The Journal of Open Source Software

# GeoHexViz: A Python package for the visualization of hexagonally binned geospatial data

**Mark Rempel***

**DOI:** 10.21105/joss.0XXXX

**Software**

- Review
- Repository
- Archive

**Editor:** Editor Name

**Submitted:** 01 January XXXX
**Published:** 01 January XXXX



*[JSON]: JavaScript Object Notation

## Summary

Geospatial visualization is an important communication method that is often used in military operations research to convey analyses to both analysts and decision makers. For example: to help commanders coordinate units over a large geographic region (Feibush et al., 2000), to depict how different terrain impacts the performance of vehicles (Laskey et al., 2010); and to inform training decisions in order to meet mission requirements (Goodrich et al., 2019). When these analyses include a large amount of point-like data, binning—in particular, hexagonal binning—may be used to summarize the data and subsequently produce an effective visualization, such as communicating risk military operations (Hunter et al., 2021) or in the context of public safety and COVID-19 (Shaito & Elmasri, 2021). However, creating such visualizations may be difficult for many since it requires in-depth knowledge of both Geographic Information Systems and analytical techniques, not to mention access to software that may require a paid license, training, and perhaps knowledge of a programming language. Open source software that provides a simple interface, requires minimal in-depth knowledge, and either limited or no programming will allow a wider range of operations researchers to produce high-quality visualizations—ultimately, leading to better informed decisions.

GeoHexViz is accessible at **FILL ME IN** and is installed via a `setup.py` script.

## Statement of need

Creating geospatial visualizations is often time-consuming and laborious (Vartak et al., 2014). This is due to that in-depth knowledge of Geographic Information System (GIS) concepts is required to use GIS software, and hence to create geospatial visualizations (Bertazzon, 2013). For example, an individual must decide which map projection to use, the colour scheme, the basemap, and in some cases how to organize the data in layers. There are many software applications that may be used to create geospatial visualizations, such as ArcGIS (Dangermond & Dangermond, 2021), QGIS (Sherman, 2021), and D3 (Bostock, 2021). ArcGIS provides a wide range of capabilities, but requires a paid license and a solid foundation in geospatial information processing (GISGeography, 2021). In contrast, QGIS is free and open-source, but also requires an in-depth knowledge of geospatial information processing to be used effectively (GrindGIS, 2021). As an alternative, programming-based approaches to create visualizations, such as D3 (Bostock, 2021) and Plotly (Smith et al., 2020b), have been developed in the last decade. In addition to an understanding of geospatial concepts, they also requires a knowledge of JavaScript and Python respectively. Common across these applications is the requirement to have knowledge of geospatial concepts, and acquiring this knowledge has been identified as a significant challenge (Sipe & Dale, 2003).

*co-first author

In addition to being time consuming to create, geospatial visualizations often require analysts to have specialized knowledge of analytic techniques. One of these techniques is binning in which a grid is placed over a data set and the individual data points are grouped by grid cell (Briney & Newcomb, 2017). This method is used when it is difficult to visualize geospatial point-like data sets; in particular, when the number of points is large, they become difficult to distinguish Briney & Newcomb (2017). In order to provide an accurate representation of the binned data, an analyst must choose a versatile grid type. There are many grid types available, such as circular, rectangular, and hexagonal. A circular grid is optimal for analysis purposes since circles are accurate for sampling, but does not provide a continuous grid (Sinha, 2019). Rectangular grids are simple to implement; however, may not be suitable when investigating connectivity or movement (Birch et al., 2007). A hexagonal grid is often selected because its more visually appealing than other grid types (Battersby et al., 2017), and shares many of its properties with a circular grid (Sinha, 2019). In addition, hexagonal grids provide many advantages including: hexagons have the same number of neighbours as they does edges; the center of each hexagon is equidistant from the centers of its neighbours (which helps when analyzing connectivity or movement); and hexagons tile densely on curved surfaces, resulting in lower edge effects (reducing analytic bias) (Sinha, 2019). The previously mentioned GIS systems provide functionality to perform hexagonal binning, albeit access to this functionality is often limited due to the issues described above.

With this in mind, GeoHexViz aims to reduce the time and in-depth knowledge required to produce publication-quality geospatial visualizations that use hexagonal binning. GeoHexViz, which is built on top of several underlying Python packages, allows an analyst to produce a publication-quality visualization in two ways. First, a user may generate a visualization via running a pre-existing command-line script whose input is a single JSON that defines the properties of the visualization. Second, a user may generate a visualization by writing a Python script that imports and invokes functions on objects found in the GeoHexViz's Python modules. Both methods require that the user provide only two arguments. The first argument is a reference to the data, which is a file path or may be a DataFrame (McKinney, 2021) or GeoDataFrame (Jordahl, 2021}) when using the second option. The second argument is a reference to the columns within the data that define the latitudes, longitudes, and the value associated with each. If no value column is present, the default of each data entry is set to one.

## Features

The aim of GeoHexViz is to simplify the production of publication-quality geospatial visualizations that utilize hexagonal binning. To do this, a user specifies a set of *layers*—where each layer is defined as a "[group] of point, line, or area (polygon) features representing a particular class or type of real-world entities' ' (Caliper, 2021)—to be visualized. At a minimum, the user must specify one layer, the *hexbin layer*, through two arguments—a reference to the point-like data to be hexagonally binned, and references to the columns containing latitudes, longitudes, and value at each coordinate.

If the output visualization is not satisfactory, GeoHexViz allows a user to adjust features of the plot. These features include:

- **scale**: the data displayed in the visualization may be on a linear (default) or logarithmic scale;
- **colour scale**: the colour scale of the visualization may be continuous (default) or discrete;
- **focus**: the visualization may have no focal point (default), showing a view of the whole Earth, or may be focused on the data; and

- **filtering**: all of the data may be present in the visualization (default) or may be clipped to a geographic region.

In addition, a user can change other properties of the visualization, such as border colour, land colour, sea colour, and figure size. In this case, these properties are passed by GeoHexViz to Plotly.

GeoHexViz may be used to create a visualization in two ways. First, the user can use GeoHexViz's command-line script `GeoHexSimple` to read a JSON file that contains properties for the visualization. The purpose of the command-line script is to give non-technical users a simple interface. Second, the user can generate a visualization via importing and invoking functions found in the GeoHexViz's Python modules. When using the second method, the reference to the data may be a DataFrame or GeoDataFrame object. If the input reference is a GeoDataFrame, the package does not *need* latitude or longitude columns. Rather, the input to the software will be the entries within the geometry column (it is up to the user to ensure that these are valid geometry types).

The hexagonal tiling is generated with the use of the Uber H3 library (Smith et al., 2020a), which provides an interface to convert conventional lat/long coordinates into a geospatial index. These hex-tiles are then stored within GeoPandas objects and are binned by common hex tile (Jordahl, 2021). The data is then converted into GeoJSON format and is visualized with the aid of the Plotly graphing library (Smith et al., 2020b).

The resulting output is a publication-quality visualization, that can be displayed, or output to a file. GeoHexViz can generate visualizations for both quantitative, and qualitative data sets.

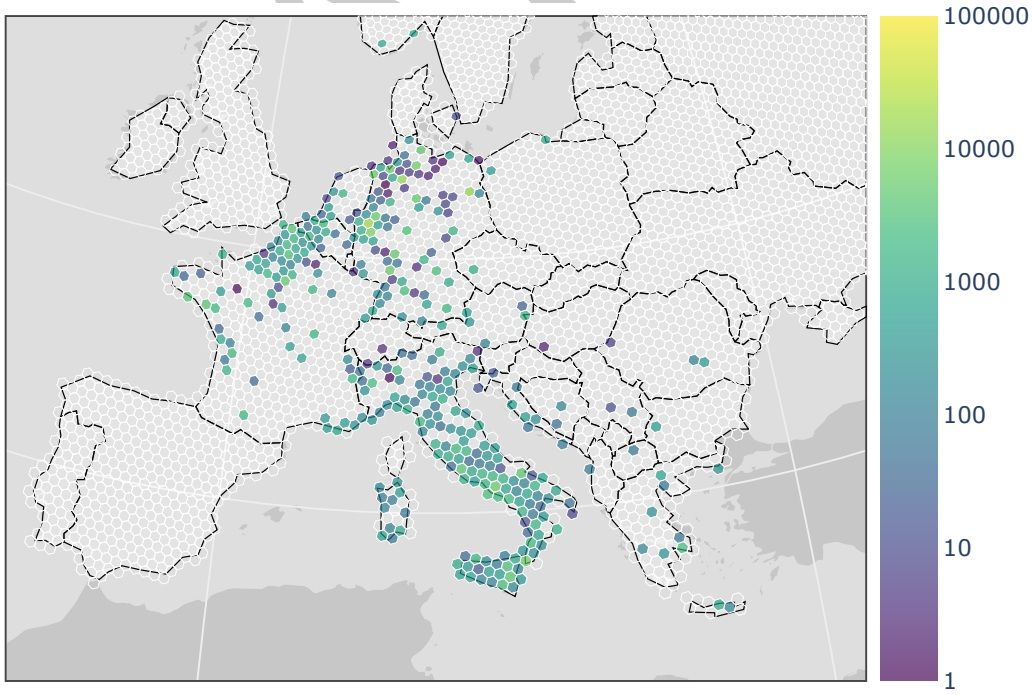


**Figure 1:** Bombings in World War 2—European Theatre; Total mass of bombs dropped in tons (1943)

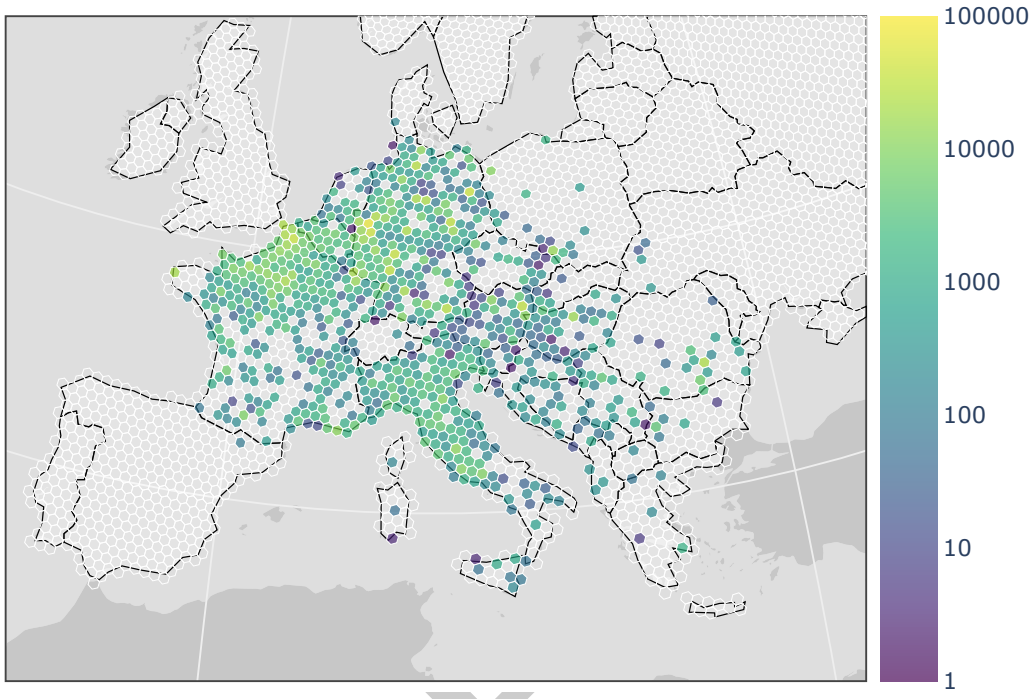


**Figure 2:** Bombings in World War 2—European Theatre; Total mass of bombs dropped in tons (1944)

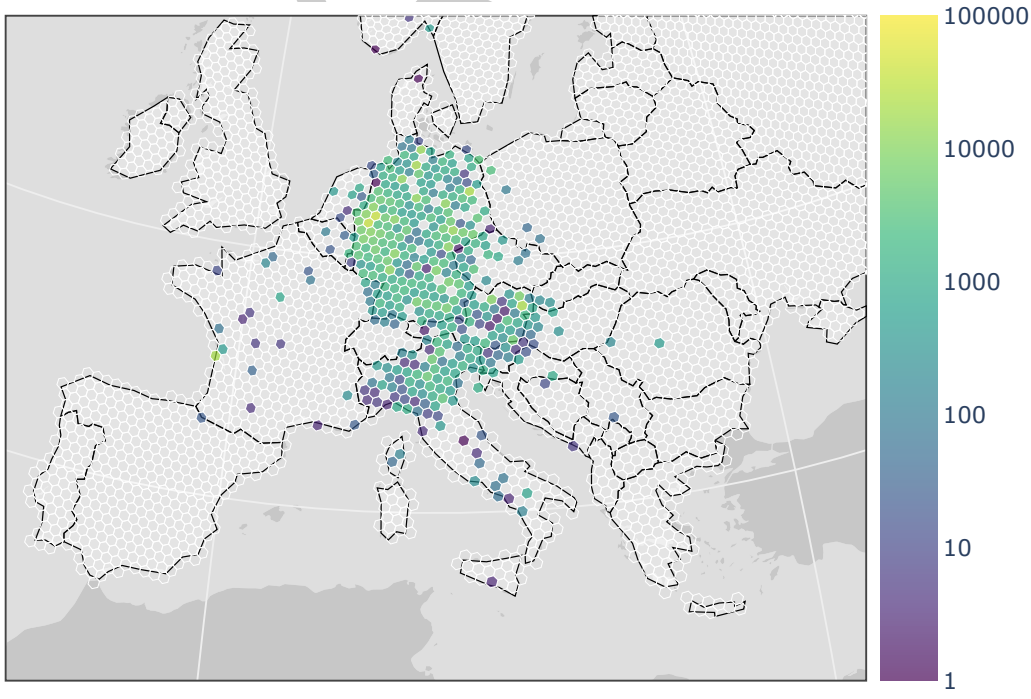


**Figure 3:** Bombings in World War 2—European Theatre; Total mass of bombs dropped in tons (1945)

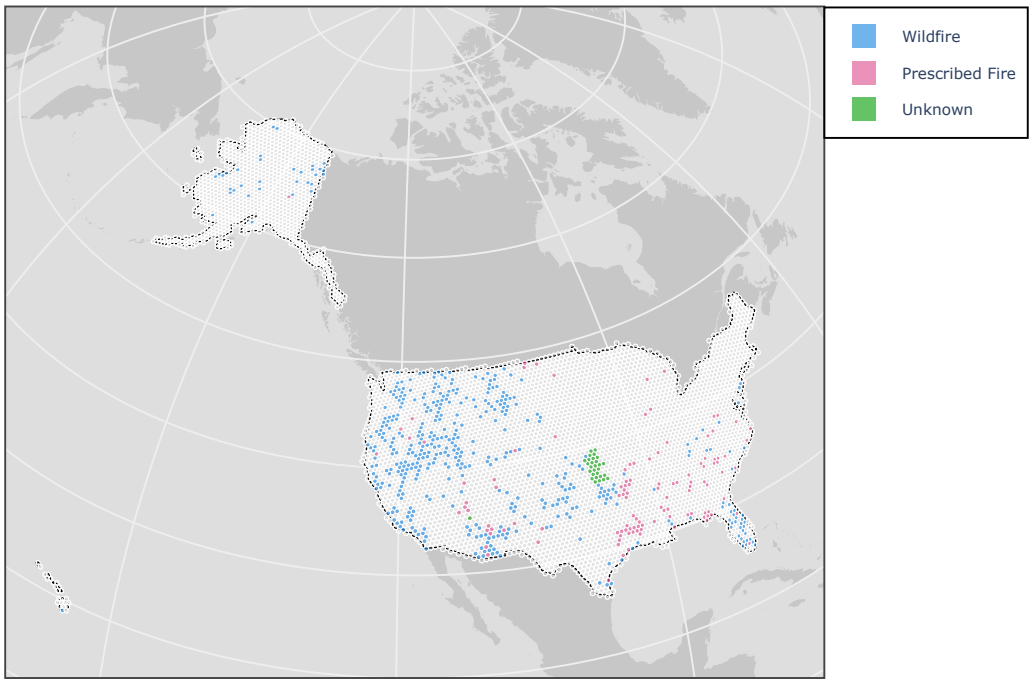


**Figure 4:** Most frequent fire category by location (United States of America: 2017)

## Saving the output

Given a JSON file that defines the hexbin layer and optional layers, such as regions, grids, and outlines, GeoHexViz outputs a geospatial visualization with hexagonally binned data. The visualization may be saved in a variety of formats, including PDF, PNG, JPEG, WEBP, SVG, and EPS formats.

# Limitations

This package uses GeoJSON format to plot data sets. With GeoJSON comes difficulties when geometries cross the 180th meridian (MacWright, n.d.). The issue appears to cause a color that bleeds through the entire plot and leaves a hexagon empty. In the final plot, this issue may or may not appear as it only occurs at certain angles of rotation. In this package a simple solution to the problem is implemented, in the future it would be best to provide a more robust solution. The solution that is used works generally, however, when hexagons containing either the north or south pole are present, the solution to the 180th meridian issue persists. This pole issue can be seen in **??**.

There also exists some issues with the generation of discrete color scales under rare circumstances. These circumstances include generating discrete color scales with not enough hues to fill the scale, and generating diverging discrete colorscales with the center hue in a weird position. These issues have been noted and will be fixed in the near future.

There exists issues with the positioning and height of the color bar with respect to the plot area of the figure. Although the user is capable of altering the dimensions and positioning of the color bar, this should be done automatically as it is a common feature of publication quality choropleth maps.

The Journal of Open Source Software

## Acknowledgements

Thank you to Nicholi Shiell for his input in testing, and providing advice for the development of this package and of its supporting documents.

## References

Battersby, S. E., Strebe, D., & Finn, M. P. (2017). Shapes on a plane: Evaluating the impact of projection distortion on spatial binning. *Cartography and Geographic Information Science*, *44*(5), 410–421. https://doi.org/10.1080/15230406.2016.1180263

Bertazzon, S. (2013). Rethinking GIS teaching to bridge the gap between technical skills and geographic knowledge. *Journal of Research and Didactics in Geography*, *1*, 67–72.

Birch, C., Oom, S., & Beecham, J. (2007). Rectangular and hexagonal grids used for observation, experiment and simulation in ecology. *Ecological Modelling*, *206*, 347–359. https://doi.org/10.1016/j.ecolmodel.2007.03.041

Bostock, M. (2021). *D3.js - data-driven documents*. https://d3js.org/

Briney, A., & Newcomb, D. (2017). Binning in GIS. In *GIS Lounge*. https://www.gislounge.com/binning-gis/

Caliper. (2021). *What is a layer?* Caliper Mapping; Transportation Glossary. https://www.caliper.com/glossary/what-is-a-map-layer.htm

Dangermond, J., & Dangermond, L. (2021). *ArcGIS online*. https://www.arcgis.com/

Feibush, E., Gagvani, N., & Williams, D. (2000). Visualization for situational awareness. *IEEE Computer Graphics and Applications*, *20*(5), 38–45. https://doi.org/10.1109/38.865878

Field, K. (2012). *Using a binning technique for point-based multiscale web maps*. ArcGIS Online. https://www.esri.com/arcgis-blog/products/arcgis-online/mapping/using-a-binning-technique-for-point-based-multiscale-web-maps/

GISGeography. (2021). *ArcGIS review: Is ArcMap the best GIS software?* https://gisgeography.com/esri-arcgis-software-review-guide/

Goodrich, D. C., Heilman, P., Guertin, D., Levick, L. R., Burns, I., Armendariz, G., & Wei, H. (2019). *Automated geospatial watershed assessment (AGWA) to aid in sustaining military mission and training*. USDA-ARS Southwest Watershed Research Center (SWRC) Tucson United States.

GrindGIS. (2021). *Pros and cons of QGIS*. https://grindgis.com/software/pros-and-cons-of-qgis

Hunter, G., Chan, J., & Rempel, M. (2021). *Assessing the impact of infrastructure on arctic operations* (Scientific Report DRDC-RDDC-2021-R024). Defence Research; Development Canada.

Jordahl, K. (2021). *GeoPandas (0.9.0)*. https://geopandas.org/index.html

Laskey, K. B., Wright, E. J., & Paulo C.G., da C. (2010). Envisioning uncertainty in geospatial information. *International Journal of Approximate Reasoning*, *51*(2), 209–223. https://doi.org/https://doi.org/10.1016/j.ijar.2009.05.011

MacWright, T. (n.d.). The 180th meridian. In *macwright.com*. https://macwright.com/2016/09/26/the-180th-meridian.html

McKinney, W. (2021). *Pandas.DataFrame*. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

Shaito, M., & Elmasri, R. (2021). Map visualization using spatial and spatio-temporal data: Application to COVID-19 data. *The 14th PErvasive Technologies Related to Assistive Environments Conference*, 284--291. https://doi.org/10.1145/3453892.3461336

Sherman, G. (2021). *QGIS - a free and open source geographic information system*. https://qgis.org/en/site/

Sinha, A. (2019). *Spatial modelling tidbits: Honeycomb or fishnets?* Towards Data Science. https://towardsdatascience.com/spatial-modelling-tidbits-honeycomb-or-fishnets-7f0b19273aab

Sipe, N., & Dale, P. (2003). Challenges in using geographic information systems (GIS) to understand and control malaria in indonesia. *Malaria Journal*, *4*(1). https://doi.org/10.1186/1475-2875-2-36

Smith, A. M., Thaney, K., & Hahnel, M. (2020a). H3: A hexagonal hierarchical geospatial indexing system. In *GitHub repository*. GitHub. https://github.com/uber/h3

Smith, A. M., Thaney, K., & Hahnel, M. (2020b). Plotly.py: The interactive graphing library for python. In *GitHub repository*. GitHub. https://github.com/plotly/plotly.py

Vartak, M., Madden, S., Parameswaran, A., & Polyzotis, N. (2014). *SeeDB: Automatically generating query visualizations*. *7*(13), 1581--1584. https://doi.org/10.14778/2733004.2733035