



SignalPOP Attribute Database

DESIGN DOCUMENT

Dave Brown | 7/23/2017

Contents

Overview.....	2
General Model.....	3
Database Structure	5
Locations	6
Attributes.....	6
Sessions.....	7
Values	7
Users	8
UserAttributeLinks	8
Database Access.....	9
Importing Data.....	9
Exporting data.....	11
Appendix A - Data Sources	12
US Census.....	12
DigitalGov.....	12
IRS.....	12
UnitedStatesZipCodes.Org	13

Overview

The SignalPOP Attribute Database is a location based database that contains a growing set of attributes for each location. The overall design allows for new attributes to be added dynamically. Each attribute set is date stamped and associated with a corresponding zip code. As a whole the database provides a general personality for each zip code thus allowing for better product placement based on how well existing (or comparable) products have succeeded at each location

According to <https://www.zip-codes.com/zip-code-statistics.asp>, there are approximately 30,000 general zip codes across the US. Each zip code has an average population of 7,000 residents and will vary in their population density. All zip code data and general attributes associated with them (i.e. population, population density, average age, etc.) is collected from the www.census.gov site using their REST API.

Specialized attributes, such as attributes associated with certain products, or more specialized data not provided by the Census site, are collected using the Facebook REST API.

The basic idea behind the Attribute Database, is to first populate the database with general attributes that give a better feeling of the personality of each location. Next, we collect the popularity of a given product or other target (movie, etc.) in each of the 30,000 zip codes to see where the product (or comparable product) has done well so far. By taking the 3.3% (1,000) top performing zip codes, 3.3% (1,000) of the worst performing zip codes, and 3.3% (1,000) of the most average performing zip codes, we build our initial labeled data set where each zip-code in this 10% is labeled as HIGHLY POPULAR, POPULAR, or NOT POPULAR. The initial model is then trained using a balanced 66.6% of the items for training and a balanced 33.3% for testing. Initially, we only want to train on the top 10% of the zip codes for we first want to bias our network toward these types of zip codes. Once we reach an initial convergence, training is re-started on the entire set of 30,000 zip codes to more finely tune the model.

Once trained, we then run the model on all zip codes to see which zip codes should be popular and compare these against the actual 'popularity' of the zip code. The delta between these two items tells us which zip codes have more opportunity for the target.

General Model

The locations and attributes are stored in tables within SQL. The main locations table stores the key indexes that all other tables reference thus allowing us to build heat maps with any attribute combination that we choose. Our main limitation is set by the size of the heat-map and its granularity. For example a 56x56 heat-map with a 1x1 pixel granularity can store 3,136 different attributes. For a more focused view, we may opt to reduce the granularity to a 2x2 size per attribute which would then have a limitation of 784 attributes per heat-map.

30,000 General Zip Codes with ave population of 7,000 each.
 (<https://www.zip-codes.com/zip-code-statistics.asp>)

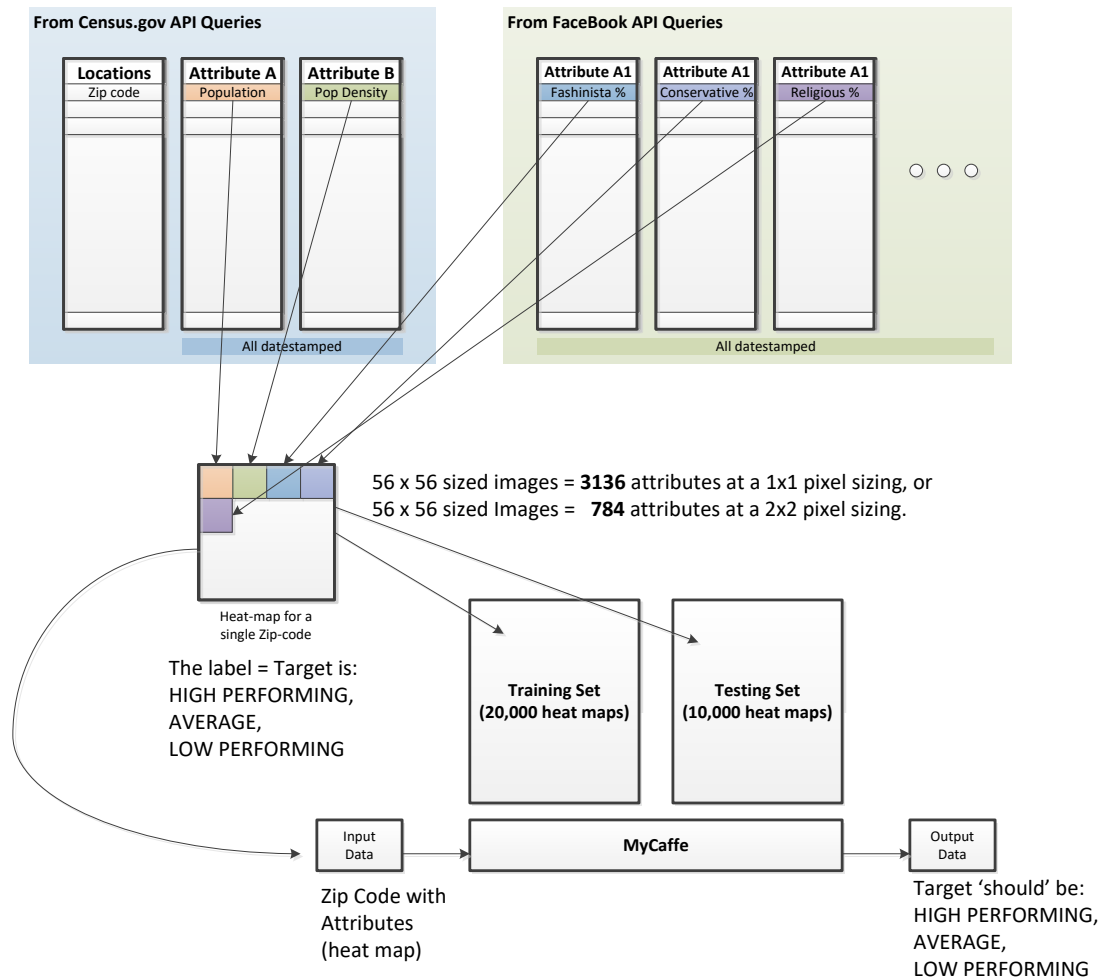
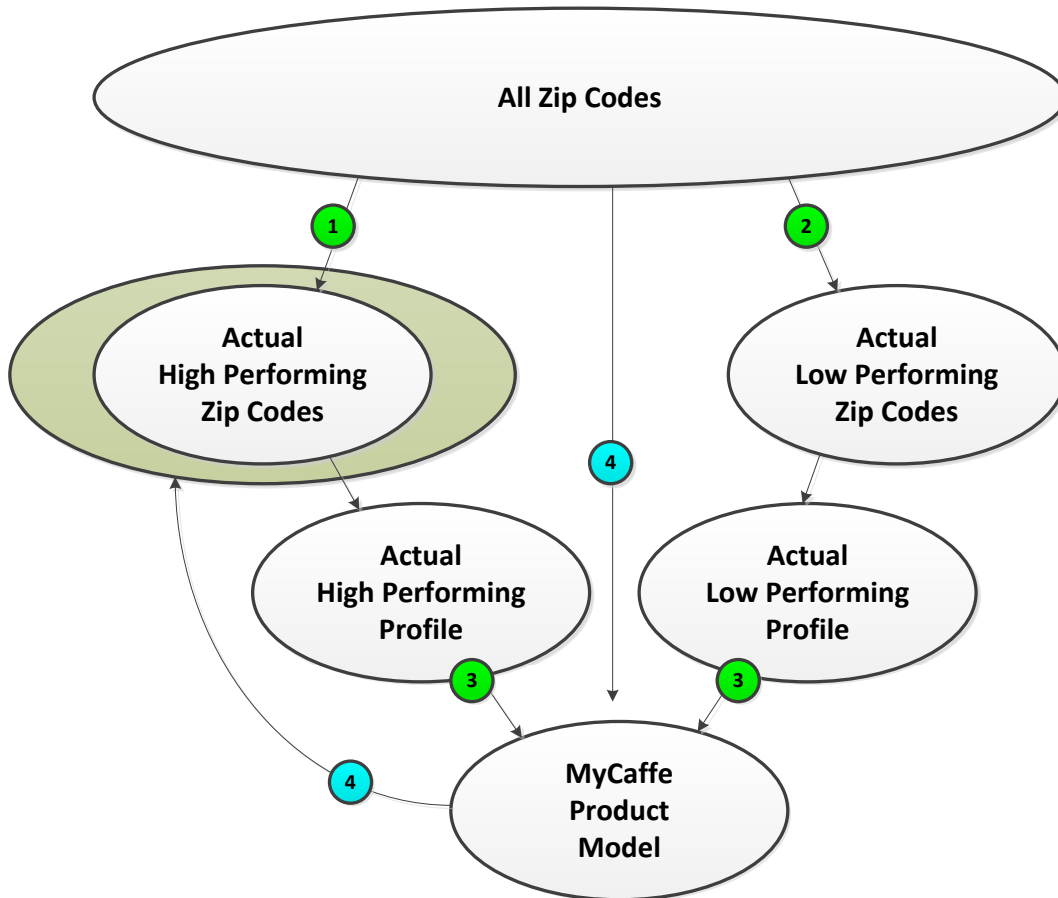


Figure 1 General Model

Once we build our set of heat maps, we must label them so that MyCaffe knows what to learn. The labels are defined by the current performance of the target item that we want to analyze. For example for each zip code, we might count the number of 'likes' per person a given product has and use that as its success in that area. Alternatively, if we have actual

sales figures for a product, we would use the revenue per person as a measure of the product's success in a given area.

Once normalized, we can use these measurements to define a set of categories to learn: HIGH PERFORMING, AVERAGE and LOW PERFORMING. For example the HIGH PERFORMING category might include the 20% of top selling zip codes for a product, whereas the LOW PERFORMING would include the 20% bottom selling zip codes and so on. See steps #1 & 2 below.



Once our data set is labeled (Step #3), we can turn MyCaffe loose to learn it. Once learned, we can then run MyCaffe on all zip codes (Step #4) and record the results which with a degree of probability will tell us all zip codes most likely matching the known top and bottom 20%.

Comparing all zip codes that we detect as high performing (for they match the profile of the other 20% high performing zip codes) but have actual low performing tells us which zip codes would benefit from an increase in marketing and sales.

Database Structure

The attribute database has four main tables: 1.) Locations, 2.) Attributes, 3.) Sessions and 4.) Values.

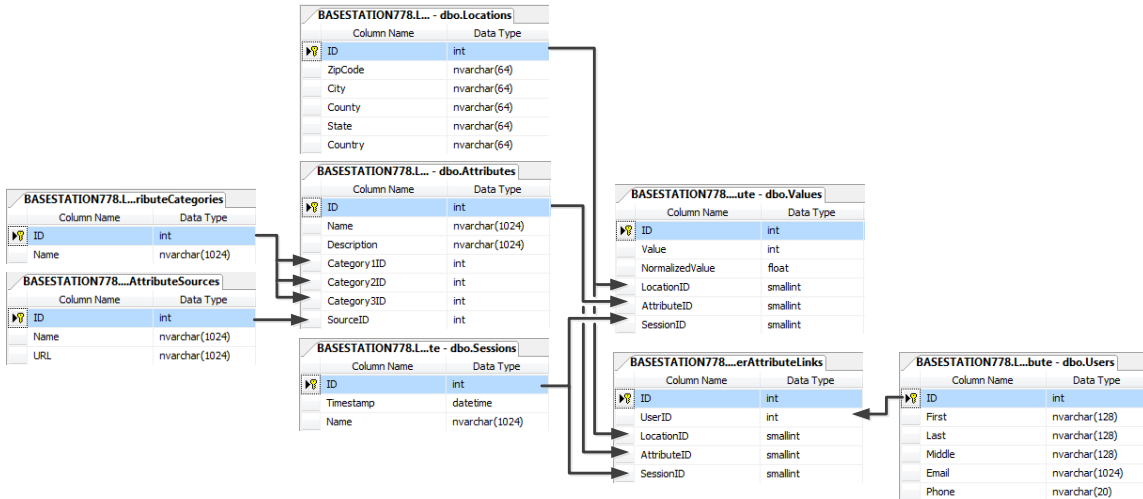
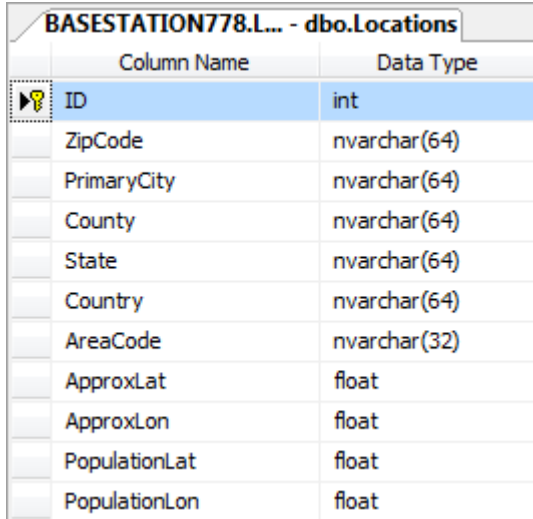


Figure 2 Attribute Database

Together these tables allow for storing a large number of attributes for each location which are all associated with a time-stamped session. Sessions can then be compared to find and analyze the change in attributes over time.

LOCATIONS

The locations table contains the information identifying each location. Typically these locations are associated with a unique postal zip code.

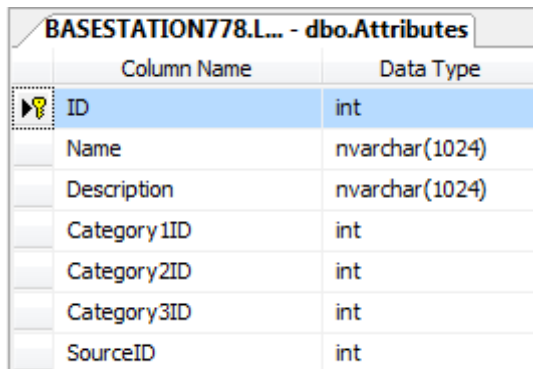


Column Name	Data Type
ID	int
ZipCode	nvarchar(64)
PrimaryCity	nvarchar(64)
County	nvarchar(64)
State	nvarchar(64)
Country	nvarchar(64)
AreaCode	nvarchar(32)
ApproxLat	float
ApproxLon	float
PopulationLat	float
PopulationLon	float

Figure 3 Locations Table

ATTRIBUTES

The attributes table contains information describing each attribute.



Column Name	Data Type
ID	int
Name	nvarchar(1024)
Description	nvarchar(1024)
Category1ID	int
Category2ID	int
Category3ID	int
SourceID	int

Figure 4 Attributes Table

Each attribute has a set of categories that help organize the attribute and a source that describes where the value was collected.

SESSIONS

Each session defines the time-stamp when the data was collected. Data from different sessions can be compared to analyze the rate of change of attributes.

BASESTATION778.L...te - dbo.Sessions		
	Column Name	Data Type
	ID	int
	Timestamp	datetime
	Name	nvarchar(1024)

Figure 5 Sessions Table

VALUES

The values table contains the actual value and a normalized value for the attribute where the normalized value is the value normalized against another attribute, such as population.

BASESTATION778....ute - dbo.Values		
	Column Name	Data Type
	ID	int
	Value	int
	NormalizedValue	float
	LocationID	smallint
	AttributeID	smallint
	SessionID	smallint

Figure 6 Values Table

Each values record takes up 24 bytes of data. A 56x56 sized heat-map stores 3,136 attributes where each heat map is associated with a given location. To perform snapshot differencing we recommend using 3 sessions of heat maps for three different time-stamps (e.g. beginning of year, mid-year, end of year), and 2 deltas.

Bytes per record			18	
Heat-map size	56	56	3136	
Number of Sessions			3	9,408
Number of Differences			2	6,272
Total Number of Records (per zip code)				15,680
Total Number of Zip Codes				30,000
Total Number of Records				470,400,000
Total Number of Bytes				8,467,200,000
Total GB				8.4672

Record Sizing	
int	4
int	4
float	4
smallint	2
smallint	2
smallint	2
Total	18

Together this configuration requires an 8.47 GB database which fits well under the 10.0 GB size limitation of Microsoft SQL Express.

USERS

The users table contains all users associated with locations and attributes for a given session. Unlike the values table which just shows the interest of a given attribute, with the user table we can easily determine 'who' values each attribute.

BASESTATION778.L...bute - dbo.Users	
Column Name	Data Type
ID	int
First	nvarchar(128)
Last	nvarchar(128)
Middle	nvarchar(128)
Email	nvarchar(1024)
Phone	nvarchar(20)

Figure 7 Users Table

USERATTRIBUTELINKS

The user attribute links table links each user to a given attribute at a given location for a given session.

BASESTATION778....erAttributeLinks	
Column Name	Data Type
ID	int
UserID	int
LocationID	smallint
AttributeID	smallint
SessionID	smallint

Figure 8 User Attribute Links Table

NOTE: Using the Users and User Attribute Links tables may expands the database sizing dramatically, for each user may take up to an addition 1k of data per user and 14 bytes per attribute link.

Database Access

Database access is a key element of the overall strategy for our goal is to expand the task of populating the database to all SignalPOP sites and beyond to external vendors with whom may be located anywhere in the world.

With this in mind, we have chosen Azure as the cloud location of the database with a JSON based REST API provided to populate and query the database. This section describes the database access design.

Importing Data

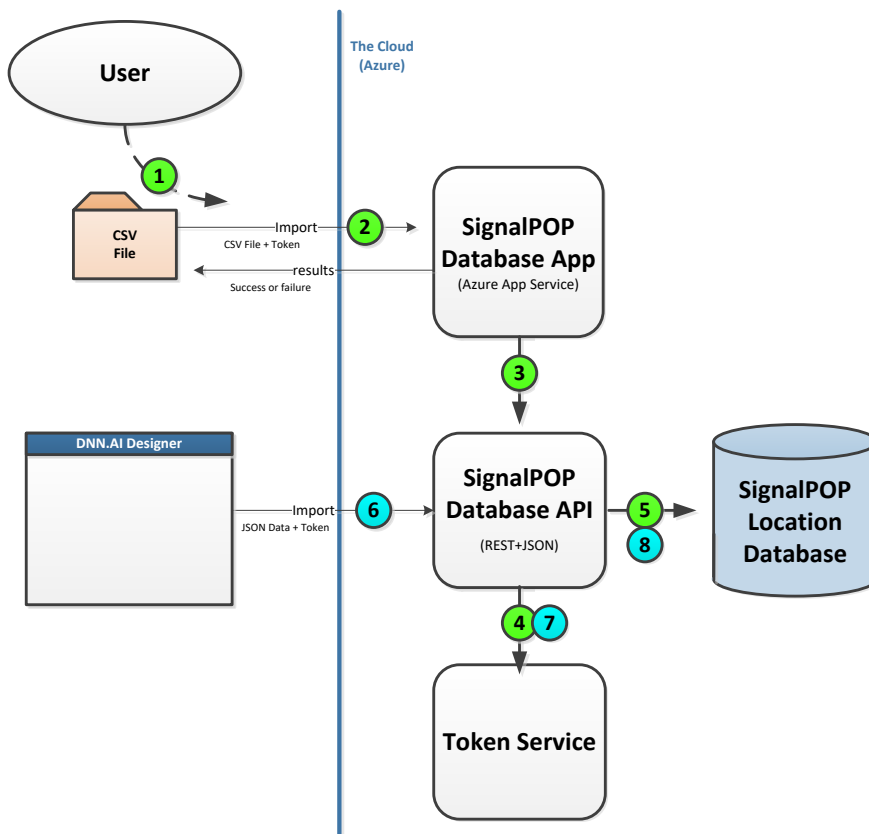


Figure 9 Database Access Model

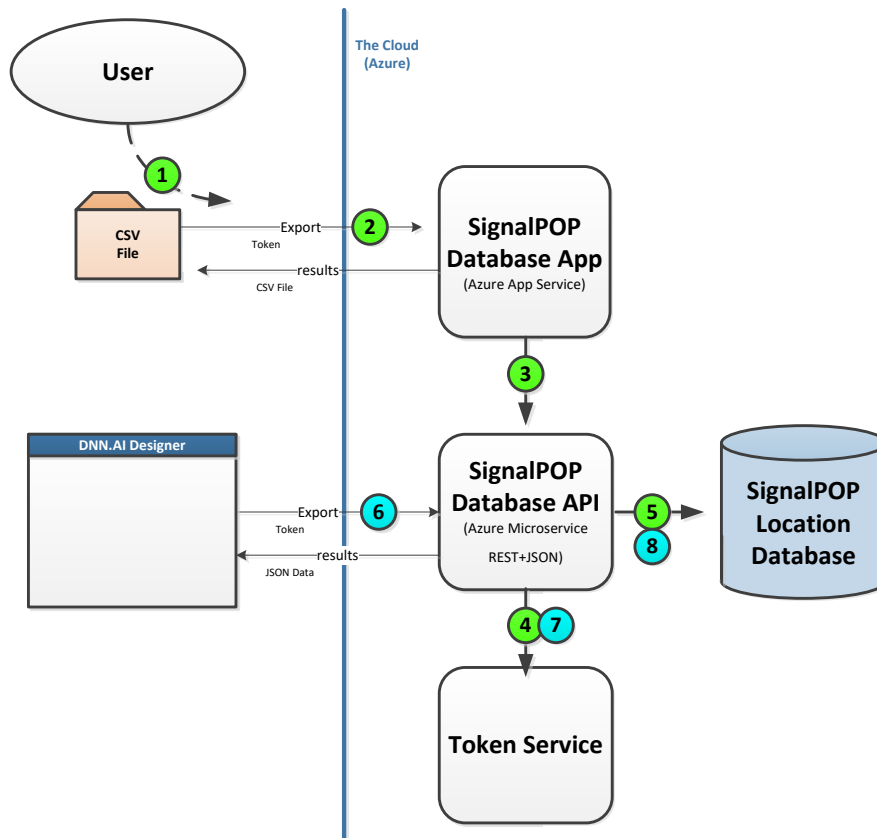
Initially, the SignalPOP Location Database is accessed by users via an Import API and an Export API. Both API require a token that is used to verify the user's access to each API.

When importing, the following steps occur:

- 1.) A standardized CSV file format is used to import data into the database where the user uploads the CSV file along with their access token to the SignalPOP Azure App.

- 2.) Upon receiving the CSV file, the SignalPOP Database App parses the CSV file and passes its data along with the access token to the SignalPOP Database API.
- 3.) The SignalPOP Database API then verifies the token and if valid...
- 4.) ...imports the data into the database (or updates the database when duplicates are found with what is already in the database).
- 5.) Alternatively, the DNN.AI Designer may import data in bulk into the database. When doing so, the DNN.AI Designer uses the SignalPOP Database REST API directly to import data.
- 6.) Upon receiving direct calls to the API, the SignalPOP Database API verifies the token and if valid...
- 7.) ...imports the data into the database (or updates the database when duplicates are found with what is already in the database).

Exporting data



Exporting works in a similar manner to Importing, with the following steps:

- 1.) The user runs the SignalPOP Azure App Service in their browser to export data from the database. For all requests, the user sends their access token.
- 2.) Upon receiving the request, the SignalPOP Database App calls the SignalPOP Database API...
- 3.) Which, upon receiving the query information and token...
- 4.) ...first verifies the validity of the access token...
- 5.) ...and if valid, queries the data requested and returns it to the SignalPOP Azure App, which packages the data into a CSV file and returns it to the user.
- 6.) Alternatively, the DNN.AI Designer may export data via queries by directly calling the SignalPOP Database API.
- 7.) Upon receiving the request, the SignalPOP Database API first verifies the token and if valid...
- 8.) ...and if valid, queries the data requested and returns it to the caller as a JSON package.

NOTE: The token service may be a service implemented by SignalPOP or an external service such as one provided by Selz License Key management.

Appendix A - Data Sources

This section describes several data sources used to collect the data for the Location based attribute database.

US CENSUS

The US Census site forms the basis for our data collection for we are centering the database around zip-codes for they are a granular distribution of population that allows us to then reasonably compare the level of an attribute across regions in a fairly normalized manner. On average Zip codes cover approximately 7,000 residents and when dividing each attribute actual value by the population of zip code we can easily normalize each attribute value so that they can be compares across regions accurately.

Description:	US Census Raw Data
Site:	https://www2.census.gov
Description:	US Census API Data
Site:	https://www.census.gov/developers/
SignalPOP Key:	54df0b9e282c39c78a7568a9dfe18fd1c9671706
Description:	US Census API Examples
Site:	https://api.census.gov/data/2015/acs1/subject/examples.html
Description:	US Census 2010 ZIP Code Tabulation Area (ZCTA) Relationships
Site:	https://www.census.gov/geo/maps-data/data/zcta_rel_download.html
Note:	Contains mapping of zip to county.

DIGITALGOV

Summary: “Data.gov is the central clearinghouse for open data from the United States federal government. It also provides access to many local government and non-federal open data resources. Find out below how federal, federal geospatial, and non-federal data is funneled to Data.gov and how you can get your data federated on Data.gov for greater discoverability and impact.”

Description:	Clearinghouse for open data from the United States Government.
Site:	https://www.digitalgov.gov/
Description:	Datasets
Site:	https://catalog.data.gov/dataset

IRS

The IRS provides numerous datasets based on zip code. For more see the links below.

Description:	Zip code data, all states, and individual income tax data.
Site:	https://www.irs.gov/uac/soi-tax-stats-individual-income-tax-statistics-2014-zip-code-data-soi

UNITEDSTATESZIPCODES.ORG

The United States Zip Codes site contains a wealth of pre-processed zip code data, including the following:

- Population (and population history back to 2009)
- Male/Female counts.
- Race counts (White, African American, Asian, Pacific Islander, Other)
- Age counts (under 10, 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80+)
- Wealth info (poverty, median earnings, median income, etc.).
- Education counts (high school graduates, BS degree, post grad degree)

We have purchased an enterprise license to the database for SignalPOP's use for this site consolidates information from the US Census, Factfinder, IRS and other government sites into one database.

Description: Detailed zip code data.
Site: <https://www.unitedstateszipcodes.org/zip-code-database/>