

Open Source Geospatial Analysis



Last Modified: February 2018

This reference and training manual was produced by the University of Maryland Libraries.

Permission to reproduce this manual or any of its parts for non-commercial, educational purposes may be granted upon written request. Appropriate citation is appreciated.

University of Maryland Libraries
GIS and Spatial Data Services Center
McKeldin Library, Room 4120
College Park, MD 20742-7011

<http://www.lib.umd.edu/gis>

GIS Facilities at the University of Maryland

McKeldin Library

ArcGIS 10.5, ArcGIS Pro 2.2 and QGIS (an open source GIS software package) are loaded on all public workstations in McKeldin Library, including those located in the McKeldin 6101 instruction laboratory. The laboratory is open to the public during library hours when not in use by a class or librarian. The laboratory schedule is posted on the window by the door and updated each week. Color printing and large format printing are also available in McKeldin Library.

The GIS laboratory is located on the fourth floor of McKeldin Library in room 4120. This laboratory is available for use by faculty, staff, and students using geospatial methods in their research at the discretion of GIS and Spatial Data Services Center Staff. ArcGIS 10.5 and QGIS are installed on all computers in this laboratory, as well as other geospatial, image processing, and statistical packages.

For a complete list of software packages available, please see our GIS Facilities link <https://www.lib.umd.edu/gis/people-and-facilities>

Other Facilities

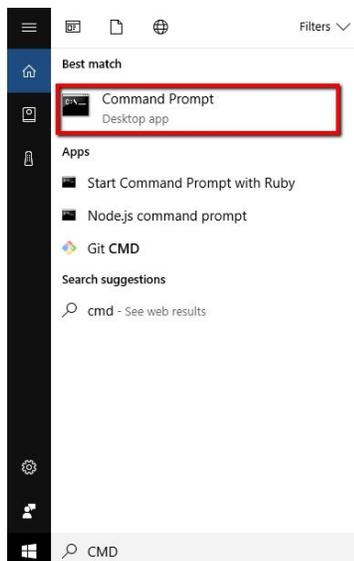
There are other departmental (e.g., anthropology, architecture/urban planning, civil engineering, geographical sciences, and landscape architecture) GIS laboratories on campus that are restricted to faculty, staff, and students in each of those disciplines. Additionally, the Office of Information Technology's software licensing program (<http://www.oit.umd.edu/slic>) offers ArcGIS at special/reduced rates for faculty and staff at the University of Maryland.

I. GDAL Utilities Exercise

In this exercise, We will use GDAL/OGR Command to manipulate the Raster and Vector data.

1. Work with the Vector layer(OGR)—Dissolve Vector features

1.1 Open your Command Prompt through search CMD in Windows search box.



1.2 Move to your work directory by **cd** Command.If your data folder in your downloads folder, you use **cd** **\Downloads\Open_Source_GIS\GDAL\states** to move to that directory. (If your folder name is different, than you will have to adjust your code accordingly).

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Li>cd Downloads\Open_Source_GIS\GDAL\states
C:\Users\Li\Downloads\Open_Source_GIS\GDAL\states>
```

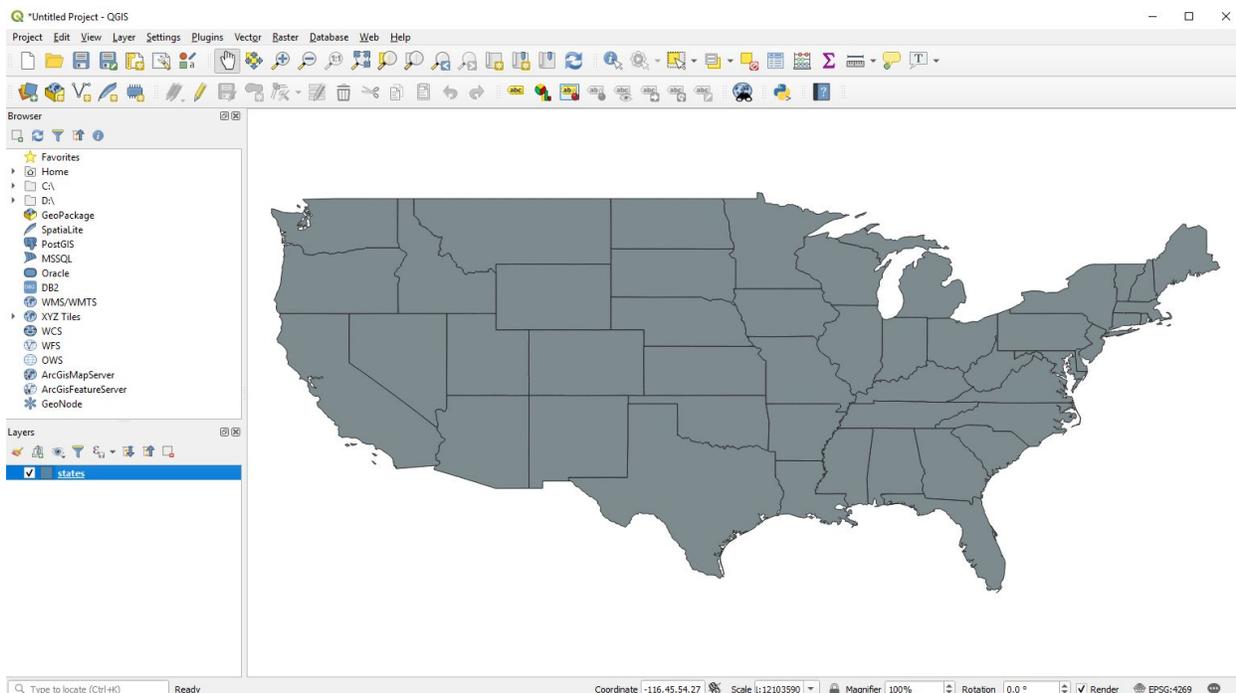
1.3 List vector layer general information by **ogrinfo**. Pass a name to the shapefile you want to check. For example, **ogrinfo states.shp** (**Do not forget the file extension and case sensitive!**)

```
C:\Users\Li\Downloads\Open_Source_GIS\GDAL\states>ogrinfo states.shp
INFO: Open of `states.shp'
      using driver `ESRI Shapefile' successful.
1: states (Polygon)

C:\Users\Li\Downloads\Open_Source_GIS\GDAL\states>_
```

1.4 Before the geoprocessing, we can open **states.shp** in QGIS to check.

To open a shapefile in QGIS, click on the Add Vector Layer toolbar button, or simply press **ctrl+shift+v**. This will bring up a new window.



2. Work with the Raster layer(GDAL)—GEOTIFF to KMZ

2.1 Now you can type the Command: **cd /** to go back your root path and move to your work directory by **cd**

Downloads\Open_Source_GIS\GDAL\ Night_Light to move to that

directory.

```
Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Li>cd Downloads\Open_Source_GIS\GDAL\Night_Light
C:\Users\Li\Downloads\Open_Source_GIS\GDAL\Night_Light>
```

2.3 List raster layer general information by `gdalinfo 2016.tif`

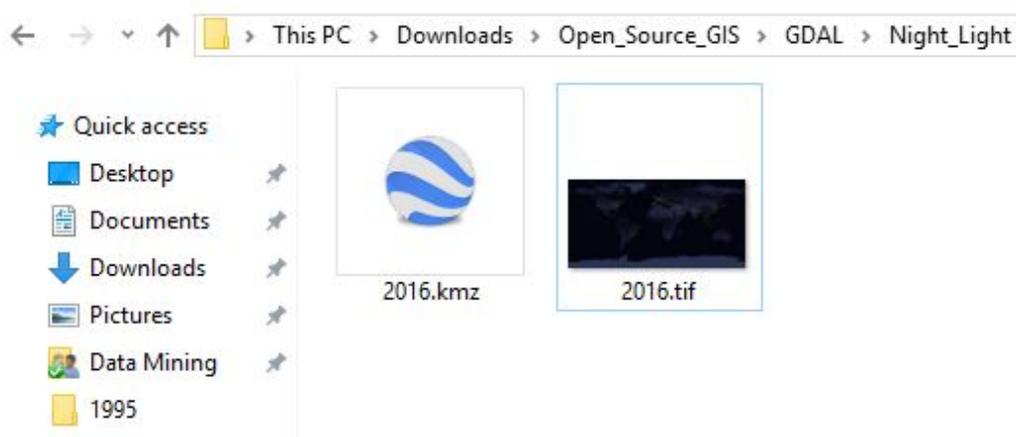
```
C:\Users\Li\Downloads\Open_Source_GIS\GDAL\Night_Light>gdalinfo 2016.tif
Driver: GTiff/GTIFF
Files: 2016.tif
Size is 13500, 6750
Coordinate System is:
NAD83 / NAD 83 / Plate Carree (deprecated),
GEOGCS["NAD 83",
  DATUM["NAD_1983",
    SPHEROID["NAD 83", 6378137, 298, 257223563,
      AUTHORITY["EPSG","7830"]],
    AUTHORITY["EPSG","8207"]],
    PRIMEM["Greenwich",0,
      AUTHORITY["EPSG","3201"]],
    UNIT["degree",0.0174532925199433,
      AUTHORITY["EPSG","3212"]],
    AUTHORITY["EPSG","3221"]],
  PROJECTION["Equidistant"],
  PARAMETER["latitude_of_origin",0],
  PARAMETER["central_meridian",0],
  PARAMETER["false_easting",0],
  PARAMETER["false_northing",0],
  UNIT["metre",1,
    AUTHORITY["EPSG","9001"]],
  AXIS["X",EAST],
  AXIS["Y",NORTH],
  AUTHORITY["EPSG","32627"]],
Origin = (-180.000000000000000,0.000000000000000)
Lyz Size = (0.026666666666667,-0.026666666666667)
Metadata:
  AREA_OR_POINT=Area
  TIFFTAG_DATETIME=2017-04-11 09:41:30
  TIFFTAG_RESOLUTIONUNIT=2 (pixels/inch)
  TIFFTAG_SOFTWARE=Adobe Photoshop CC 2014 (Macintosh)
  TIFFTAG_XRESOLUTION=72
  TIFFTAG_YRESOLUTION=72
  Image Structure Metadata:
    COMPRESSION=LZW
  INTERLEAVE=BGR
  Other Coordinates:
  upper Left (-180.0000000, 90.0000000) ( 0d 0' 5.827N, 0d 0' 2.917W)
  upper Right (-180.0000000, 90.0000000) ( 0d 0' 5.827N, 0d 0' 2.917W)
  lower Right (-180.0000000, -90.0000000) ( 0d 0' 5.827S, 0d 0' 2.917W)
  lower Left ( 0.0000000, -90.0000000) ( 0d 0' 0.017S, 0d 0' 0.017W)
  and 1 Block=1350x1 Type=Byte, ColorInterp=Red
  and 2 Block=1350x1 Type=Byte, ColorInterp=Green
  and 3 Block=1350x1 Type=Byte, ColorInterp=Blue
```

2.4 Then we can convert the 2016 Night light image from GEOTIFF format to Google KMZ format by the Command:

`gdal_translate -of KMLSUPEROVERLAY 2016.tif 2016.kmz -co FORMAT=JPEG`

```
C:\Users\Li\Downloads\Open_Source_GIS\GDAL\Night_Light>gdal_translate -of KMLSUPEROVERLAY 2016.tif 2016.kmz -co FORMAT=JPEG
Input file size is 13500, 6750
0...10...20...30...40...50...60...70...80...90...100 - done.
C:\Users\Li\Downloads\Open_Source_GIS\GDAL\Night_Light>
```

2.5 Now we can check the folder.



Study Drills:

* 1. Use GDAL Command to mosaic adjacent two views of Landsat 8 images (any periods). The data can be download from earth explorer: <https://earthexplorer.usgs.gov/>.

** 2. Convert the Mosaicked data from tiff to KMZ format with GDAL Command.

***3. Why do we need to use GDAL to process spatial data?

II. 3D Thematic Web-Mapping Exercise

II.1 Background of QGIS

Quantum GIS (QGIS) is a powerful Open Source GIS platform, which integrated lots of other open source software and packages such as SIG GIS, Grass GIS. Many of developers or Cartographers would create their plugins or tools to contribute this society.

In this exercise, we will aggregate and grid 2016 Crime incidents Data of Washington D.C and also involved several QGIS plugins in creating a 3D web map. In the end, we will publish our 3D Map to Github Page. So everyone can see the Web Map through a Github page URL.

Qgis2threejs plugin is a QGIS plugin, which can export terrain data, map canvas image, and vector data to the web browser. We can view 3D objects in the web browser (You need a web browser which supports WebGL).

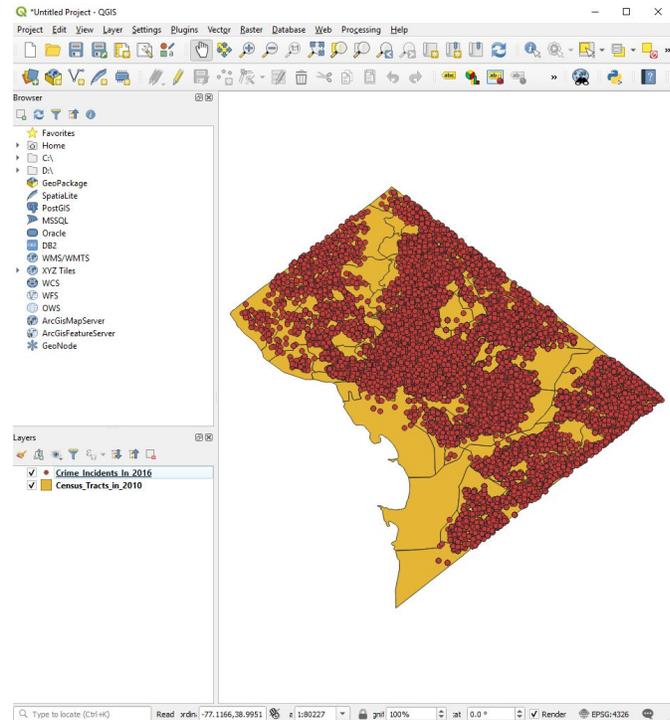
II.2 Tasks Including in this exercise:

1. Points Data Aggression.
3. QGIS Scripts tool
4. QGIS plugins Mangement
5. Hexagon Mapping.
6. HTML code modified.

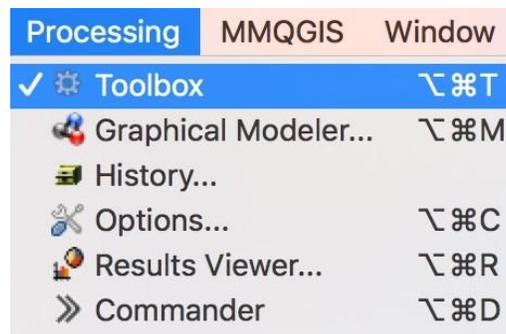
II.3 Exercise:

1. Using QGIS (Script Tool) to aggregate data

1.1 Open QGIS and add “Crime_Incidents_in_2016.shp” and “Census_Tracts_in_2010.shp” by clicking on Add Layer > Add Vector Layer. (In the layers panel, make sure Crime_Incidents_In_2016 is on top of Washington_DC_Boundary layer)



1.2 Click **Processing > Toolbox**.



1.3 In the **Processing Toolbox**, search for the tool called “**Create Grid**”. Double click the tool to open it.

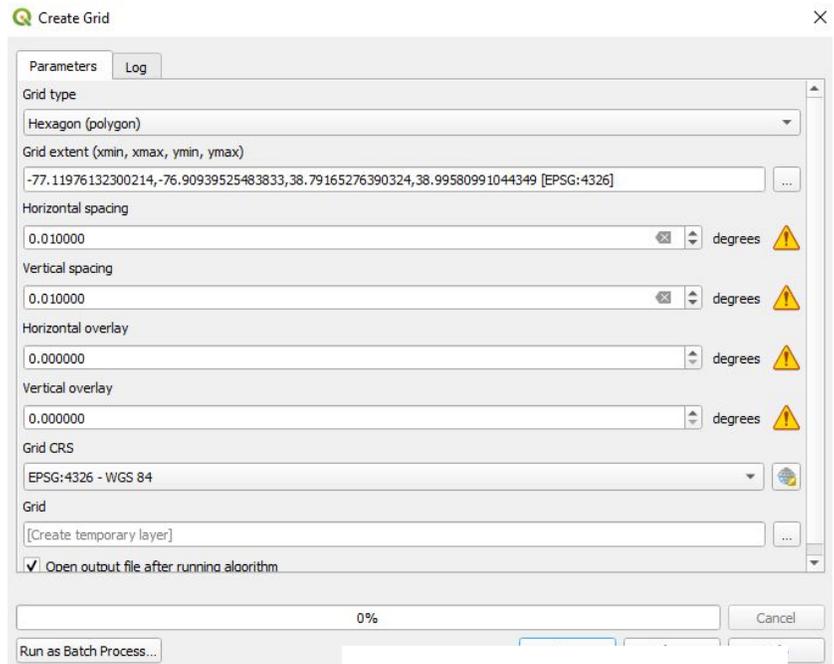
1.4 Set the Grid Type to **Hexagon (Polygon)**.

1.5 For the grid **Extent**, click the ... box on the far right and select the layer extent of the census layer.

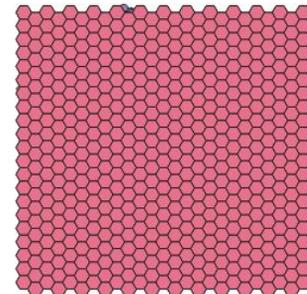
1.6 Set both **Horizontal** and **Vertical spacing** to 0.01

1.7 Click **Run**

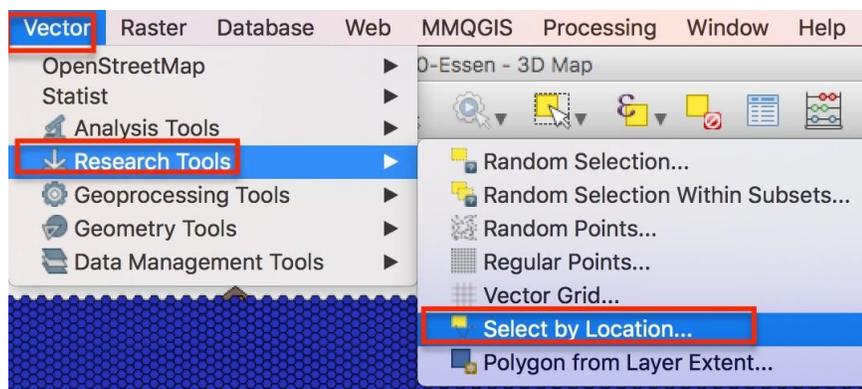
A grid layer should appear over the extent of your layer. The current grid layer is in the memory not on the hard disk. If you close the QGIS right now the grid layer would be missing, and you have to redo the whole process again.



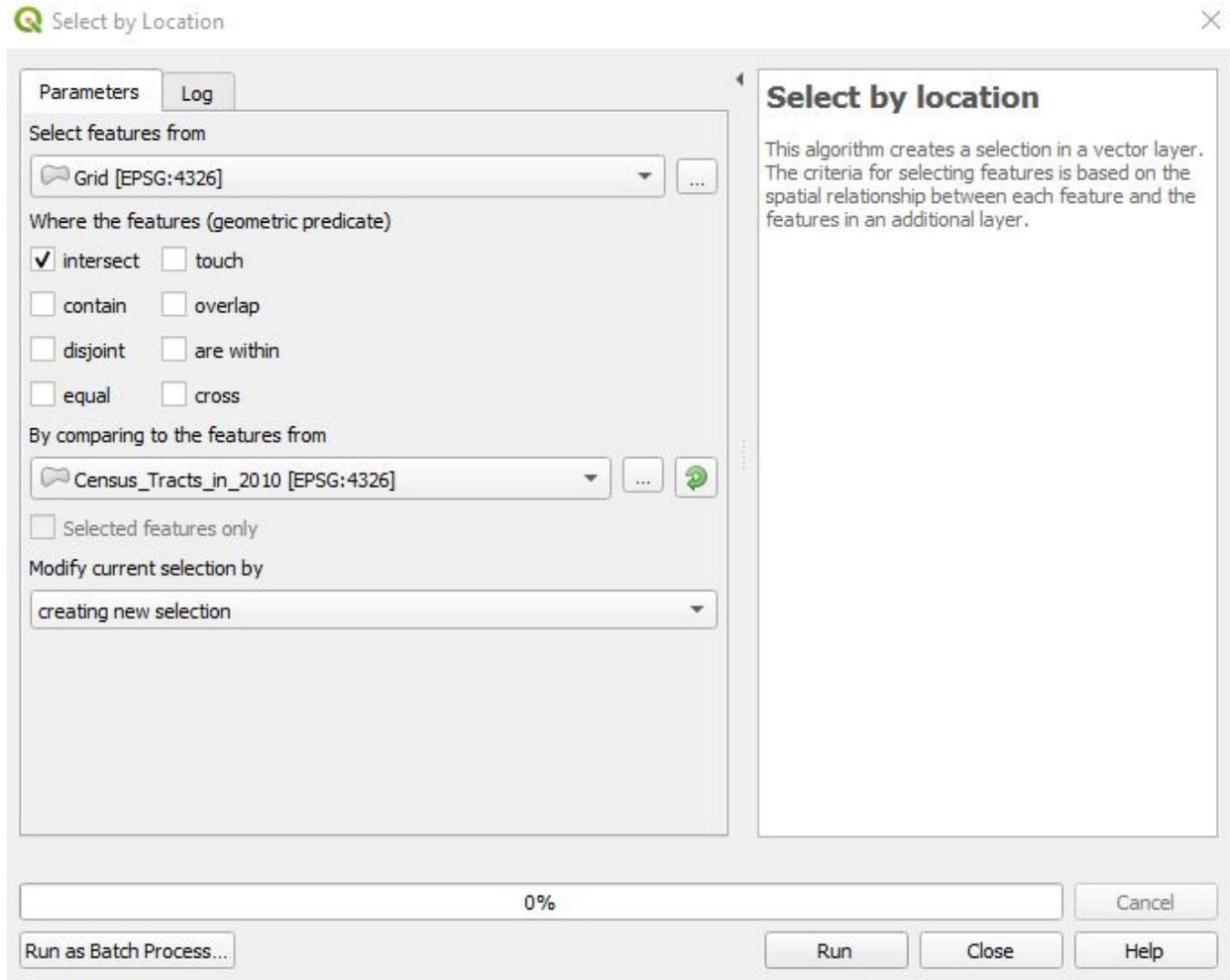
1.8 The hexagon layer will generate and show it in the QGIS canvas.



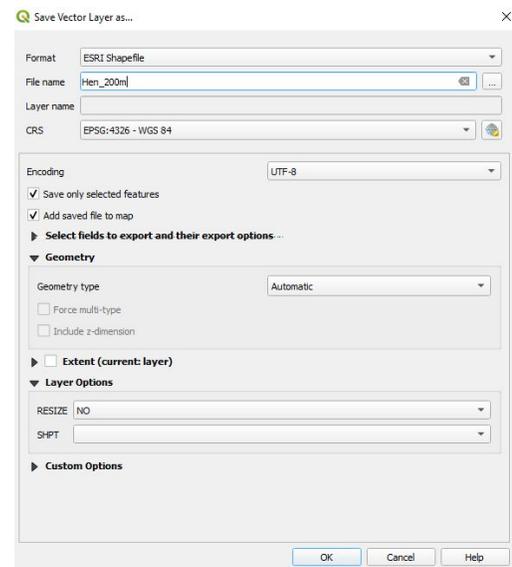
1.9 Now we need to aggregate the crime incident points to the grid later. Open **Vector > Research Tools > Select by Locations...**



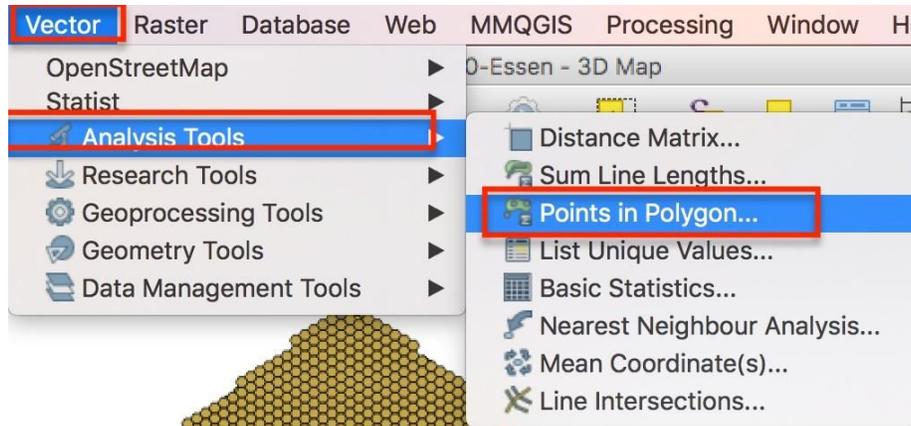
1.10 Choose “**grid**” in select features and “**Census_Tracts_in_2010**” in “By comparing to the features from” and make sure to check the **box** for **intersect**. Click Run then close the window.



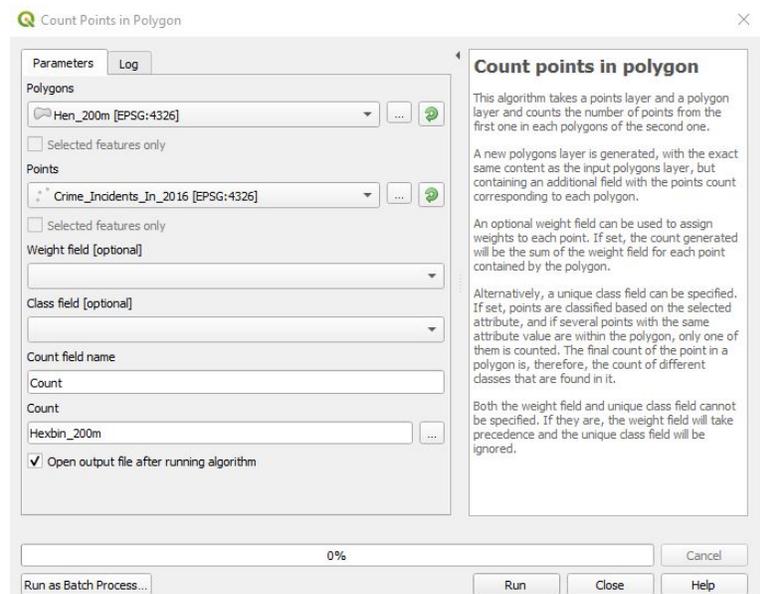
1.11 Right click “grid” layer in **Layers Panel >Export> Save Selected Features as**. Choose a path and name it as “Hen_200m” for the output layer and check the box of “Save only selected features.” Click OK.



1.12 Now you can remove the temporary grid layer from the **Layer Panel**. Click **Vector > Analysis Tools > Count Points in Polygon...**



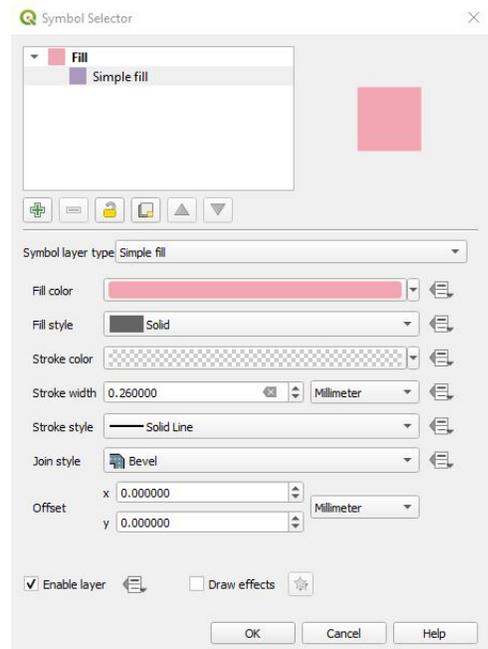
1.13 Now we will count the crime incidents in each Hexagon. In the popup window, set the file of **polygon** as “**Hen_200m**”, set the field of **Points** as “**Crime_incidents_in_2016**,” set the field of **Count field name** as “**Count**”. Finally, set the fields of **Count** to the same path and name it as “**Hexbin_200m**”. Check on the **Open output file after running algorithm**. Click **OK** and **close** the dialog window.



1.14 Right click “Hexbin_200m” layer to open **the attribute table**. We can see there is a **Count** field in the attribute table. (This layer may also be called Count - this sometimes occurs as a bug, but can be renamed)

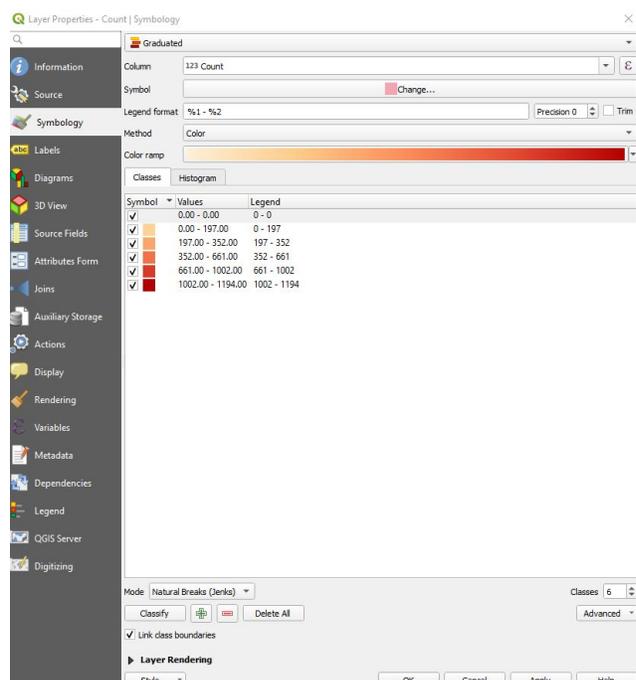
1.15 Single left clicks the field **Count** and set the descending order of the crime incidents of each hexagon. Then we will get the biggest number in the hexagon, which is 1194.

1.16 Right click “Hexbin_200m” layer to open **properties**. On the top drop down menu, change the selection from “**Single Symbol**” to **Graduated**. Choose “Count” in Column and click **Symbol > Fill > Simple fill** and change the Stroke color to transparent (may have to scroll down). Hit the OK.



Then choose “OrRd” in **Color ramp** and “Natural Breaks (Jenks)” in **Mode**. Enter “6” in **Classes**. In the **Symbol field**, double-click and set the first field from 0 to 0. Then double-click **the first symbol** and make the **Opacity** to 0% and click OK.

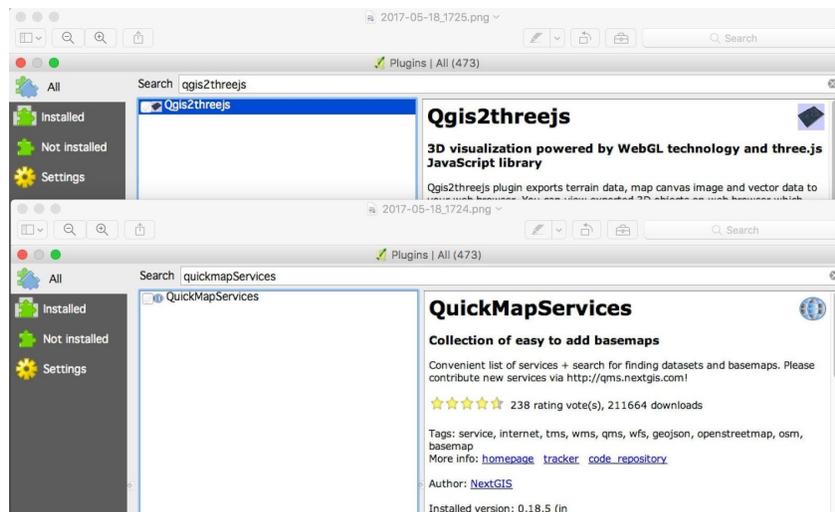
1.17 Zoom out the view of Hexagon Map a little bit. Now we have finished the Hexagon Map of crime Incidents of Washington DC in 2016!



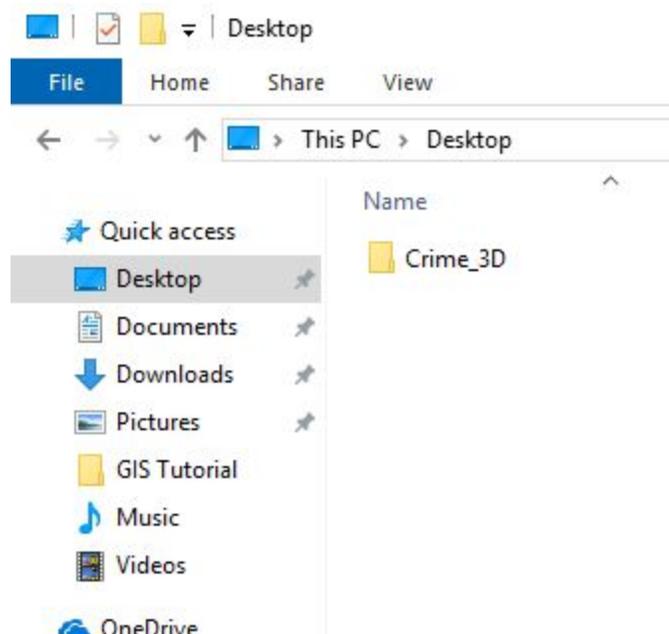
2. Using QGIS plugin (Qgis2threejs) to create 3D Map

2.1 Open Plugin > Manage and Install Plugins...

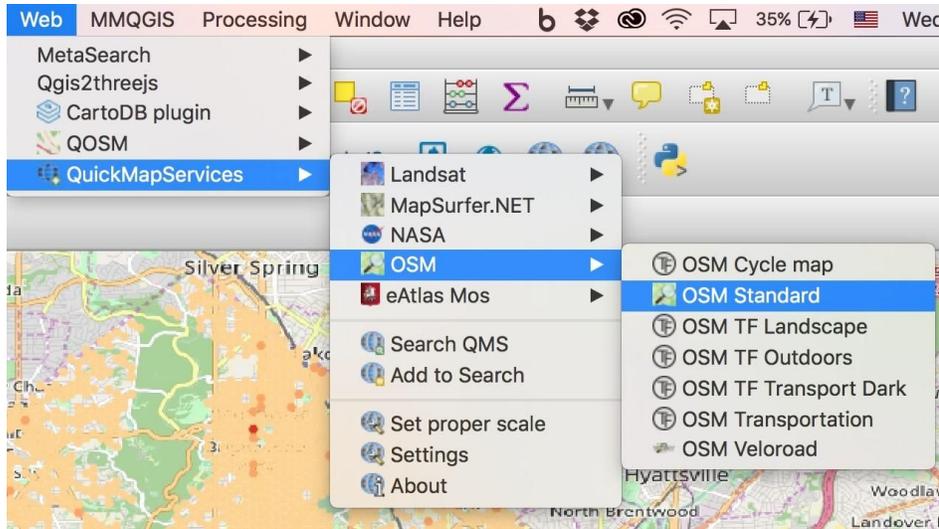
A **plugin search box will pop up** and select **All** in left side of this window Then input “Qgis2threejs” and install the plugin “**Qgis2threejs**”. Then we install another plugin name as “quickmapServices” After we installed these two plugins. Close the current window.



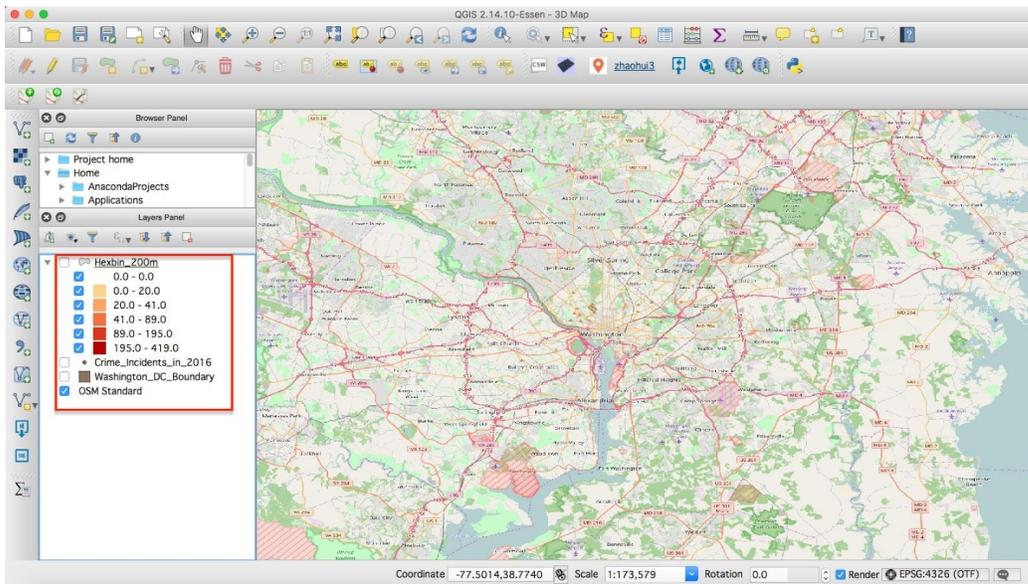
2.2 Minimize the QGIS window and create a new folder on your PC Desktop and name it as Crime_3D.



2.3 Open **Web > QuickMapServices > OSM > OSM Standard**.
Now the OSM Standard base-map would add to your QGIS canvas.



2.4 Maximize the QGIS window and Check off all layers except the OSM Standard layer in Layers Panel.



2.5 Now we open the **Qgis2threejs Exporter** tool. Under **Polygon** check on box of “Count”. Now select File > Export to Web. Choose the path to the folder, which was we created in our desktop.

Name the output HTML file as **Crime_2016** and run it. The 3D map will pop up in our browser!

III. 2D Thematic Web-Mapping Exercise

In this exercise, we will aggregate and grid 2016 Crime incidents Data of Washington D.C and also involved more HTML CSS programming in creating a 2D thematic web map. In the end, we will publish our 2D Map to Github Page.

qgis2web plugin generates a web map from your current QGIS project, either as OpenLayers or Leaflet. It replicates as many aspects of the project as it can, including layers, styles (including categorized and graduated), and extent. No server-side software required.

CARTO (formerly CartoDB) is a Software as a Service (SaaS) cloud computing platform that provides GIS and web mapping tools for display in a web browser. The company is positioned as a Location Intelligence platform due to tools with an aptitude for data analysis and visualization that do not require previous GIS or development experience.

CARTO users can use the company's free platform or deploy their instance

of the open source software. CARTO is offered as freemium service, where accounts are free up to a certain size. For larger accounts, a fee is applied.

Carto Exercise

What is Carto?

Carto is an open source software platform that provides GIS applications and web mapping tools to display in a web browser. You do not need to have GIS or developmental experience because the company runs on a basis for creating tools for data analysis and data visualization for people with little to no experience. It uses JavaScript intensely.

Many organizations, groups, and companies use Carto, such as Nokia, Wall Street Journal, Twitter, and NBC.

● Tasks in this exercise:

1. Account Setup
2. Connecting data set
3. Creating a Filter
4. Creating a Time Series Map

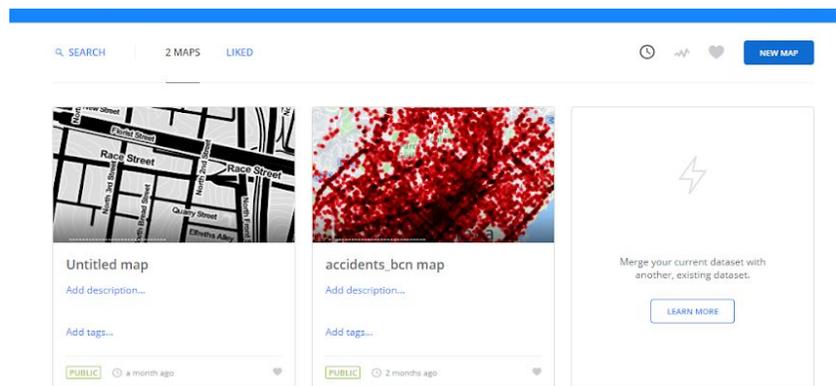
1. Account Setup

1.1 Navigate to: <https://github.com/join?source=header-home> and create a github account.

1.2. Once you have created an account, navigate to: <https://carto.com/students-access/> and click where it says “Sign up with Github”. This will allow you to access the student version of Carto.

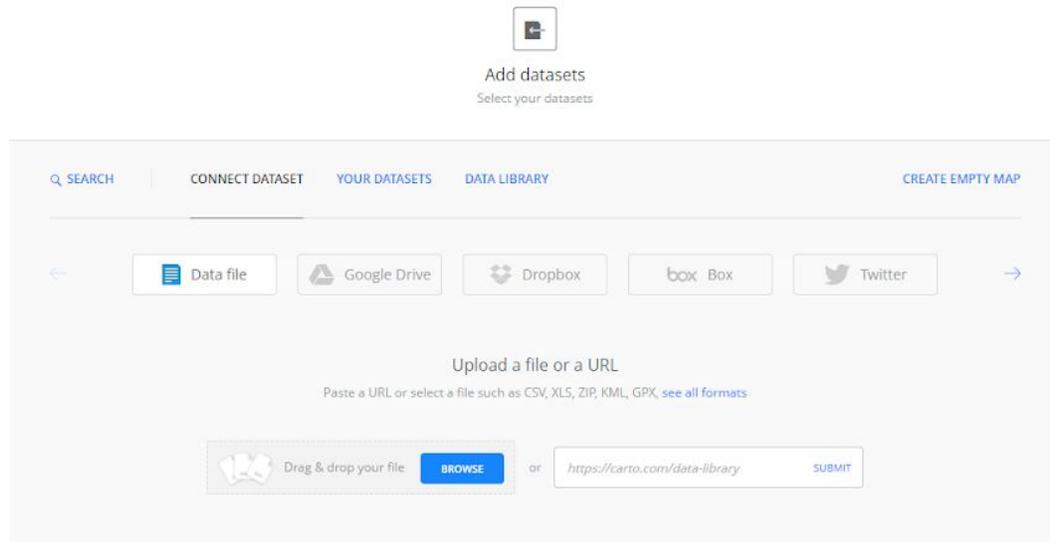
- a. Enter your GitHub credentials

1.3. Once logged in to Carto, select “**New Map**” on the top right

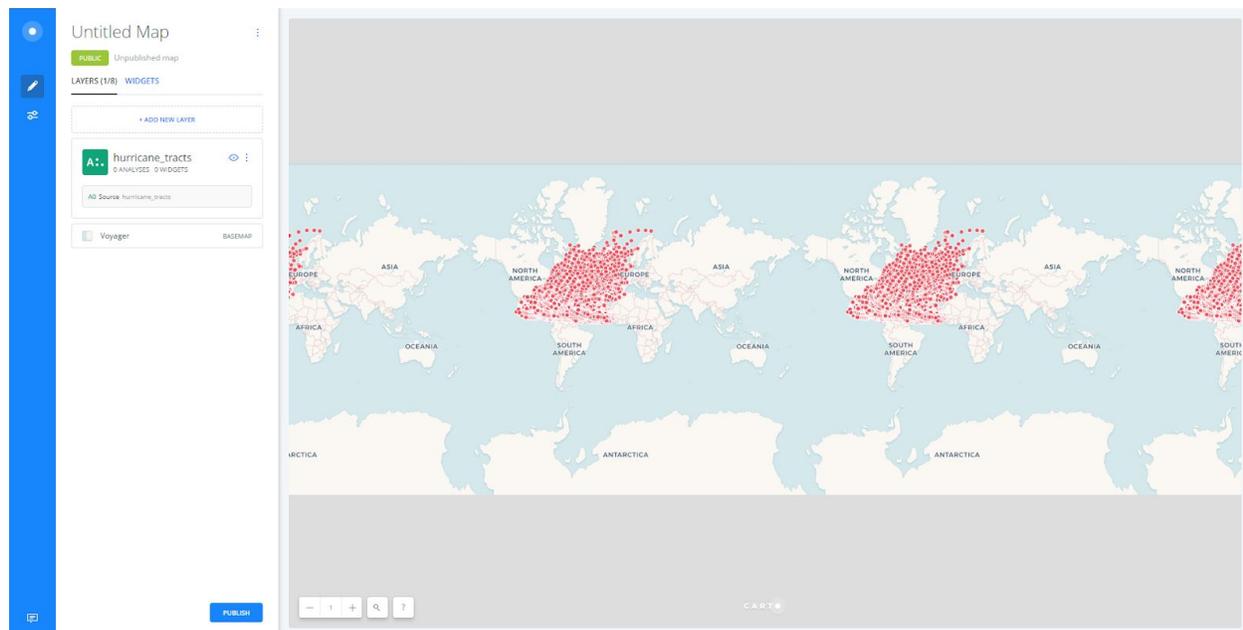


2. Connecting data set

- 2.1. Make sure you are brought to **Add datasets window** and click on **Connect Dataset**, and then click on the blue **Browse** button.



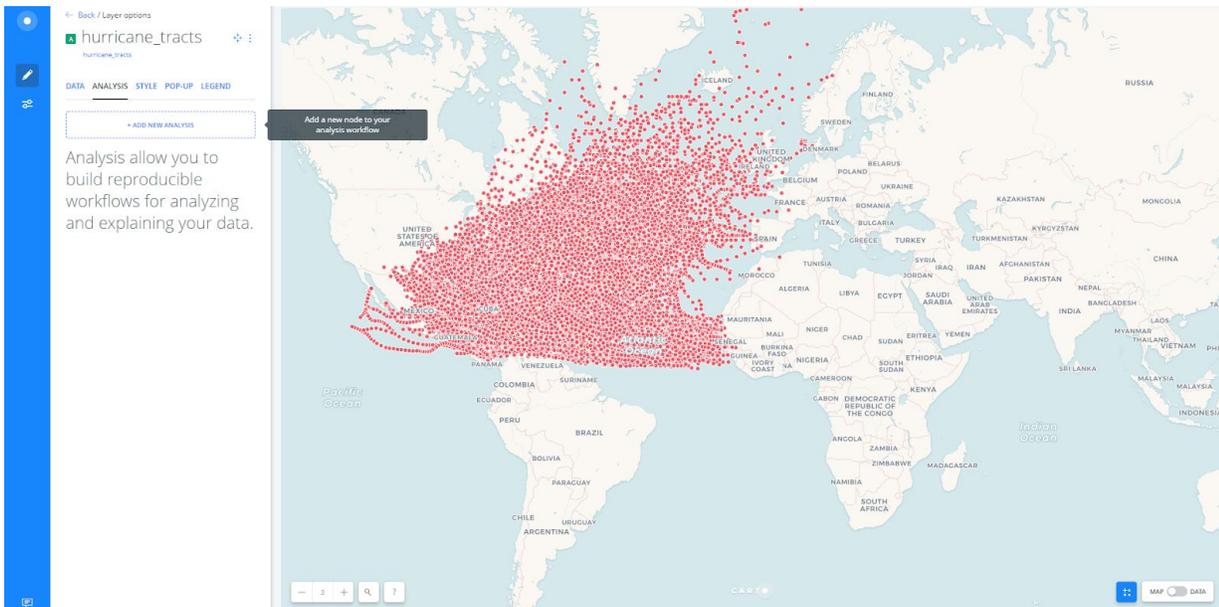
- 2.2. Browse through your files and upload the “hurrican_tracts.xlsx” file.
 - a. Connect the dataset



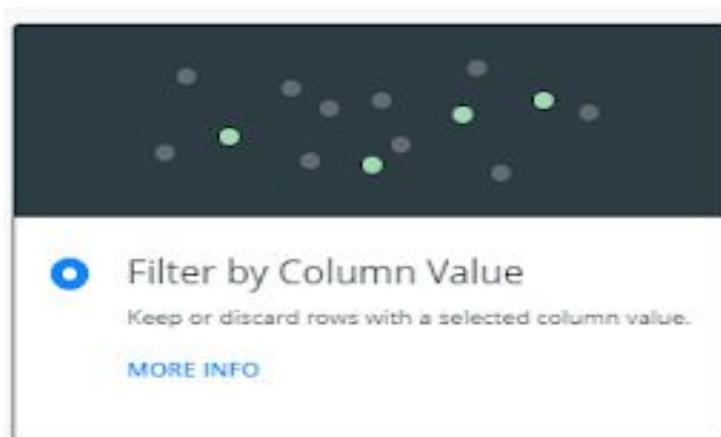
- 2.3. Once the dataset has loaded, you will be brought to your map.

3. Creating a Filter:

- 3.1 Click on the “Hurricane Tracts” layer that is on the left.
- 3.2 Click on the “Analysis” page that is right between Data and Style on the left.
- 3.3. Click to “Add new analysis”



3.4. Select, Filter by Column Value

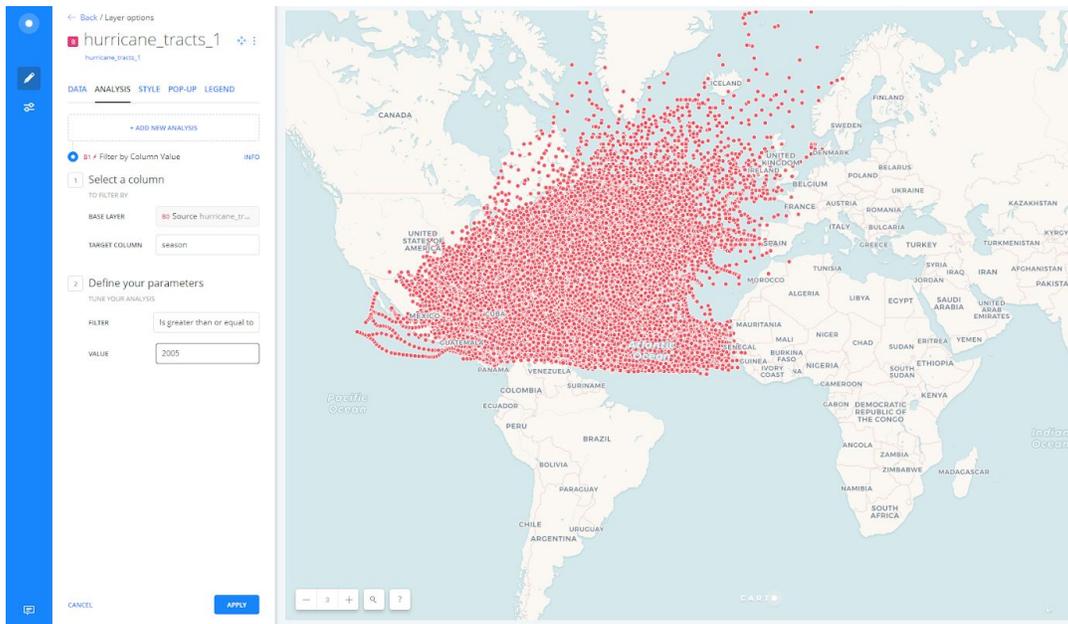


- a. Click to “Add Analysis”

- 3.5. Change the **Target** Column to “Season” - This is the year that the each hurricane took place. Note, if you don’t see it when you scroll through, you

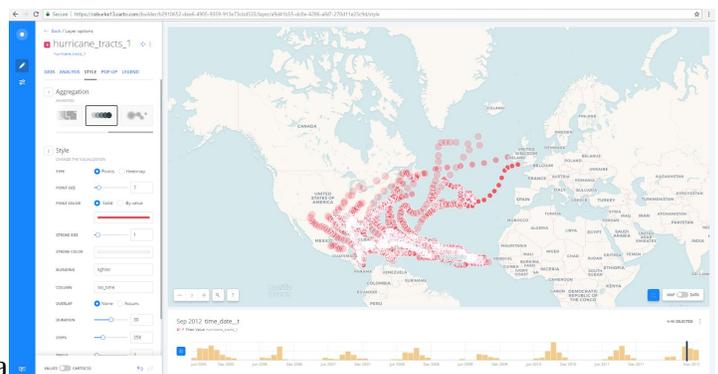
can search for it in Search for column.

- 3.6. Change the filter parameters so that only hurricanes that occurred in 2005 or later are displayed.
 - a. Filter=Is Greater than or Equal to
Value=2005
 - b. Click Apply
 - c. Click Done



4. Creating a Time Series Map

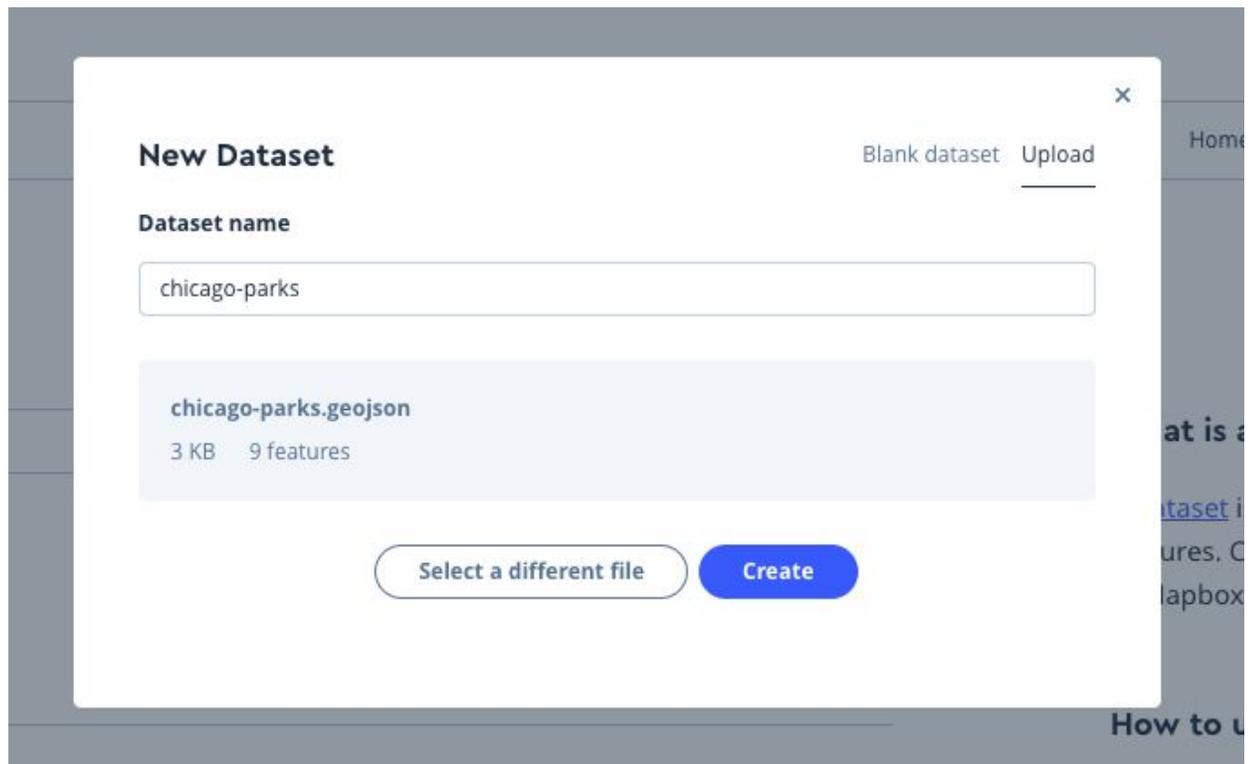
- 4.1. Mouse over to the “Style” Tab
- 4.2. Under “Aggregation”, select “Animated”(To find animated, push the scroll bar to right)
- 4.3. Under “column”, select “Iso_time”
(If you don't see "column", click on "labels" box, then you will see column.)
 - i. This will now display the data as time series data based on when each hurricane occurred.
- 4.4. You can now manipulate the settings of this visualization to make the map more visually appealing.



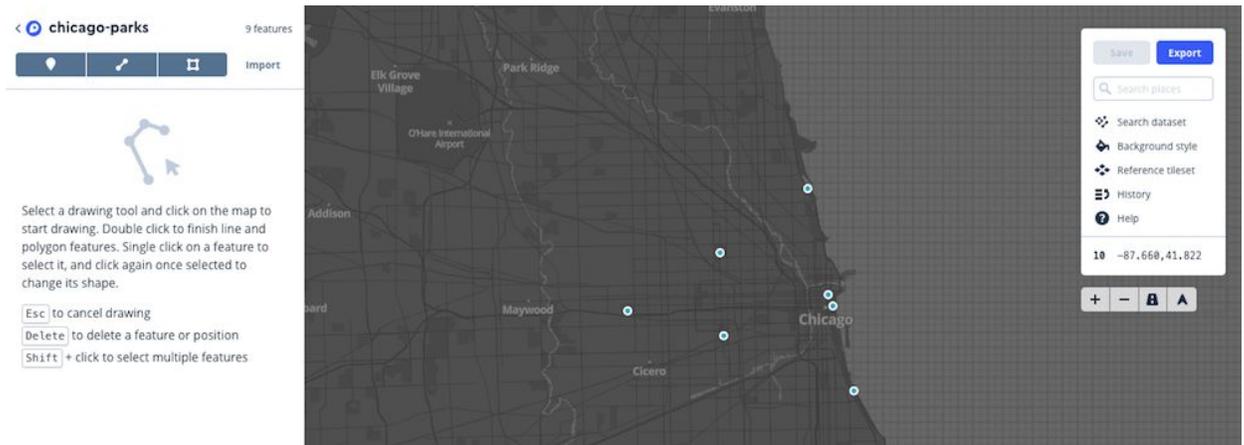
1. Mapbox

Mapbox is a mapping platform for developers. The building blocks make it easy to integrate location into any mobile or online application. You can customize your map tile through Mapbox, which data Mainly from the Open street map.

1. Create a Mapbox account. Sign up for free at mapbox.com/sign-up.
2. Create a dataset
 1. Log into [Mapbox Studio](https://www.mapbox.com/studio) (<https://www.mapbox.com/studio>) and navigate to the [Datasets page](#).
 2. Click the New dataset button.
 3. Select the Upload option in the upper right corner of the *New Dataset* modal.
 4. Select the GeoJSON you just downloaded. Click Confirm, then click Create.



5. Once your file has completed uploading, click Start editing. The dataset editor automatically opens, and the data is displayed on a dark base map to help visualize the features.



About datasets

You can edit both dataset features and properties in the dataset editor:

- **Features** are the points, lines, and polygons on your map. You can add new features using the draw tools, edit the placement or shape of features using by clicking and dragging features on the map, or remove features all together by clicking on them and hitting the delete key. You can also click on each feature in the dataset editor to view its properties.
- **Properties** can be strings, numbers, or booleans. In the sample dataset you uploaded, there are `title` and `description` properties for each point with a unique text string. You can edit properties, add new properties, or delete properties in the dataset editor. *Make sure any content that you want to display in popups in your final product is included as a property while you are working in the dataset editor.*

Draw a new feature

1. Click inside the Search places field in the upper right side of the editor and search for `Garfield Park Chicago`.
2. Use the draw tool to create a new point on the map at that location.
3. Next, click the Add Property button. Use the properties list on the left side of the screen to add the following field name and description:



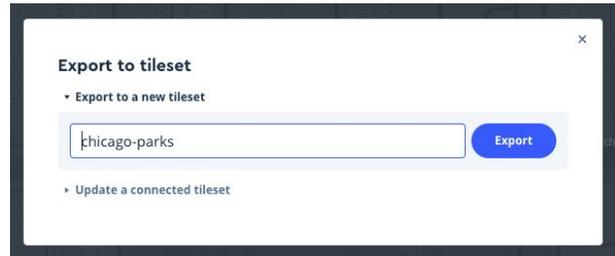
- Add the field name `title` (must be lowercase) and give it the value `Garfield Park`. Click **Confirm**.
- Add the field name `description` (must be lowercase) and give it the value `Home of the Garfield Park Conservatory`. Click **Confirm**.

Export the dataset as a tileset

Next, you will save and export the dataset as a tileset so that you can add it to a Mapbox style.

Create a tileset

1. Click the **Save** button in the upper right corner of the editor and save your changes.
2. Click the **Export** button, then select *Export to a new tileset*.
3. Name your tileset `chicago-parks` and click **Export**



About tilesets

Web maps are made up of [map tiles](#). A collection of tiles is called a tileset. Data must be cut up into tiles so it can be added to a web map and displayed at various zoom levels. To make Mapbox maps more performant, a dataset's features are simplified when it is converted into a tileset.

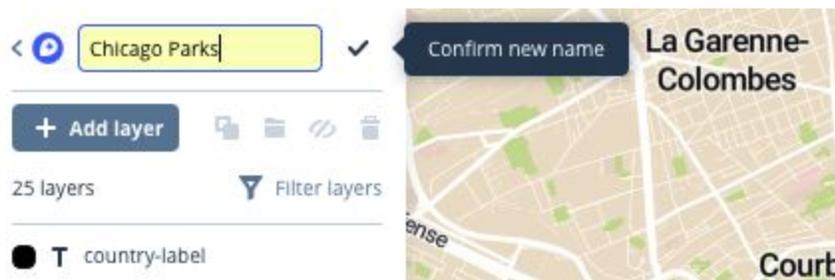
Add data to a style

Web maps are made up of [map tiles](#). To add your data to a web map, it needs to be cut up into tiles so the data can be displayed at various zoom levels. In Mapbox maps, the collection of tiles your data is cut up into is called a **tileset**.

[Create a new style](#)

On your [Styles page](https://www.mapbox.com/studio/styles) (<https://www.mapbox.com/studio/styles>) in Mapbox Studio, click the **New style** button. Find the *Basic Template* style and click **Create**.

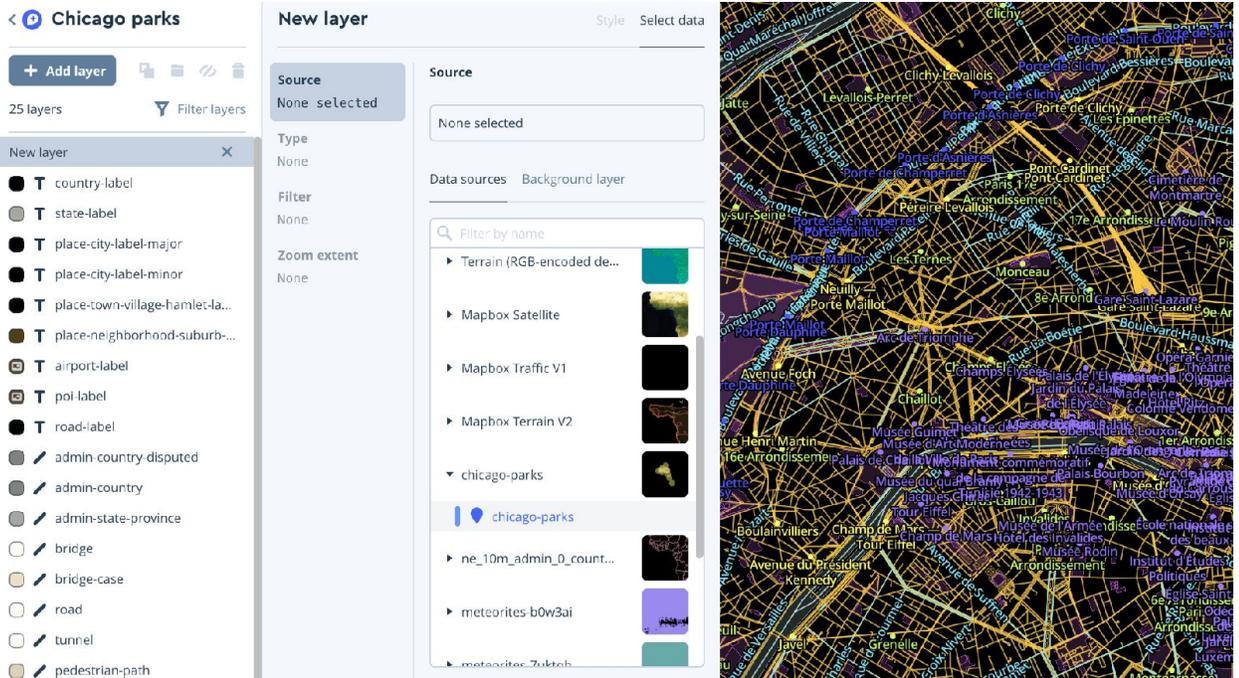
The style editor will open automatically. Rename your new style to *Chicago Parks* using the title field in the upper left side of the screen.



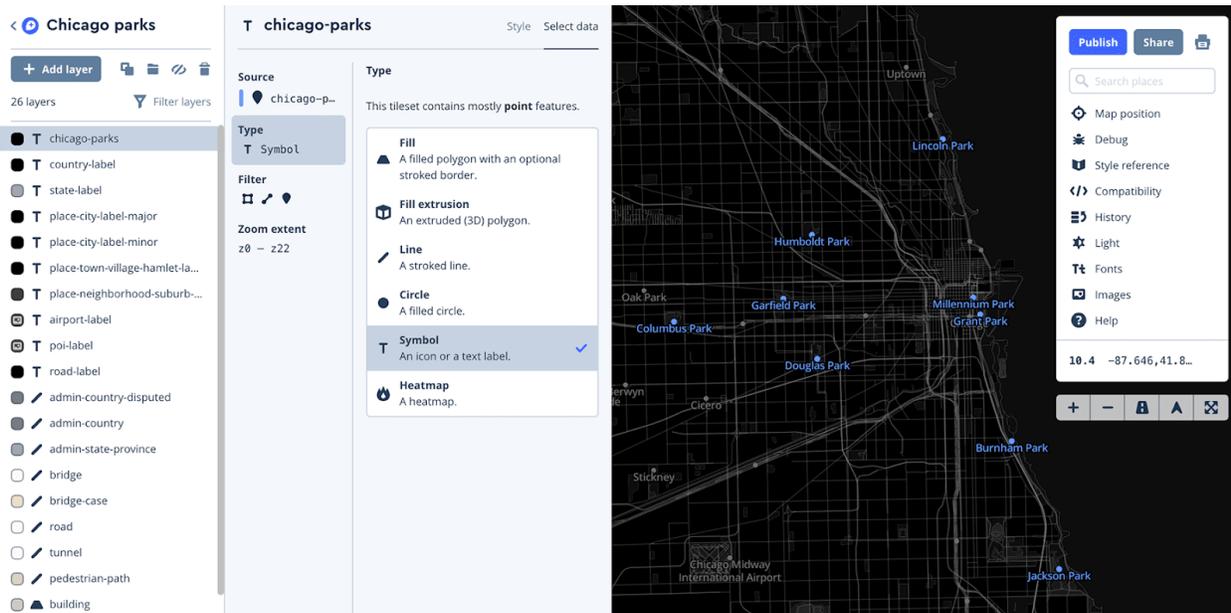
[Create a new layer](#)

You will add your tileset as a new layer here:

1. When the style editor opens, click **+ Add layer** in the upper left.
2. In the list of *Data sources*, find your **chicago-parks** tileset, click the name of the tileset, and then click the source layer that appears to add it as the source for the layer.



3. Mapbox Studio recognizes that the data you have uploaded is focused on a different location, so it displays the message *"This tileset isn't available from your map view."* Click **Go to data**, and the map view will refocus on Chicago.
4. Click the **Type** option, and then choose the **Symbol** layer option so you can create a layer with [markers](#).
5. Click back to the **Style** tab.

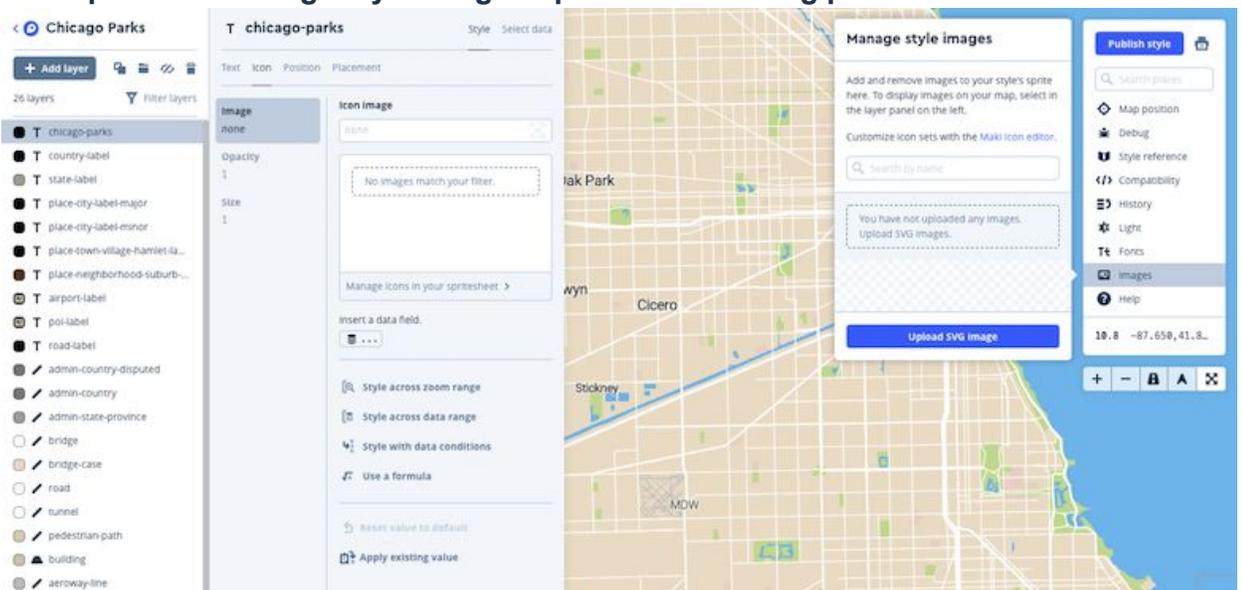


Style a layer

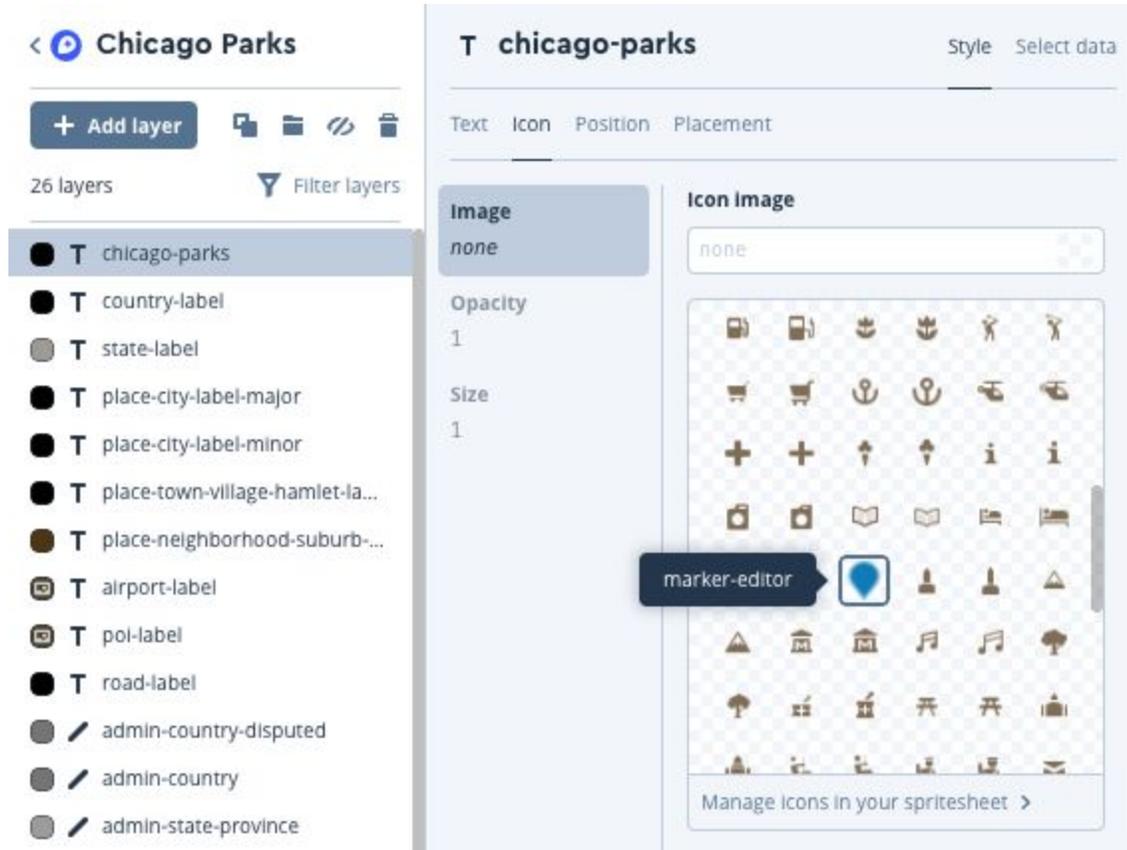
In the Mapbox Studio style editor, you can specify the style properties of each layer. This includes the layers in the Mapbox default styles and any layers you've added with custom data. For this example, you will style a *symbol* layer. You can style symbol layers with both text and icons.

Style as a symbol layer

1. Click on the **chicago-parks** layer you created in the layer list on the left side of the style editor.
2. When the style panel opens, click the **Style** tab (for **chicago parks**) if you're not already there.
3. Select the **Icon** tab and click on **Manage icons** in your spritesheet.
4. This opens the **Manage style images** option in the debug panel.

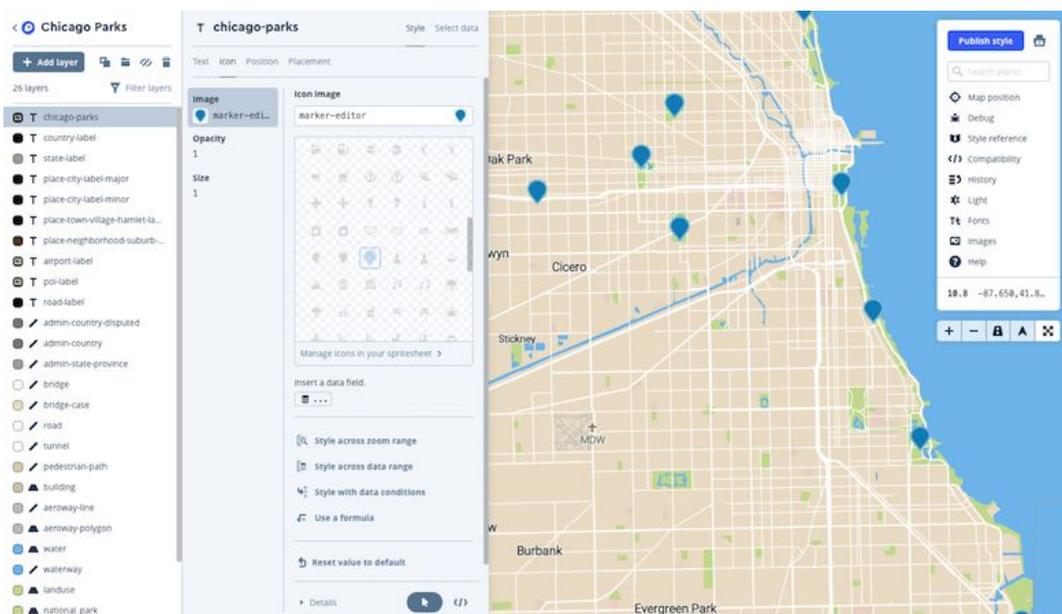


5. Click the **Upload SVG Image** button to and choose the marker-editor file you downloaded at the beginning of this tutorial from your files.
6. Once the icon has loaded, choose it from the list.



7. If you would like to show all the markers, even if they overlap, click on the Placement tab. Scroll down to Allow icon overlap and set it to `True`.

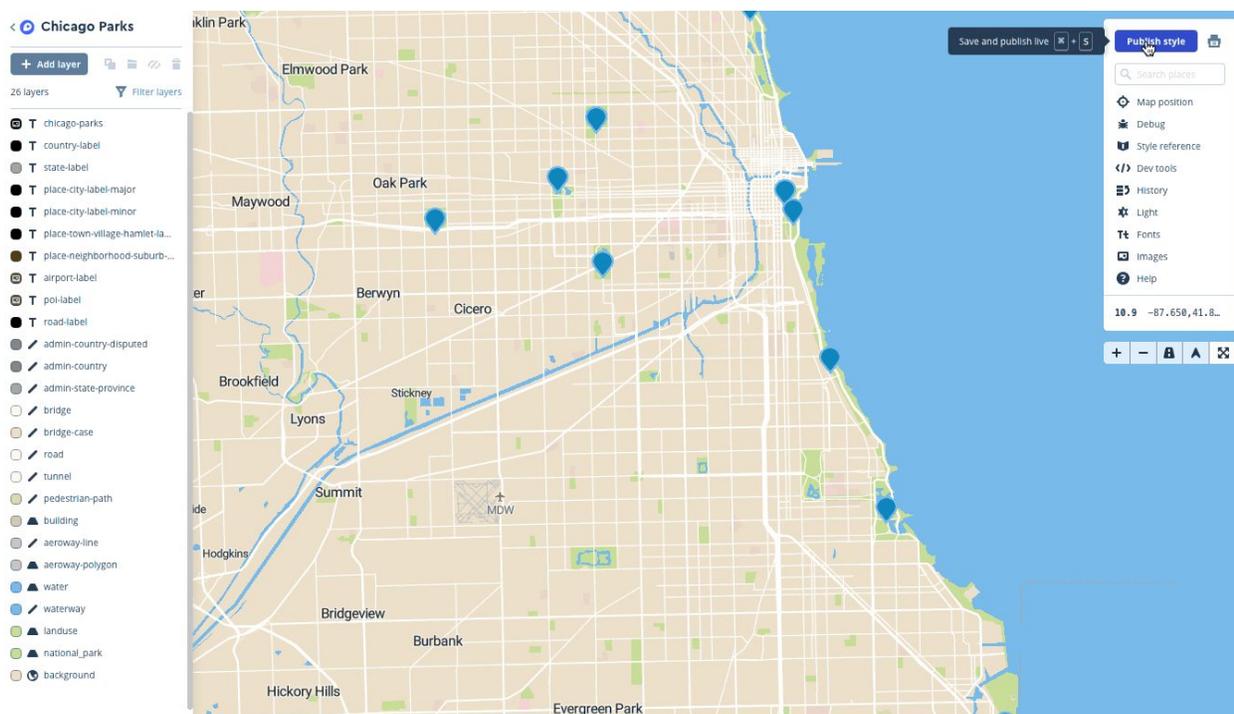
You should now see the marker icon appear on all of your points.



Publish a style

Now that you have finished styling your map, you need to publish your style in order for those changes to be live on the web.

1. Click the Publish style button in the top right of the editor.
2. A window will pop up asking you to review your changes.
3. Click Publish.

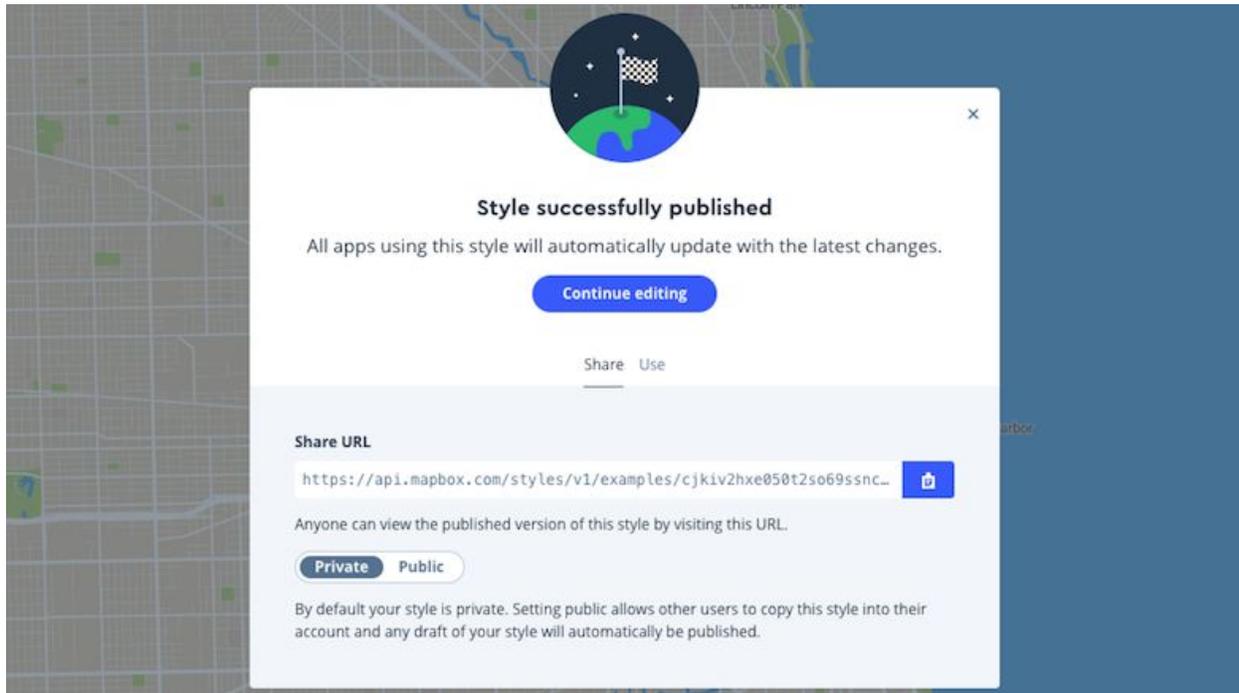


Share & use

The [Share & use modal](#) contains the resources you need to publish your style in a web app, mobile app, or with another third party tool.

Share

The Share tab contains a share URL that allows you to easily share a preview of your style with others.



More Training and Information

University of Maryland at College Park students, faculty, and staff have access to free online courses provided by ESRI. These cover a wide range of GIS topics and skills. To gain access to online classes, please contact kelleyo@umd.edu with the name of the course you are interested in and your UMD e-mail address to obtain a registration code.

Course list: <https://www.esri.com/training/unlimited-esri-training/>

University of Maryland Libraries' GIS and Spatial Data Services Center website: <http://lib.umd.edu/gis/>

University of Maryland Libraries' GIS and Spatial Data Services Center workshop series: <https://www.lib.umd.edu/gis/workshops>

Credits

1. GDAL: http://www.gis.usu.edu/~chrisg/python/2009/lectures/ospy_slides1.pdf2
2. GDAL Cheat Sheet: <https://github.com/dwtkns/gdal-cheat-sheet>
3. QGIS Wikipedia: <https://en.wikipedia.org/wiki/QGIS>
4. Open DC Data: <http://opendata.dc.gov/>
5. QGIS hexagonal grid- Ryan Ruthart's blog:
<http://ryanruthart.com/using-qgis-to-create-a-hex-bin-map-and-aggregate-data/>
6. MDN HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML>
7. Javascript Wikipedia: <https://en.wikipedia.org/wiki/JavaScript>
8. Three.js Wikipedia: <https://en.wikipedia.org/wiki/Three.js>
9. Qgis2threejs Documentation : <http://qgis2threejs.readthedocs.io/en/docs-release/Tutorial.html>
10. Map Legend: https://github.com/jakobzhao/GeovisualAnalytics/blob/master/wk03/wk03_1_lec05/readme.md
11. QGIS tutorial and tips: <http://www.qgistutorials.com/en/>
12. Atom Text Editor: <https://atom.io/>
13. CSS Wikipedia: https://en.wikipedia.org/wiki/Cascading_Style_Sheets
14. Github Pages: <https://pages.github.com/>
15. Carto Wikipedia: <https://en.wikipedia.org/wiki/CartoDB>
16. CartoDB Plugins: <https://github.com/gkudos/qgis-cartodb/wiki/Connect-your-CartoDB-Account-to-QGIS>
17. Mapbox About: <https://www.mapbox.com/about/>
18. Qgis2web : http://www.qgistutorials.com/en/docs/web_mapping_with_qgis2web.html
19. Leaflet: <http://leafletjs.com/>
20. Leaflet Choropleth Map: <http://leafletjs.com/examples/choropleth/>