

Visualization with Excel Tools and Microsoft Azure

Introduction

Power Query and Power Map are add-ins that are available as free downloads from Microsoft to enhance the data access and data visualization capabilities of Microsoft Excel 2013. You can use Power Query to retrieve data from a variety of data sources and integrate that data as part of an Excel data model. In this demonstration, you learn how to work with these tools to analyze data in your Microsoft Azure Storage account.

Prerequisites

Your computer must be running one of the following operating systems:

- Windows Server 2008 R2 (with Microsoft .NET Framework 4.0)
- Windows 7
- Windows 8, 8.1

In addition, you must install one of the following versions of Microsoft Office on your computer:

- Office Professional Plus 2013
- Office 365 ProPlus
- Office 365 Midsize
- Office 365 E3, E4, A3, A4, G3, or G4

Although a 32-bit version is available, you should use a 64-bit computer if you will be analyzing large volumes of data. With a 32-bit computer, you need a minimum of 1 GB of RAM, but a 64-bit computer should have 2 GB of RAM.

Your computer must have Internet connectivity to use Power Map.

Microsoft Azure Storage

You can use a Microsoft Azure Storage account to store data that you want to analyze with Excel. To create a new storage account, open the Microsoft Azure [Management Portal](#), click the **New** button, click **Data Services**, click **Storage**, and then click **Quick Create**. Type in a URL for your storage account, select a location or affinity group, and assign the storage to a Microsoft Azure subscription. Finally, click **Create Storage Account**.

After you create your storage account, you can load data into storage. One option is to use [blob storage](#) to store any type of file. For unstructured tabular data, you can use [table storage](#). You can choose to use one of several different types of tools to work with managing storage, such as Azure Storage Explorer or Azure Management Studio as described later. (You can view a list of other storage explorer tools on the [Microsoft Azure Storage Team blog](#)). Or you can create your own client applications to manage storage programmatically. You can learn more about at [How to Use the Microsoft Azure Blob Storage Service in .NET](#) and [How to Programmatically Access Table Storage](#). Another popular option is using the [Python API](#) to access the Microsoft Azure Storage Services.

Blob Storage and Azure Storage Explorer

An easy way to work with your storage account is to download and install [Azure Storage Explorer](#). When you launch the application, click the **Add Account** button, type in your storage account name and storage account key, as shown in Figure 1, and click Add Storage Account. You can optionally select the **Use HTTPS** check box to use a secure connection for sensitive data. You can locate your storage account key in the Microsoft Azure Management Portal by accessing the Storage page, selecting the storage account, and then click the **Manage Access Keys** button at the bottom of the page. Click the icon to the right of the Primary Access Key to copy it to your clipboard, and then paste it into the Add Storage Account dialog box in Azure Storage Explorer.

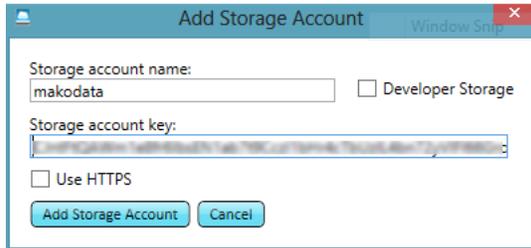


Figure 1: Add Storage Account in Azure Storage Explorer

Next, you need to create a container in your storage account to hold one or more blobs. Each blob is a file of any type. You use block blobs for files up to 200 GB and page blobs for files up to 1 TB. For this example, the data is small enough to fit into a block blob. To create the blob container, click the **New** button in the Container section of the ribbon, type in a name for the container, and click **Create Container**.

Next, you need to access data to put into storage. For this demonstration, download the [Mako Real Actual Sharks](#) file to your computer. Then, in Azure Storage Explorer, select the container you created, click the **Upload** button, and select the CSV file that you just downloaded. When the upload completes, you can view the newly created blob in the selected container, as shown in Figure 2. Now you're ready to use this file in Excel.

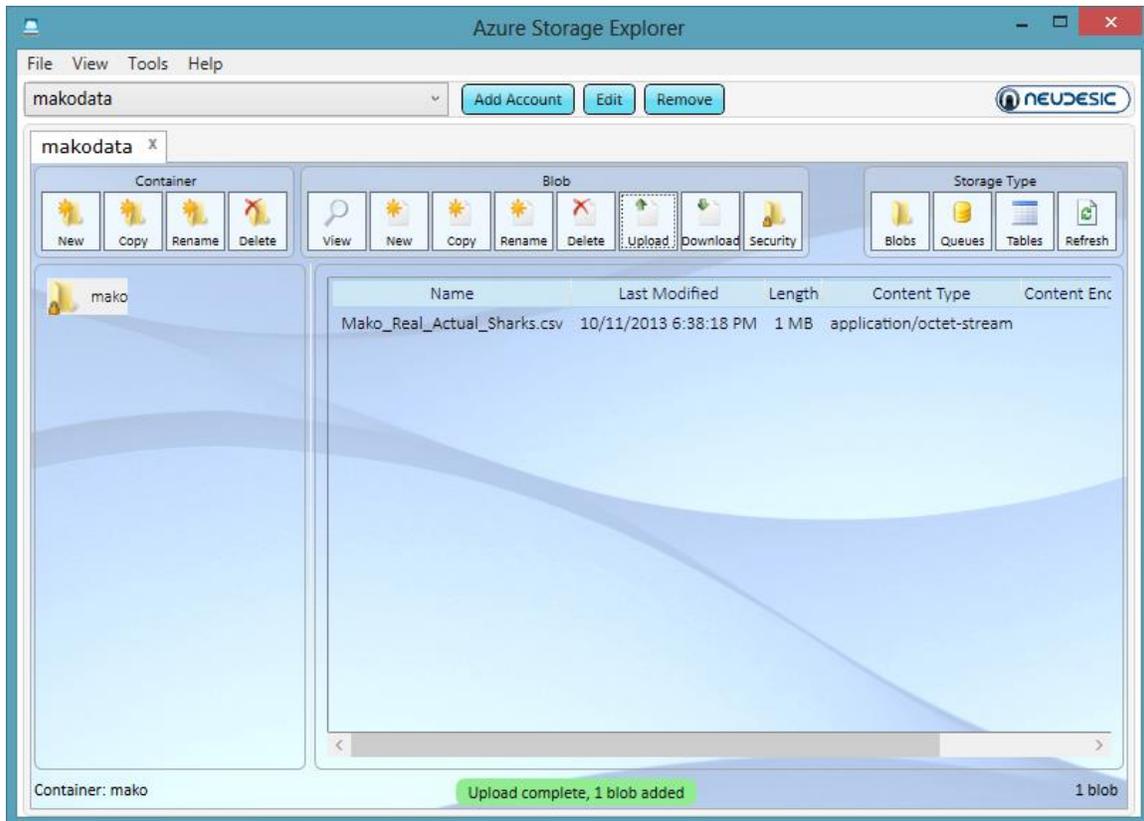


Figure 2: Blob in Microsoft Azure Storage Container

Table Storage and Azure Management Studio

Another option is to use table storage for your data. To manage storage, you can download and install a trial version of [Cerebrata's Azure Management Studio](#). When you open Azure Management Studio, you need to connect to your Microsoft Azure subscriptions. An easy way to do this is to download your publishsettings file from <https://windows.azure.com/download/publishprofile.aspx> and select the subscription file to download. Next, in Azure Management Studio, open the **File** menu, point to **Import**, click **Publish Settings**, and select the publishsettings file to import. Your subscription displays in the Connection Group pane on the left side of the screen. You can open the Storage Account node to access existing storage accounts, or right-click the Storage Account node and select **New Storage Account** if you need to create a storage account.

In the Connection Group pane, expand the storage account that you want to use for table storage, right-click **Tables**, point to **New**, and then choose one of the options shown in Figure 3. For example, you can use the Table from CSV option to upload the [Mako Real Actual Sharks](#) file into your storage account.

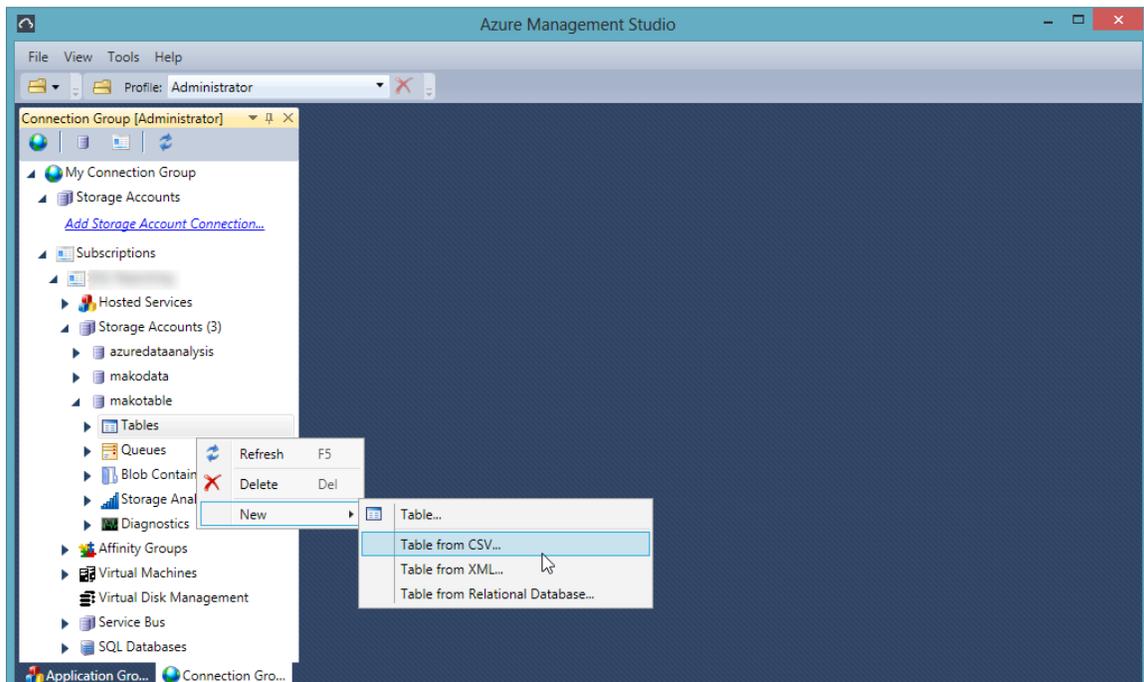


Figure 3: New Table Creation in Azure Management Studio

After you specify the file name to upload and configure any additional properties specific to your data, such as renaming a column name, the upload begins. You can monitor the upload progress in the Transfers pane that displays at the bottom of Azure Management Studio. When the upload is complete, you can view the data in Azure Management Studio, as shown in Figure 4, by expanding the **Tables** node, right-clicking your table, and selecting **Open**. Your data is also now accessible when you use Excel.

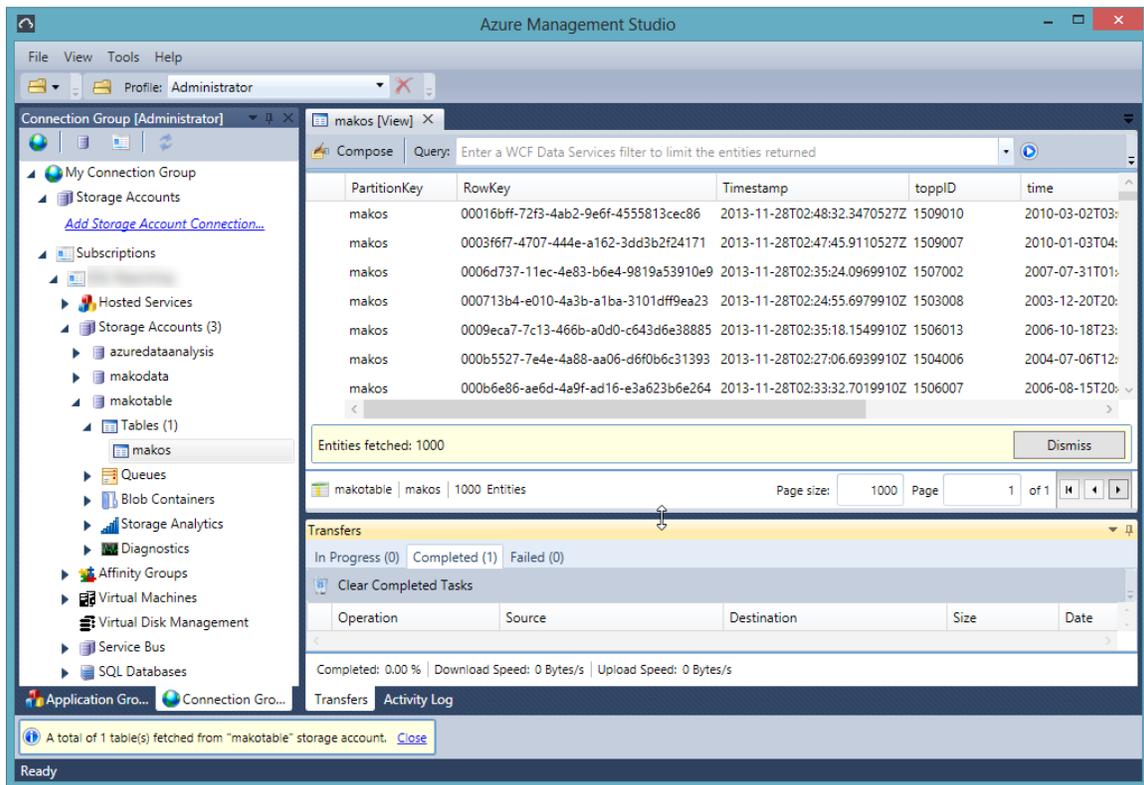


Figure 4: Data in Table Storage Accessible in Azure Management Studio

Installation of Power Map and Power Query

Start by downloading the add-ins from the Microsoft Download Center:

- Power Map: <http://www.microsoft.com/en-us/download/details.aspx?id=38395>
- Power Query: <http://www.microsoft.com/en-us/download/confirmation.aspx?id=39933>

Locate the downloaded files on your computer's file system and double-click each file to start the respective installation wizard. Follow the instructions to complete each wizard. Your computer must be connected to the Internet during installation, because additional files might need to be downloaded by the installer.

Power Query

Power Query is a tool that allows you to find, merge, and manipulate data from many different types of data sources, including both structured and semi-structured data. You can even search for data by entering a keyword. Power Query will suggest websites that contain data related to your query and can scrape the data from a webpage. If you already know where data is located, you can import it from a URL, your network, or your computer. Power Query can import many different file types or retrieve data from a database, a SharePoint list, an OData feed, Hadoop, Microsoft Azure HDInsight, Microsoft Azure Data Marketplace, Microsoft Azure Blob Storage, Microsoft Azure Table Storage, Active Directory, or Facebook. After importing the data, you can manipulate the data in a variety of ways, applying transformations, filtering, and splitting columns, to name just a few operations. If you import multiple data sets that share data, you can combine these datasets into a common table for easy analysis. Think of Power Query as a fast and easy data integration and cleansing tool that you can use within Excel.

Import Data from Blob Storage

In this demonstration, you use Power Query to retrieve data from your Microsoft Azure Storage account and prepare the data for visualization in Power Map. Start by creating a new workbook in Excel 2013. On the Power Query tab of the ribbon, click **From Other Sources** and select **From Microsoft Azure Blob Storage**. Type the name of your storage account and paste in the access key (which you obtain as described earlier in the Blob Storage section of this document).

When you click **Save**, the Query Editor window displays the containers in your storage account. When you click the container, you can see the blobs in the selected container, as shown in Figure 5.

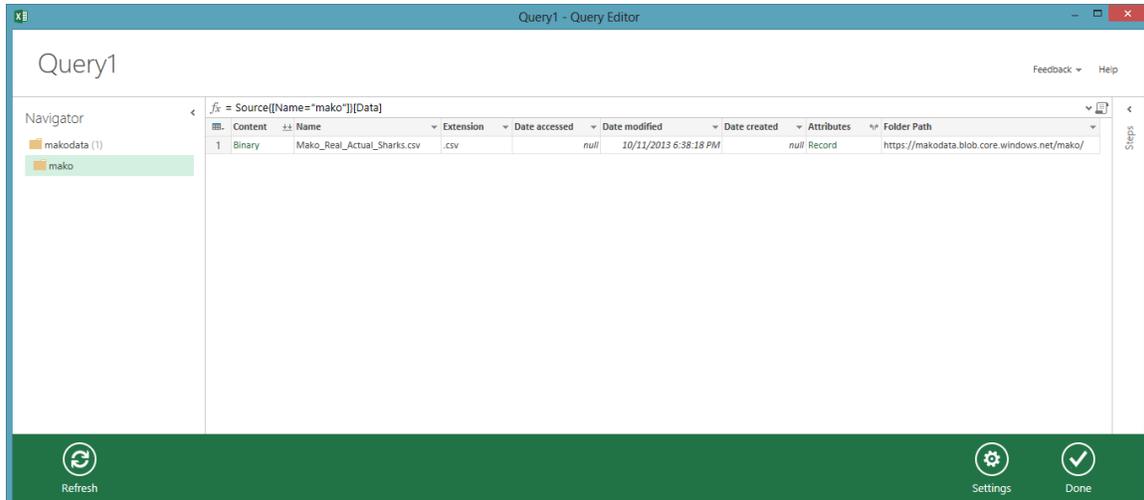


Figure 5: Blob in Query Editor Window

When you click the **Binary** link in the Content column, Power Query imports the data into the Query Editor window where you can view the data. In this case, the data imports as multiple columns, as shown in Figure 6. If necessary, you can transform the data, reshaping it as needed to support your analysis. In this example, no transformations are required before using the data, so click **Done**.

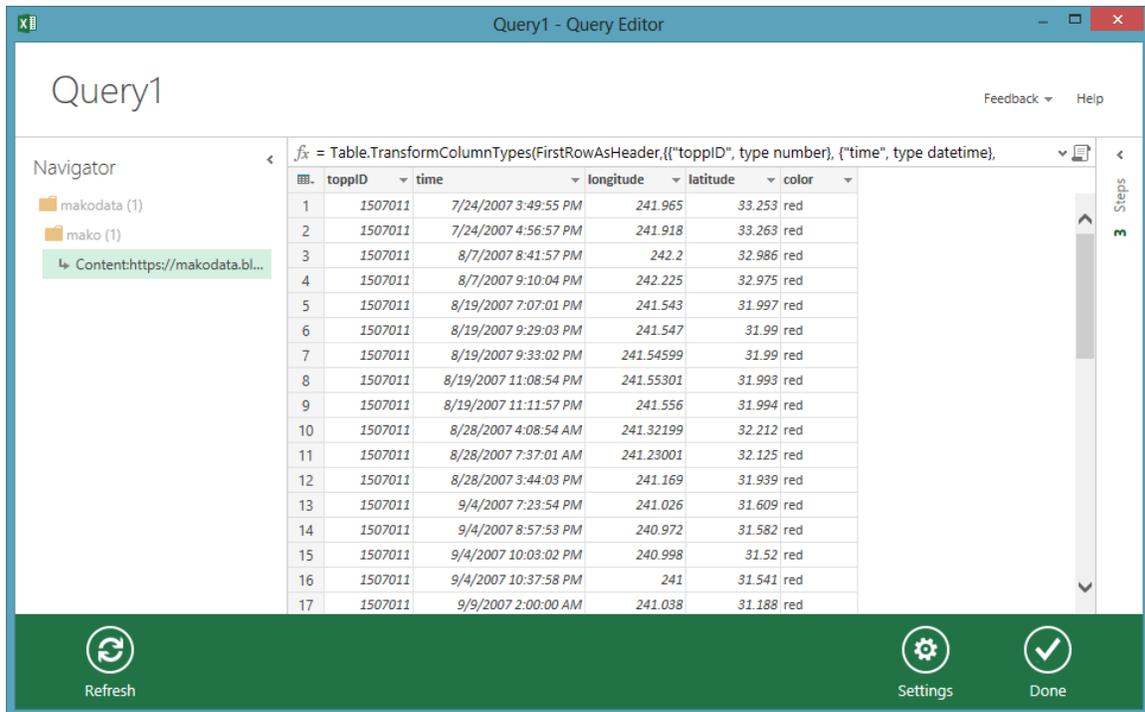


Figure 6: Query Editor Window with Imported Data

Import Data from Table Storage

The process to import data from table storage by using Power Query is slightly different from when you import data from blob storage. In your Excel 2013 workbook, on the Power Query tab of the ribbon, click **From Other Sources** and select **From Microsoft Azure Table Storage**. Type the name of your storage account and paste in the access key (which you obtain as described earlier in the Blob Storage section of this document). When you click **Save**, the Navigator pane opens to display your table. You can select the table and then click the **Load** button at the bottom of the Navigator pane if you want to use all the data without modification. Otherwise, select the table and click the **Edit Query** button in the Navigator pane to open the Query Editor dialog box, as shown in Figure 7.

Visualization with Excel Tools and Microsoft Azure

The screenshot shows the Microsoft Power Query Editor interface. The main area displays a table with 25 rows and 4 columns: PartitionKey, RowKey, Timestamp, and Content. The Content column contains the word 'Record' for every row. The Query Settings pane on the right is open, showing the query name 'Query1' and a list of applied steps, with 'Source' being the only step shown. The formula bar at the top indicates the source is '= Source[[Name="makos"]][Data]'.

Figure 7: Table in Query Editor Window

Click the icon in the Content column and then click **OK** to expand the columns in the table into separate columns, as shown in Figure 8. You can now add additional steps to the query to modify the results to retrieve from the table, such as filtering by a selected color or range of dates. When you close the query editor and keep the results (rather than discard them), the data is downloaded from Microsoft Azure table storage into your workbook.

The screenshot shows the Microsoft Power Query Editor interface after expanding the 'Content' column. The main area displays a table with 25 rows and 6 columns: Timestamp, Content.color, Content.latitude, Content.longitude, Content.time, and Content.topplD. The Content.color column contains various colors like 'white', 'purple', 'orange', 'lightgreen', 'yellow', 'blue', 'purple', 'cyan', 'pink', 'blue', 'green', and 'red'. The Query Settings pane on the right is open, showing the query name 'Query1' and a list of applied steps, with 'Source' and 'Expand Content' being the steps shown. The formula bar at the top indicates the source is '= Table.ExpandRecordColumn(makos, "Content", {"color", "latitude", "longitude", "time", "topplD"},)'.

Figure 8: Expanded Columns for Table Content in Query Editor

To learn more about Power Query, refer to the documentation at <http://office.microsoft.com/en-us/excel-help/microsoft-power-query-for-excel-help-HA104003813.aspx>.

Power Map

Power Map provides a new perspective for your data by plotting geocoded data onto a three-dimensional view of the earth and optionally showing changes to that data over time. To use Power Map, you import raw data into a Microsoft Excel 2013 workbook, add the data to an Excel data model, and enhance the data in the data model if necessary. Once the data model is prepared, you can then insert a Power Map into the workbook and add items from the data model for viewing on the map. You start by identifying the geographical elements in your data model, select the numeric data to display as well as an aggregation function like sum or count, and then you specify a visualization method such as a column chart, bubble map, or heat map. If your data model includes a date data type, you can identify the date as a time element and then use the map's play axis to view location changes to the numeric data over time.

To use the data imported by using Power Query, open the Insert tab of the ribbon, click the **Map** button, and then click **Launch Power Map**. A new window opens after a few seconds. Power Map attempts to detect geocoded data in your data model, as shown in Figure 9, and performs a simple mapping. If necessary, you can change the check box selections for geographical fields in the Task Panel on the right side of the screen. In addition, if Power Map fails to correctly identify the type of data that the selected fields represent, you can assign the field to an applicable type from the drop-down list to the right of the field in the Geography and Map Level section of the Task Panel.

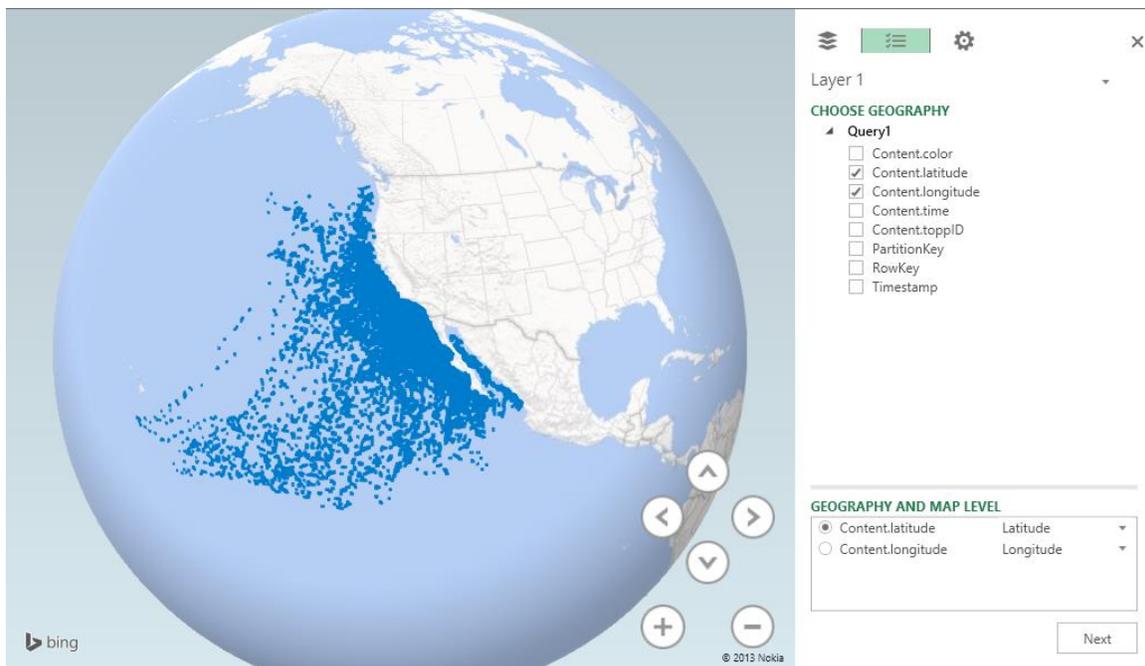


Figure 9: Geography and Map Level Selection

Click the **Next** button in the bottom right corner to continue defining the map properties. You now specify the type of chart to display by choosing one of the following options from the Type drop-down list: Column, Bubble, HeatMap, or Region. For this demonstration, keep the default selection of Column.

Your next step is to specify the field containing the numeric value to aggregate and visualize on the map as a column. In the case of the shark data, there is no numeric value available, but you can select the toplD field, which represents an individual shark, to add it to the Height section of the Task panel. The

default aggregation is Sum, but you can click the arrow icon to the right of topplD and change the aggregation to Count (Distinct).

The map adjusts to show columns representing the count of sharks in each location, but the imported data contains another variable to help us better visualize the shark population. Select the color check box and notice that it now displays as a category. The map now displays columns with multiple colors and a legend lists each color next to the color to which it corresponds on the map, as shown in Figure 10. To close the legend, hover the mouse over the legend, and then click the **X** that displays in the top right corner.

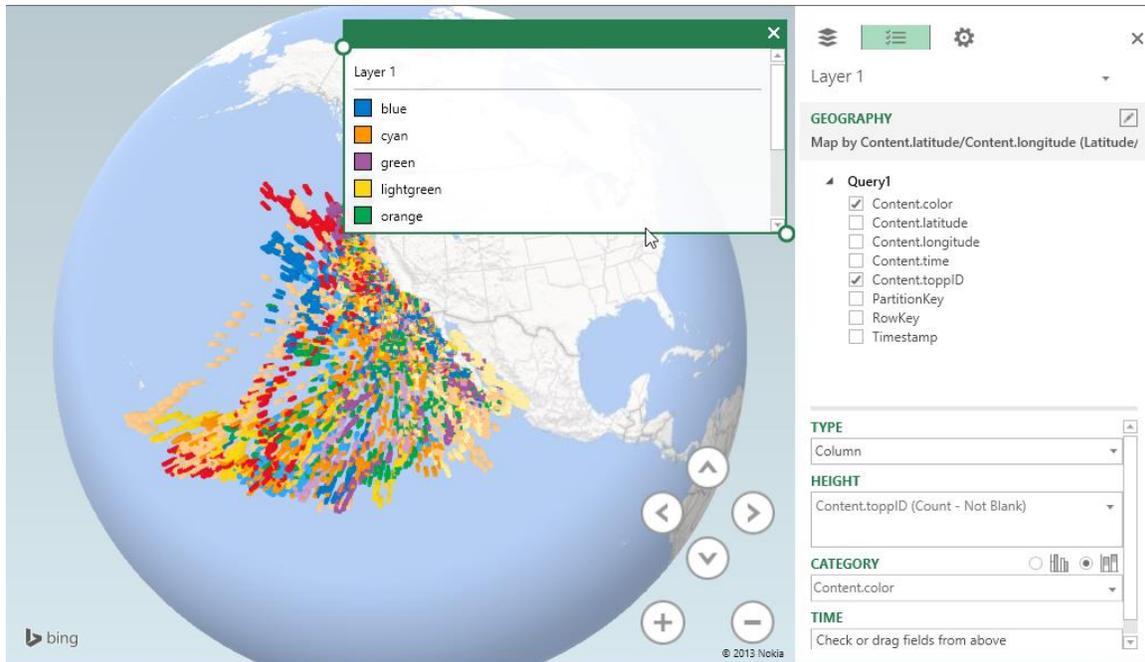


Figure 10: Layer Definition

Note: In this example, the colors in the legend are not consistent with the labels because Power Map understands the color values from the data set as text labels with no meaning rather than specific colors.

The settings you configure here are associated with Layer 1. Click the **Add Layer** button in the toolbar to add a new layer and begin the process again by selecting geographical fields first. You can click the icon in the top left corner of the Task Panel to control which layers are visible. For example, you could obtain data related to sea temperatures by date and include that in the model, allocating the shark movement data to Layer 1 and the temperature data to Layer 2.

You can also visualize changes to the data over time by adding a play axis to the map. To do this, drag **Date** from the model's field list to the **Time** box in the bottom right corner of the window. A play axis displays below the map. When you click the **play** button on the left side of the play axis, the map renders the data points applicable to the current date playing. In addition, you can use the arrow buttons in the bottom right corner of the map to rotate the angle of the view or use the plus or minus buttons to change the zoom factor.

Try changing the Type to a different selection to observe the results.

Power Query and Power Map Using Multiple Datasets

As another example, you can download data from the US Census Bureau and use Power Query to prepare the data for visualization in Power Map. The US Census Bureau has a data set containing [quick facts](#) about states and counties in the US. These facts are identified by a code in one file, [DataSet.txt](#), while the code is described in another file, [DataDict.txt](#). A third file, [FIPS CountyName.txt](#), maps the FIPS code in DataSet.txt to its corresponding county and state name. Power Query makes it easy to download the data from these three files and combine them so that the resulting data can be visualized on a map.

Import DataSet.txt

You start by creating a new workbook in Excel 2013. On the Power Query tab of the ribbon, click **From Web** and type the following URL into the dialog box:

`http://quickfacts.census.gov/qfd/download/DataSet.txt`

Power Query imports the data into a Query Editor window where you can view the data. In this case, the data imports as a single column in which each row is a comma-delimited string, as shown in Figure 11. The Query Editor contains many commands that allow you to reshape the data into a different format.

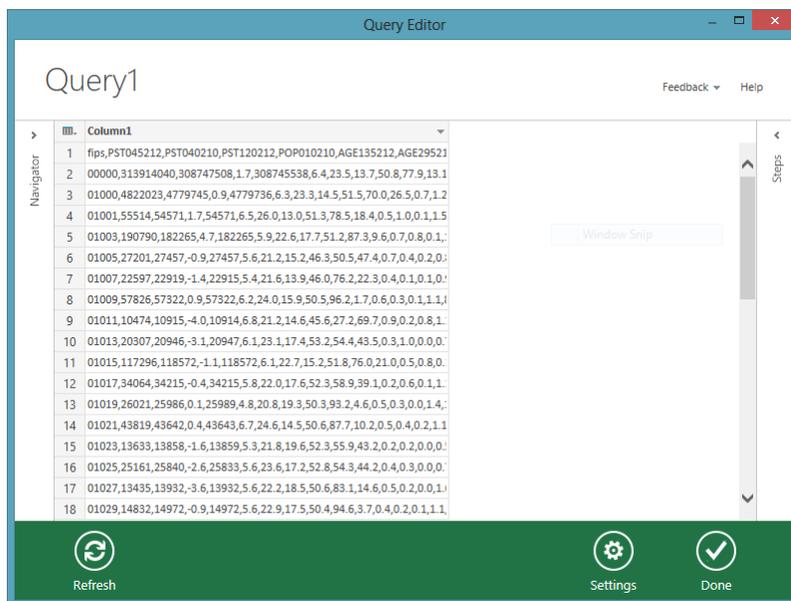


Figure 11: Query Editor Window with Imported Data

Your first goal is to transform this data into a multi-column format by splitting the data at each comma. Expand the **Steps** panel on the right side of the screen. This panel documents each step of the transformation for you. Each step has settings that you access by clicking the gear icon to the right of the step name. If you decide that a particular step did not produce the results you desire, you can click the **Delete** button that displays when you position the cursor over the step name to remove the step, and then try a different action.

Change Source File Type

In this case, the settings for the source show the URL from which you downloaded the file. There is also an Open File As drop-down list, which currently displays Text. Open the dropdown list to select **Csv Document**, as shown in Figure 12. As you can see, there are other file types from which to choose here.

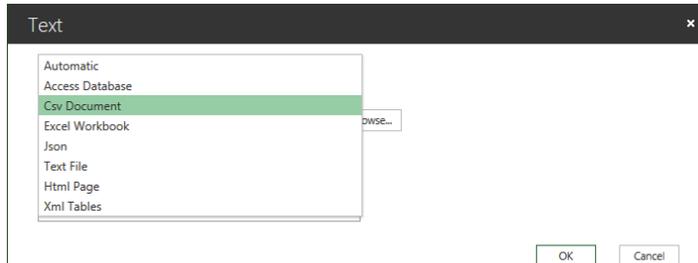


Figure 12: Change Source File Type

Unpivot Columns

After you click **OK**, Power Query refreshes the Query Editor window to display the data in separate columns. Notice the first column is “fips” which you will later use to find the county and state name by using a separate file. Each column is a separate quick fact, but it’s difficult to identify a fact from a code like PST045212. You will also use this code to find a better description in yet another file. First, you need to reshape the data again by converting each column into a separate row. That is, you need to generate one row per fact for each county. To do this, scroll to locate the last column in the Query Editor, right-click that column, and then select **Unpivot Columns**. The unpivoted data appears in the Attribute and Value columns, as shown in Figure 13.

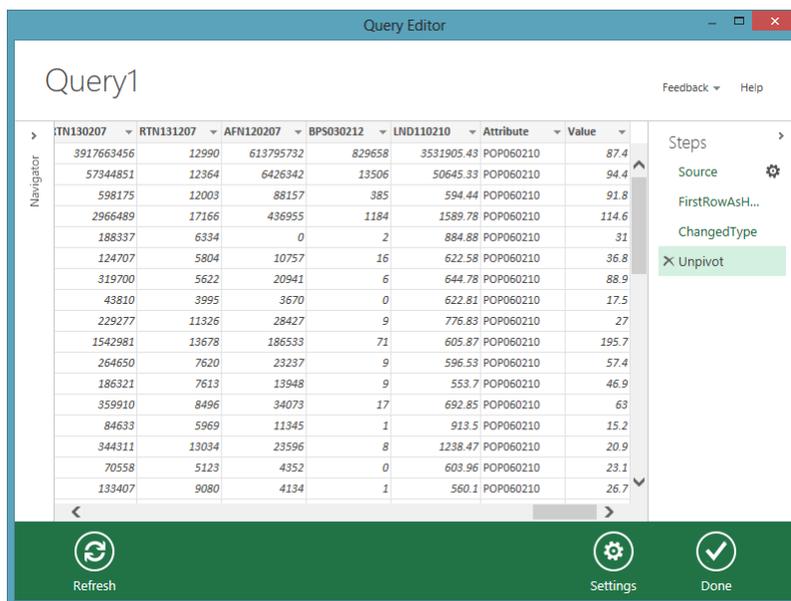


Figure 13: Unpivoted Data

Rename Columns and Query

At any time, you can rename columns to provide a more suitable label. Right-click the **Attribute** column, select **Rename**, and then type a new name, such as *Item Code*. You can also name the query, which is useful when you later want to merge multiple queries. Having a recognizable name will help you locate the correct query. Just click the name, **Query1**, and type a new name, *Dataset*. Click **Done** to add the transformed query results to a sheet in the Excel workbook.

Import DataDict.txt

Now start a new query by clicking the **From Web** button on the Power Query tab of the ribbon, and type the following URL into the dialog box:

`http://quickfacts.census.gov/qfd/download/DataDict.txt`

Split Column by Number of Characters

Again, Power Query imports the data as a single column, but this time the column does not contain comma-delimited strings. It appears that a space separates two types of data, a code and a description, and the code is a fixed width. You can split the data at a specific location to separate it into two columns. To do this, right-click the column header, point to **Split Column**, and then select **By Number Of Characters**. Change Number of Characters to **9**, select the **At The Left Most Delimiter** radio button, and click **OK** to transform the data as shown in Figure 14.

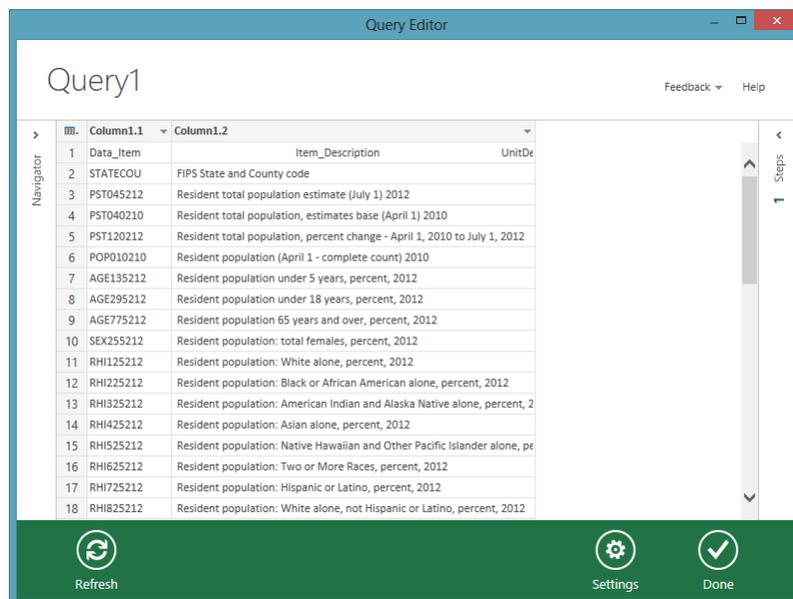


Figure 14: Split Data

Remove a Column

The second column contains some extraneous data that you don't need. The purpose of this query is simply to get the item code and description, so you can split the second column to separate out the description. Right-click the second column, point to **Split Column**, select **By Number Of Characters**,

replace 2 with **105**, select the **At The Left Most Delimiter** radio button, and click **OK**. Right-click the third column, and select **Remove**.

Use First Row as Headers

Next, notice the first row of data contains column headers. You can remove this row from the data and make it header row by right-clicking the first column and selecting **Use First Row As Headers**. Then rename the second column to *Description*, rename the query as *Data Dictionary*, and then click **Done**.

Import FIPS_CountyName.txt

Now click the **From Web** button on the Power Query tab of the ribbon, and type the following URL to retrieve the third and final file:

```
http://quickfacts.census.gov/qfd/download/FIPS_CountyName.txt
```

Split Column by Delimiter

This data contains the FIPS code and county name in a single column. The code is separated from the description by a space that you can use as a delimiter for splitting. Right-click the column, point to **Split Column**, select **By Delimiter**, select **Space** in the dropdown list, select the **At The Left Most Delimiter** radio button, and click **OK**.

The next step is to separate the country from the state by using the comma as a delimiter for splitting. Right-click the second column, point to **Split Column**, select **By Delimiter**, keep the default selection of **Comma**, select the **At The Left Most Delimiter** radio button, and click **OK**. Then rename the first column as *FIPS*, the second column as *County*, the third column as *State*, and the query as *FIPS*.

Text Filter

This data contains rows for the United States, each state, and each county in a state. However, you want to keep only the rows that contain county names. You can eliminate rows by using a filter. Notice that the United States and individual state rows have null values in the State column. To remove these rows, right-click the **State** column in the first row, point to **Text Filters**, and select **Does Not Equal**. The data is now transformed into the proper structure and filtered to include only the county rows, as shown in Figure 15. Click **Done**.

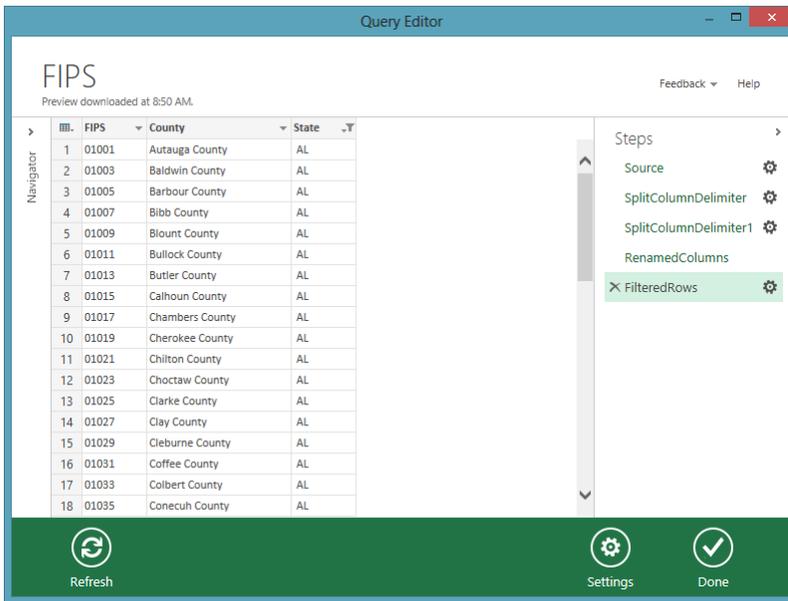


Figure 15: Filtered and Transformed Data

Merge Data

On the **Power Query** tab of the ribbon, click **Merge**. In the first drop-down list, select **Dataset**, and then select **Data Dictionary** in the second drop-down list. The data for each table displays after you select it in the dropdown list. Scroll to the far right of the first table, and click the header for the **Item Code** column to select it. Then select the **Data Item** column. In the **Privacy Levels** dialog box, select **Public** for both files, and click **Save**. Notice the message that displays at the bottom of the Merge dialog box, as shown in Figure 16. This message confirms that the data in each table matches up in this column for all rows.

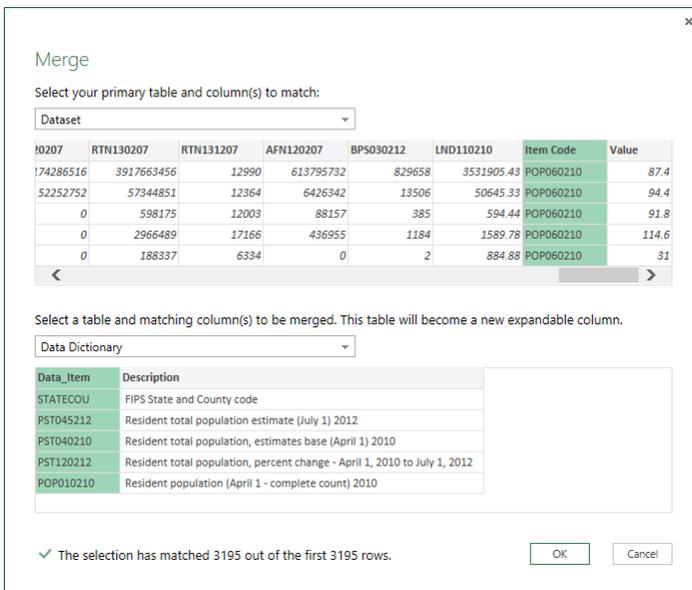


Figure 16: Merging Data

Click **OK** to perform the Merge and display a new Query Editor window. Scroll to the last column, labeled NewColumn, and then click the icon that displays to the right of the column name. Click **(Select All Columns)** to clear the selections, click **Description**, as shown in Figure 17, and then click **OK**. Click **Done** to complete the query for the intermediate table.

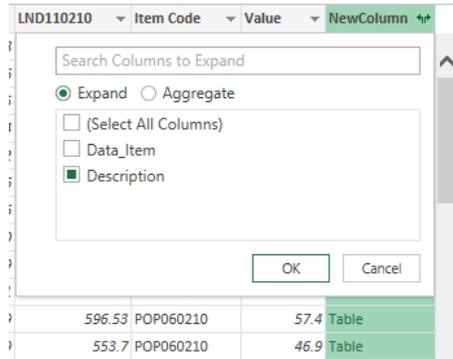


Figure 17: Expand Table After Merge

Now bring in the FIPS code with another merge operation. Click **Merge** on the Query tab of the ribbon, select **FIPS** in the second dropdown list, and then click the FIPS column in both tables. Set the privacy level to **Public** for the FIPS_Count.txt file, and notice that the evaluation of the data fails to find matches because the FIPS data has a leading zero.

Filter and Shape

Even after you have completed a query and created a table, you can always return to the query and continue making modifications. Click **Cancel** in the Merge dialog box and then locate the sheet containing the FIPS table. In the Query Settings panel on the right side of the workbook, click the **Filter and Shape** button. Right-click the **FIPS** column, point to **Split Column**, and select **By Number Of Characters**, change 2 to 1, select **At The Left-Most Delimiter**, and click **OK**. Right-click the first column and select **Remove** to delete that column.

Another problem exists in this data that will prevent a merge. Specifically, the data in the FIPS column is a text data type, but needs to be a number data type to merge successfully. Right-click the **FIPS** columns, point to **Change Type**, and select **Number**. Now the data is ready for merging, so click **Done**.

Next, click **Merge** in the ribbon, select **Merge1** in the second dropdown list, and click the **FIPS** columns in each table. The status will show a warning that some matches could not be found, but you can safely ignore this. Click **OK** to continue, and then click the icon to the right of NewColumn. Click the **(Select All Columns)** check box to clear the selections, and then select Item Code, Value, and NewColumn.Description. Click **OK** to see the expanded results.

Although they are not currently visible, there are nulls in the description column in some rows that you must filter out. Click the arrow icon to the right of the NewColumn.Description label and click **Sort Ascending**. Right-click the first row in the NewColumn.Description column, point to **Text Filters**, and select **Does Not Equal**. Remove the FIPS column and NewColumn.Item Code columns, and then rename the last two columns as *Value*, and *Description* respectively.

Visualizing Data

The last step is to select a description for visualization. Click the arrow icon to the right of Description, click **(Select All)** to clear the description, and then select **Population Per Square Mile, 2010**. Click **OK**, and then click **Done**. On the Insert tab of the ribbon, click **Map**, and then click **Launch Power Map**. The geography fields County and State are automatically detected and selected. In the bottom right corner, you can choose the level of detail to map, county or state. Keep the county selected, and click **Next**.

Select **Value** in the list of fields to show the population as columns on the map. You can clearly see which areas of the country have a high population density by the size of the columns. Close the layer legend to remove it from the view, and then use the arrow keys and zoom keys on the map to view the data in more detail, as shown in Figure 18. You can also drag the globe in the map to change the perspective.

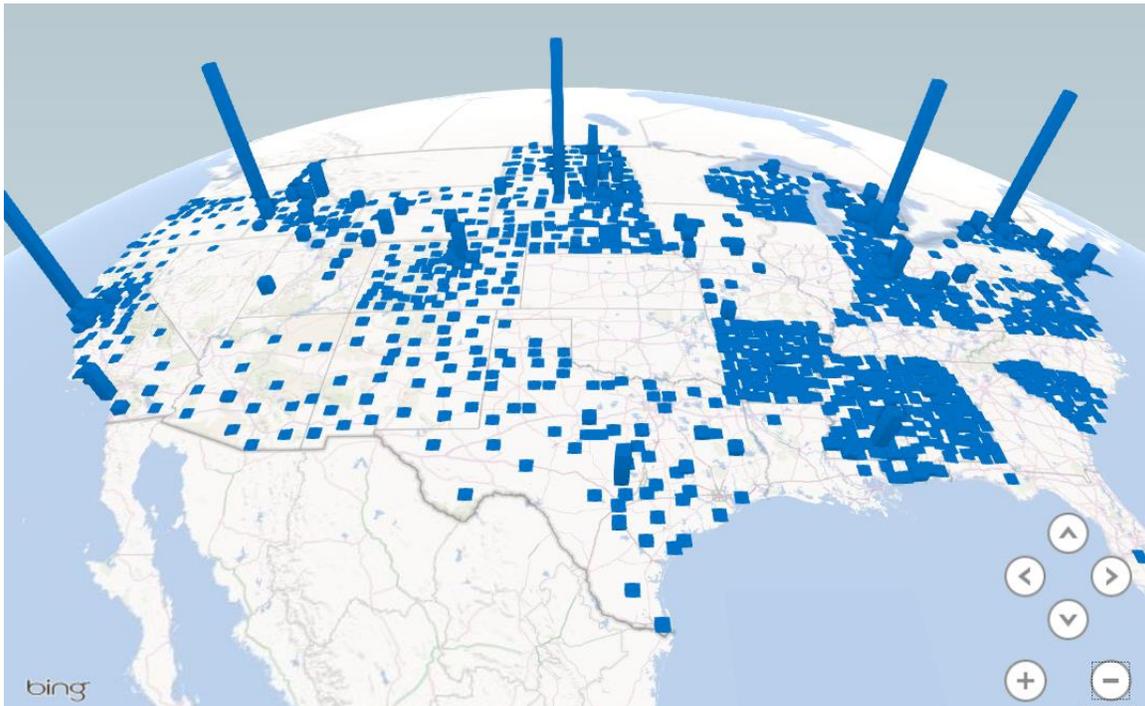


Figure 18: Population Per Square Mile, 2010