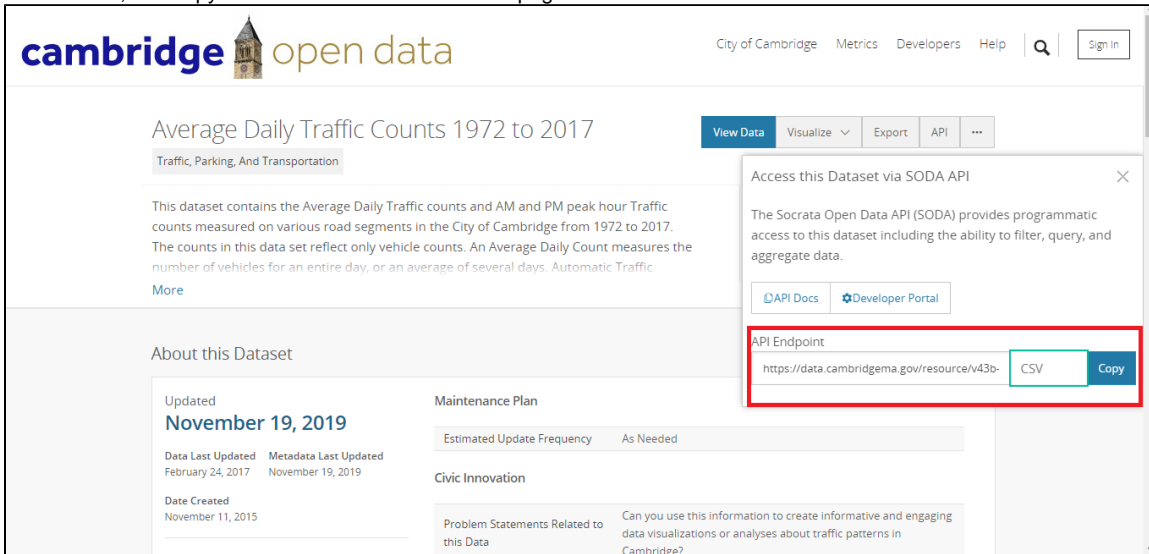


How to create a map from geolocation data in Zeppelin

How to create a map from geolocation data in Zeppelin

In this article, we will show you how to generate maps from geolocation data in Zeppelin notebook. We will be using the [Average Daily Traffic Counts 1972 to 2017](#) dataset from City of Cambridge Open Data project.

1. Click on API, and copy the CSV URL from the dataset page.



cambridge open data

City of Cambridge Metrics Developers Help

Average Daily Traffic Counts 1972 to 2017

Traffic, Parking, And Transportation

This dataset contains the Average Daily Traffic counts and AM and PM peak hour Traffic counts measured on various road segments in the City of Cambridge from 1972 to 2017. The counts in this data set reflect only vehicle counts. An Average Daily Count measures the number of vehicles for an entire day, or an average of several days. Automatic Traffic

[More](#)

About this Dataset

Updated
November 19, 2019

Data Last Updated February 24, 2017 Metadata Last Updated November 19, 2019

Date Created November 11, 2015

Maintenance Plan

Estimated Update Frequency As Needed

Civic Innovation

Problem Statements Related to this Data Can you use this information to create informative and engaging data visualizations or analyses about traffic patterns in Cambridge?

Access this Dataset via SODA API

The Socrata Open Data API (SODA) provides programmatic access to this dataset including the ability to filter, query, and aggregate data.

[API Docs](#) [Developer Portal](#)

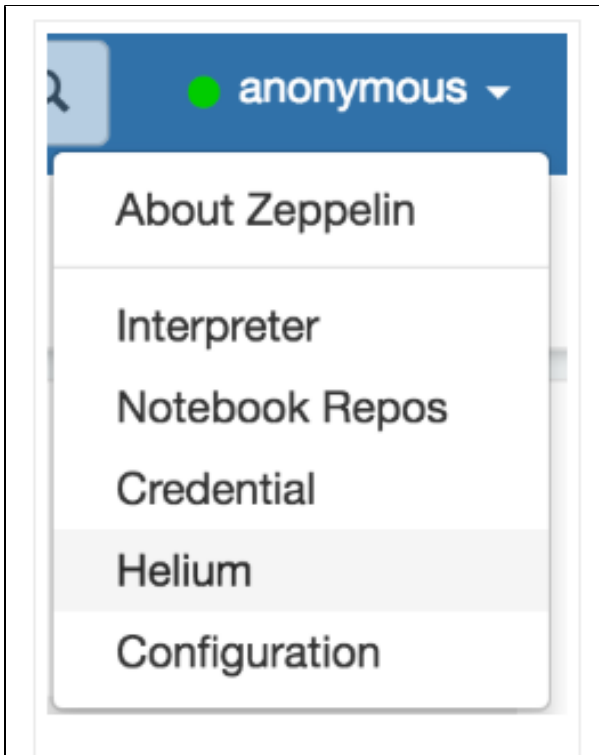
API Endpoint

<https://data.cambridgema.gov/resource/v43b-kqeq.csv> [CSV](#) [Copy](#)

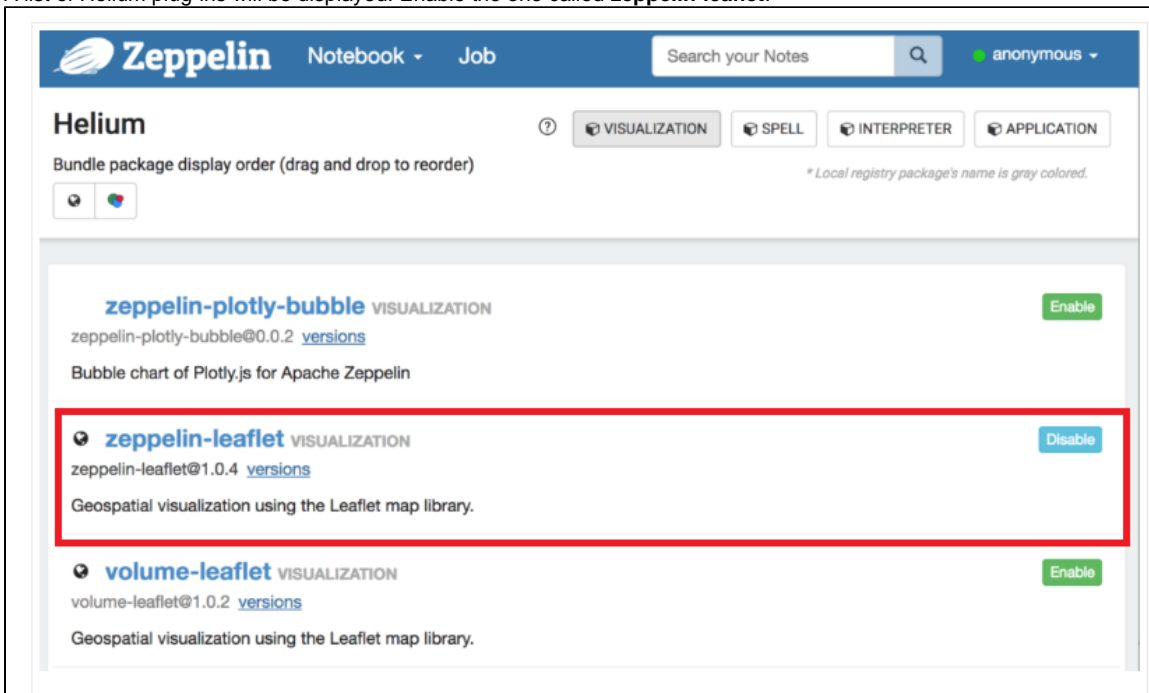
2. Enter the URL here:

```
python
import pandas as pd
url = 'https://data.cambridgema.gov/resource/v43b-kqeq.csv'
data = pd.read_csv(url)
```

3. In the upper right corner click on the login indicator and select **Helium**.



4. A list of Helium plug-ins will be displayed. Enable the one called **zeppelin-leaflet**.



5. Run the first paragraph of the notebook.

```
%python
import pandas as pd
url = 'https://data.cambridgema.gov/resource/v43b-kqeq.csv'
data = pd.read_csv(url)
data.head()
```

	id	primarystreet	pm_peak_2017	classification_counts
0	126	Putnam Avenue	NaN	NaN
1	206	T Egress Road (Alewife)	NaN	NaN
2	223	Frontage Road	NaN	NaN
3	336	Cardinal Medeiros	NaN	NaN
4	337	Cardinal Medeiros Avenue	NaN	NaN

[5 rows x 91 columns]

6. Once the data is loaded, run the second paragraph. Edit the SQL statement according to the column names of the dataset, and make sure to select the longitude and latitude columns in the SQL statement.

```
%python.sql
select longitude,latitude,_1999,source from data
```

Table Options

Name	Value
useFilter	<input type="checkbox"/>
showPagination	<input type="checkbox"/>
showAggregationFooter	<input type="checkbox"/>

7. Click on the map button, this will open up a map and will show you the column names you have selected. Drag and drop the longitude, latitude, and other corresponding column names in the fields, and the map will be plotted.

```
%python.sql
select longitude,latitude,_1999,source from data
```

Available Fields

longitude latitude _1999 source

latitude longitude tooltip popup

latitude x longitude x _1999 x source x

