joined by using a shared primary key. We realized that most data that would have a strong primary key would often be something like a social security number, or a customer ID. However, with respect to public data, finding a primary key that was reliable on this level wasn’t as easy. We settled on using zip codes in an effort to use a data point that would be unique but also be relatively common in public datasets.  As we researched through different data sets on Kaggle.com, data.world, and American Community Survey, we identified lots of robust and interesting data provided by government and nonprofit entities related to NYC. So, we decided to look more into NYC related datasets. We ended up settling on a combination of non-profit, federal, and local datasets.  Some complications arose around zip code-based data. We found that many researchers preferred to use a “fixed” zip-code called a “ZCTA” - A data point that is often similar to postal zip codes, but instead represent a generalized area representation of the U.S. zip codes. The purpose of a “ZCTA” is to resolve the issue of zip codes being changed regularly, which can make correlating census data problematic if not impossible. Thus, the reliability of our primary key was called into question. However, after doing some research, we discovered that the zip codes in NYC don’t significantly differ significantly from regular zip codes.  One other issue we ran into was finding data for the same year. Our most interesting data set from WNYC was from 2011, so this forced us to match our other data sets to this year.  All of the government level and nonprofit datasets were available in CSV form and many were available with either a zip code and/or ZCTA, so they were easy to download. From there, we were able to come up with ideas on how the data could be merged and manipulated. However, we also wanted to work with data obtained from another resource, so we also utilized a more complicated web scraping application on city-data.com. However, this data was from 2016, which wasn’t ideal but was the best we could do. Transform You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly.  To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template. Load On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look. |

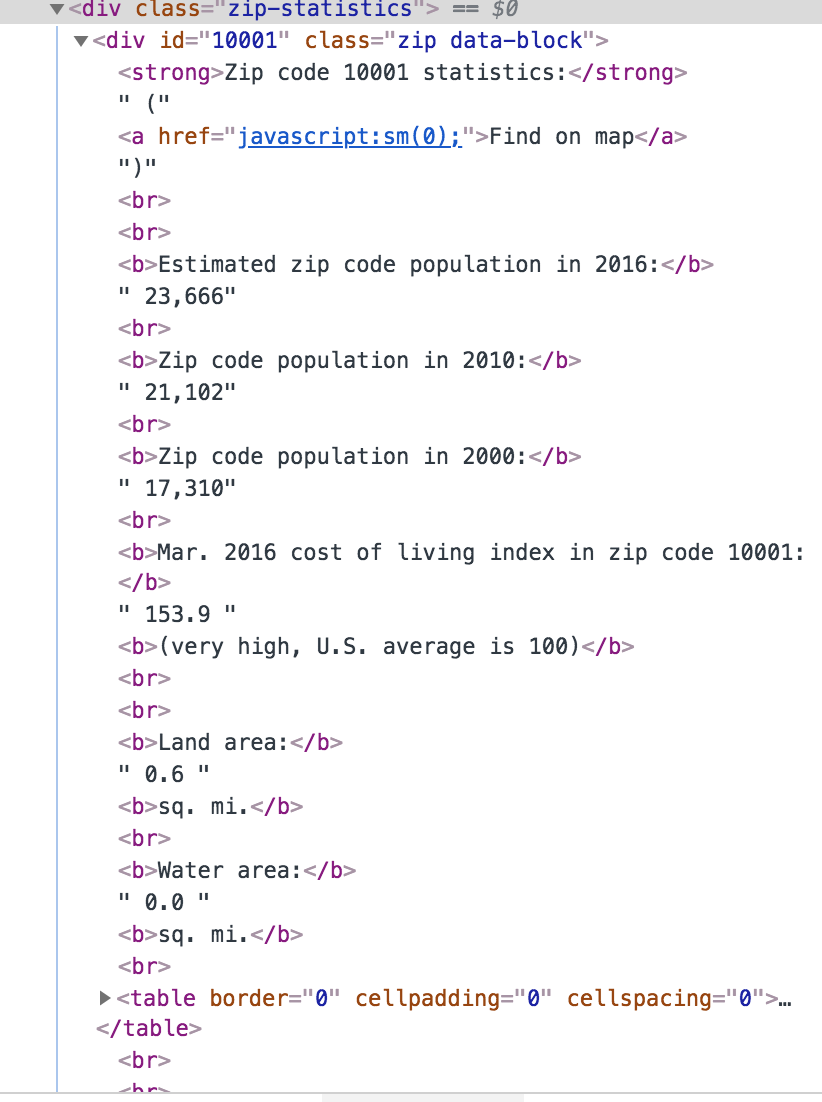
## Transform: Web Scraping

In order to scrape city-data.com, we first had to inspect the code in the browser to ascertain how we needed to accommodate our code to properly pull data from their website. We discovered that the data was buried deep within various div classes in a effort to dissuade web scraping.



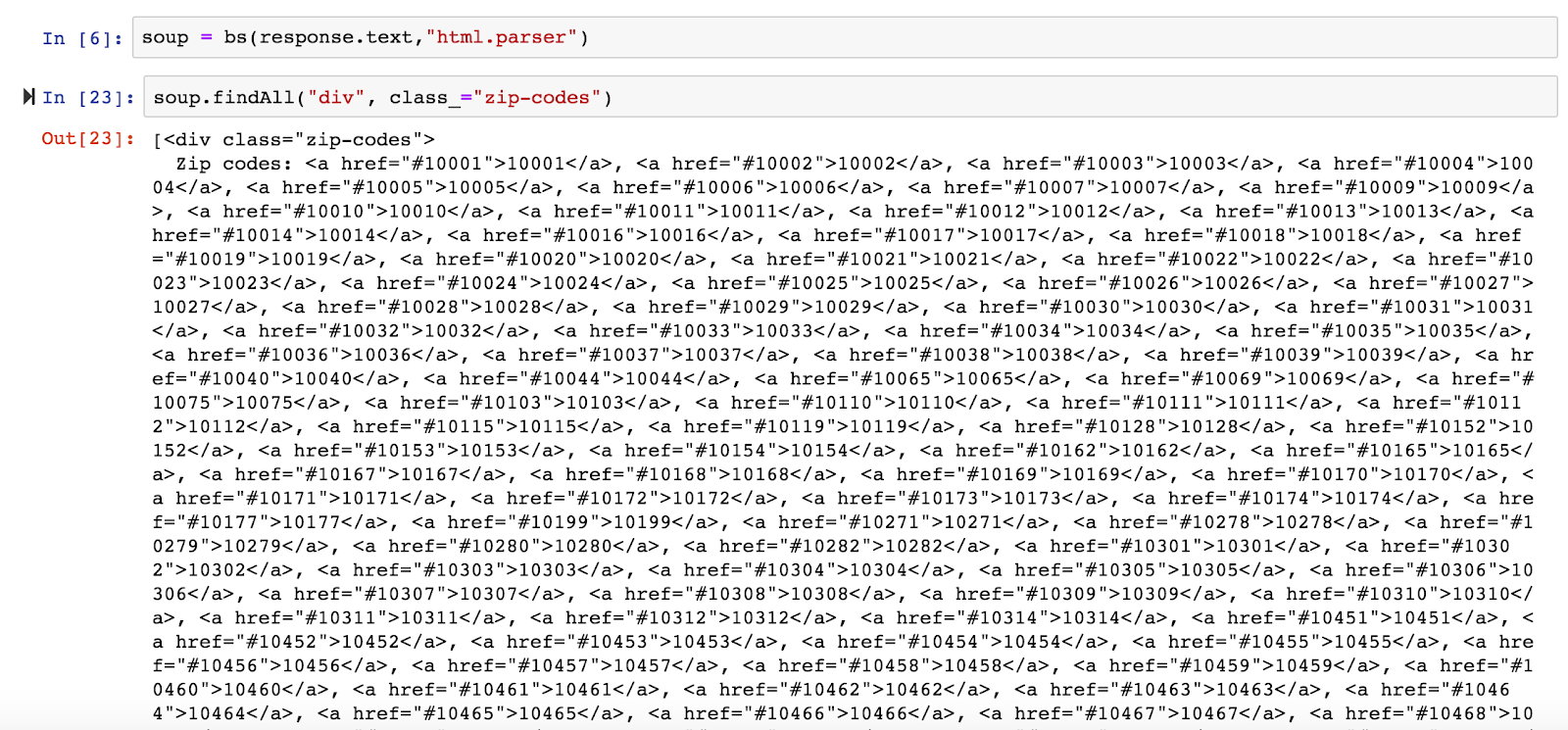
However, we did manage to find data on each zip code buried in the div id’s.







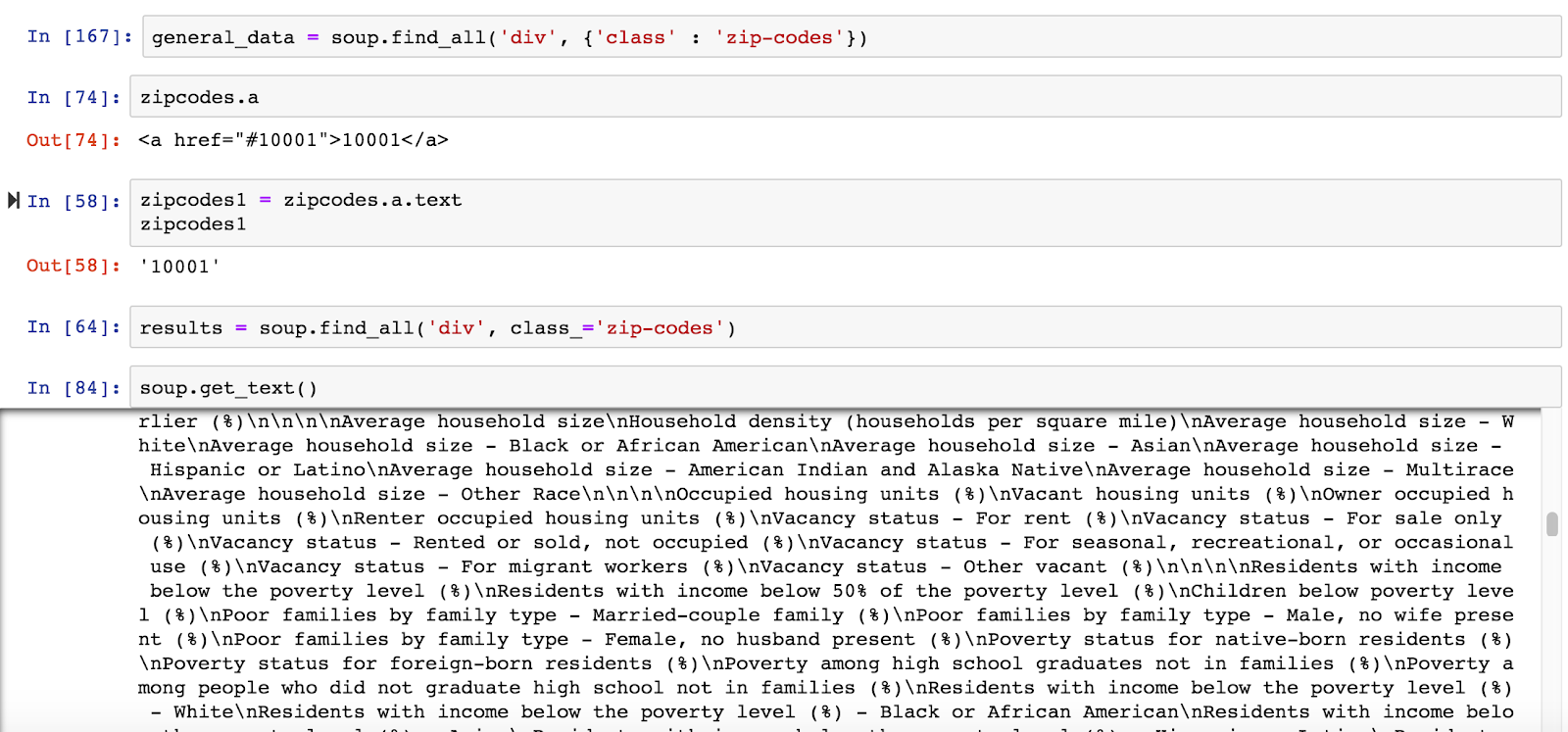
Next, we used beautifulsoup, splinter, and pandas to attempt to scrape the website. Our goal was to attempt to parse the data into a usable format. We succeeded at being able to pull zip codes, but we still had to figure out how to get to the meat of the data that was buried behind the zipcodes.



When we remembered all the data we wanted was sitting behind a div id, we attempted to ping the id which appeared to yield a large amount of data. It seemed like the data was hidden behind links and styling that we could not penetrate easily.



In an effort to circumvent the issues with the div id, we attempted a class scrape and hoped this would yield an easier way to parse the data.

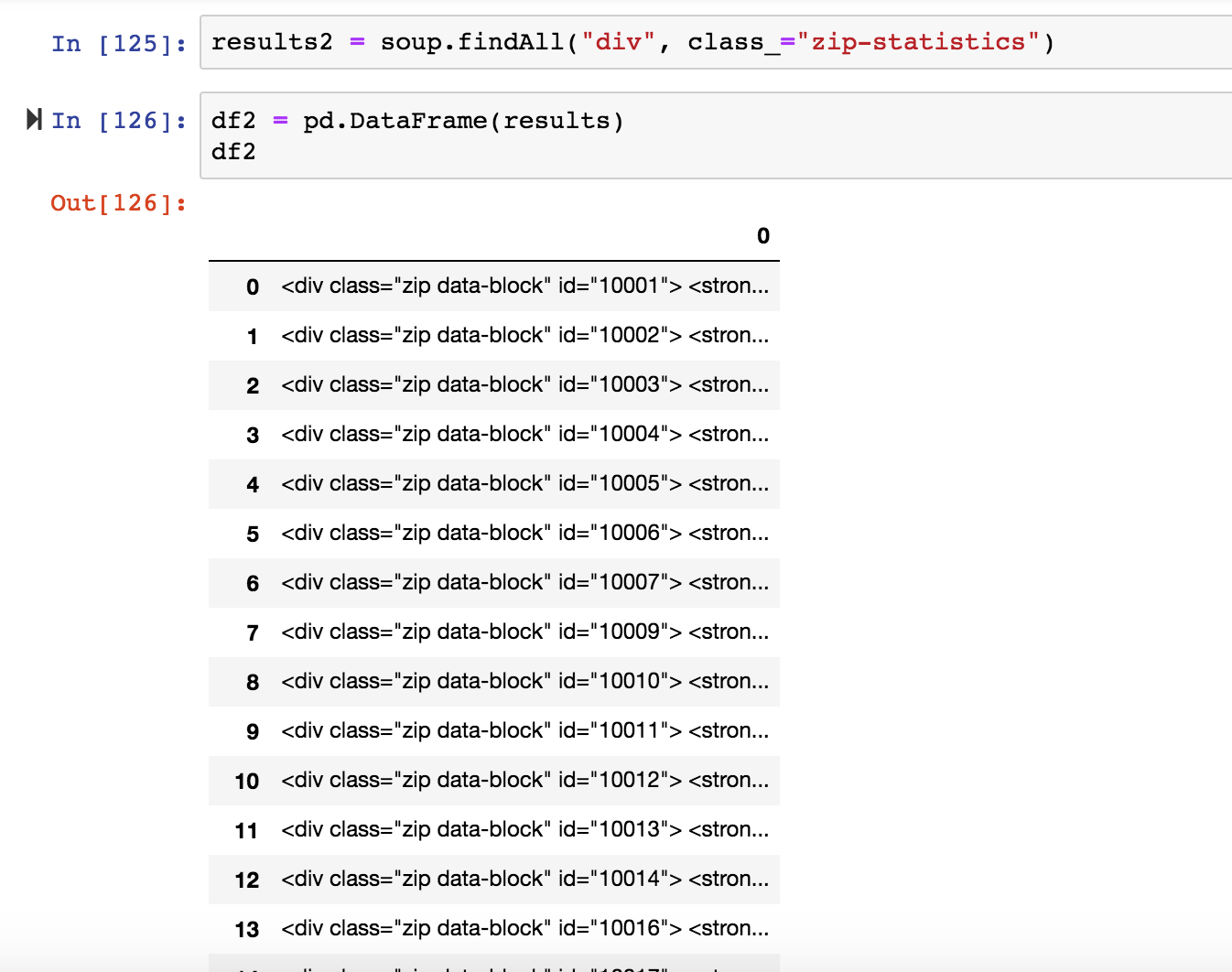


The data needed to be more readable, so we tried to prettify which didn’t do much.



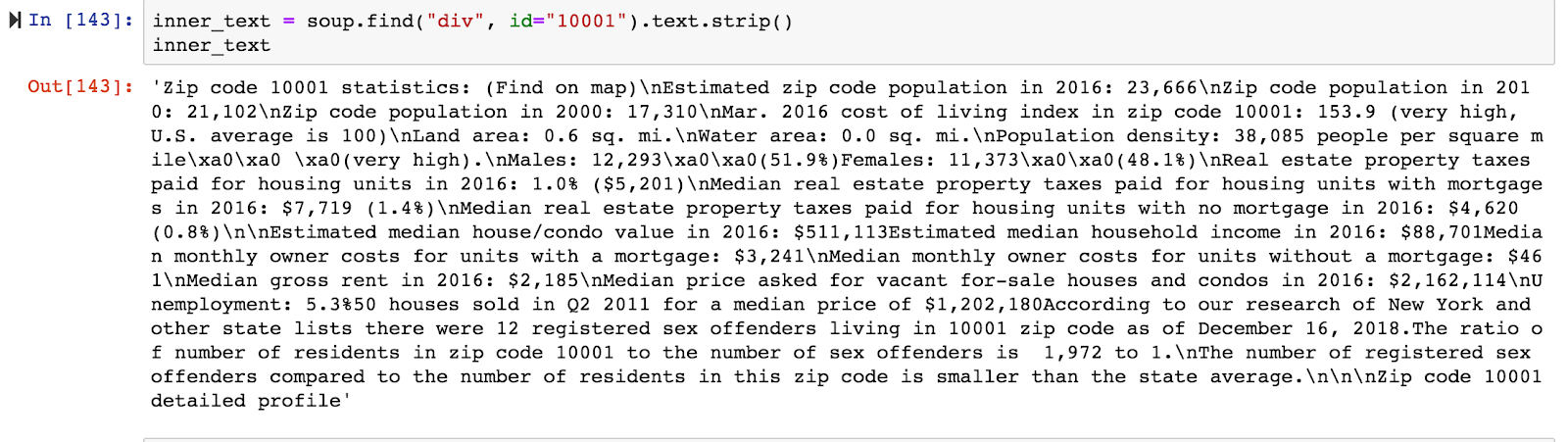


We figured out that if we used the zip-statistics found under the div that we could produce a data frame. When we got this to work, we discovered the data needed to be separated from the links it was embedded in.





We also found that we could get the data by calling each zip code.



Ultimately, we were able to download this data to a csv file and used google sheets to split text to columns.

## Transform: CSV’s

Our next step was to prep the data to load into an SQL database.

One of the main issues we ran into was regarding the sheer granularity of the datasets provided by census.gov. For example, transportation data illustrating most utilized modes of commuter transportation would differentiate between data that excluded taxi’s as a form of public transportation, or whether those driving worked from home or not. Housing data also got granular on race, family size, and age/income bins. Often we would have 80+ columns and over 100,000 rows of data to contend with in each of these instances. The transit CSV alone was so large it never successfully loaded into SQL after running for 3 hours. We had to make some executive decisions about what data to keep and what data to get rid of in an effort to simplify the database.

We found that some of the data had “0” values, so we utilized a simple VBA script that looped through the data and got rid of any rows that added up to a “0” value, sans the header.



We also noticed a lot of the data had extra information from other states, so we needed to figure out the best way to exclude this data. We accomplished this by combining the datasets against a zip codes data set from the City of NY in SQL because it had all the proper NYC zip codes.

NYC Commuter

• • •

We realized at the beginning of the process that we could have saved time if we joined the data sets in such a way so that the smallest amount of data would be combined with the largest. Instead, we combined the biggest data set with the other biggest dataset which then took quite a bit of time to process.

Another issue we ran into was that we noticed the Housing CSV has 3396 rows while only 3386 rows were imported, and we couldn’t figure out why.

## Load

We used SQL import wizard to bring in the data. We then used some queries to clean out erroneous data that was large, cumbersome, and not useful like the previously references extraneous zip codes from other states.

### Queries we used

#### Example of queries used to verify correct number of records were imported (similar query was run for each table):

SELECT COUNT(\*)

FROM transit\_types

#### Queries used to delete records for areas outside of NYC:

DELETE

FROM transit\_types

WHERE NOT zip\_code IN(

SELECT zipcode

FROM nyc\_zipcodes)

DELETE

FROM commute\_times

WHERE NOT zipcode IN(

SELECT zipcode



FROM nyc\_zipcodes)

You can use the flask app to access our database from the internet.

We ultimately chose our datasets mostly because we thought they would create a workable database. However, we did naturally gravitate towards data that we found more interesting. Initially we happened upon the WNYC 2011 dataset illustrating how long it takes a population from various zip codes around the city to travel into work, seeing obvious differences between those who lived in Staten Island versus midtown Manhattan. Then, we thought it would be interesting to gather housing data that parsed out lower/low/middle income housing by zip code as well as median income by zip code, which would illuminate how those in lower income brackets are impacted by longer commutes. Then we found a dataset that showed what form of transportation populations in various zip codes most relied upon for their commute which we also thought would be an interesting data point to see.

