Density Heatmap in Python

How to make a density heatmap in Python with Plotly.

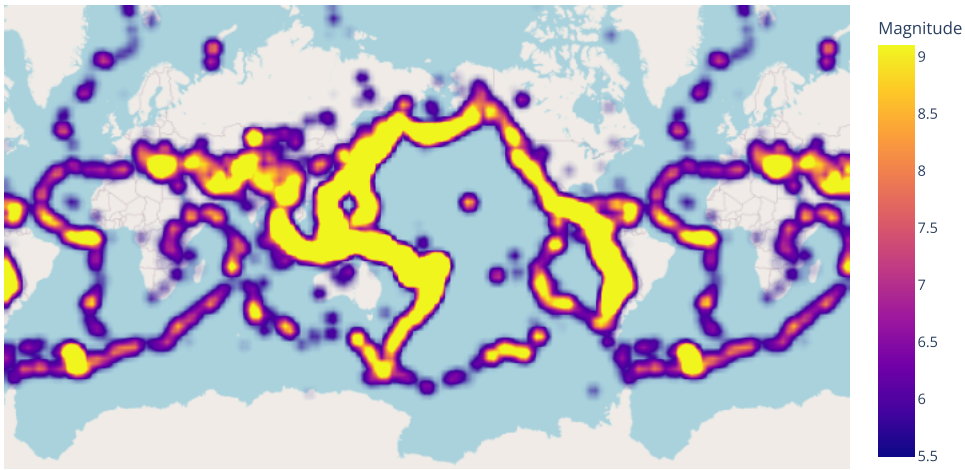
Density map with plotly.express

Plotly Express (/python/plotly-express/) is the easy-to-use, high-level interface to Plotly, which operates on a variety of types of data (/python/px-arguments/) and produces easy-to-style figures (/python/styling-plotly-express/).

With px.density_map, each row of the DataFrame is represented as a point smoothed with a given radius of influence.

```
import pandas as pd
df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/earthquakes-23k.csv')

import plotly.express as px
fig = px.density_map(df, lat='Latitude', lon='Longitude', z='Magnitude', radius=10,
                        center=dict(lat=0, lon=180), zoom=0,
                        map_style="open-street-map")
fig.show()
```



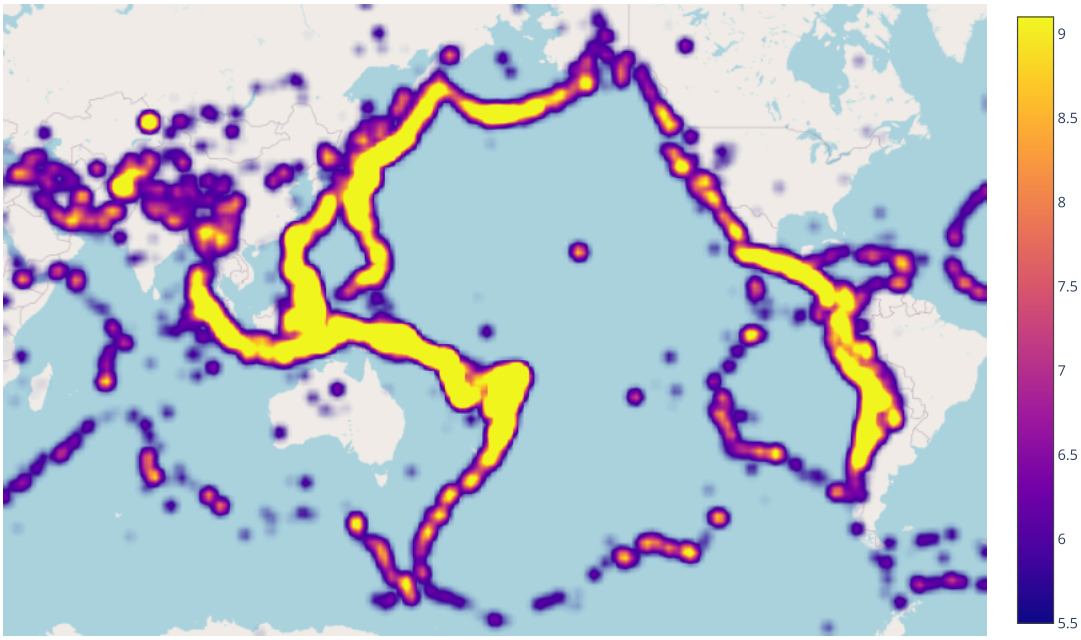
© OpenStreetMap (https://www.openstreetmap.org/copyright) contributors

Density map with plotly.graph_objects

If Plotly Express does not provide a good starting point, it is also possible to use the more generic go.Densitymap class from plotly.graph_objects (/python/graph-objects/).

```
import pandas as pd
quakes = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/earthquakes-23k.csv')

import plotly.graph_objects as go
fig = go.Figure(go.Densitymap(lat=quakes.Latitude, lon=quakes.Longitude, z=quakes.Magnitude,
                              radius=10))
fig.update_layout(map_style="open-street-map", map_center_lon=180)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



Mapbox Maps

> Mapbox traces are deprecated and may be removed in a future version of Plotly.py.

The earlier examples using `px.density_map` and `go.Densitymap` use Maplibre (https://maplibre.org/maplibre-gl-js/docs/) for rendering. These traces were introduced in Plotly.py 5.24. These trace types are now the recommended way to make tile-based density heatmaps. There are also traces that use Mapbox (https://docs.mapbox.com): density_mapbox and go.Densitymapbox.

To use these trace types, in some cases you *may* need a Mapbox account and a public Mapbox Access Token (https://www.mapbox.com/studio). See our Mapbox Map Layers (/python/mapbox-layers/) documentation for more information.

Here's one of the earlier examples rewritten to use px.density_mapbox.

```
import pandas as pd
df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/earthquakes-23k.csv')

import plotly.express as px
fig = px.density_mapbox(df, lat='Latitude', lon='Longitude', z='Magnitude', radius=10,
                        center=dict(lat=0, lon=180), zoom=0,
                        mapbox_style="open-street-map")
fig.show()
```

Stamen Terrain base map with Mapbox (Stadia Maps token needed): density heatmap with plotly.express

Some base maps require a token. To use "stamen" base maps, you'll need a Stadia Maps (https://www.stadiamaps.com) token, which you can provide to the mapbox_accesstoken parameter on fig.update_layout. Here, we have the token saved in a file called .mapbox_token, load it in to the variable token, and then pass it to mapbox_accesstoken.

```
import plotly.express as px
import pandas as pd

token = open(".mapbox_token").read() # you will need your own token

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/earthquakes-23k.csv')

fig = px.density_mapbox(df, lat='Latitude', lon='Longitude', z='Magnitude', radius=10,
                        center=dict(lat=0, lon=180), zoom=0,
                        map_style="stamen-terrain")
fig.update_layout(mapbox_accesstoken=token)
fig.show()
```

Reference

See function reference for px.(density_map) (https://plotly.com/python-api-reference/generated/plotly.express.density_mapbox) or https://plotly.com/python/reference/densitymap/ (https://plotly.com/python/reference/densitymap/) for available attribute options.

For Mapbox-based maps, see function reference for px.(density_mapbox) (https://plotly.com/python-api-reference/generated/plotly.express.density_mapbox) or https://plotly.com/python/reference/densitymapbox/ (https://plotly.com/python/reference/densitymapbox/).

What About Dash?

Dash (https://dash.plot.ly/) is an open-source framework for building analytical applications, with no Javascript required, and it is tightly integrated with the Plotly graphing library.

Learn about how to install Dash at https://dash.plot.ly/installation (https://dash.plot.ly/installation).

Everywhere in this page that you see fig.show(), you can display the same figure in a Dash application by passing it to the figure argument of the Graph component (https://dash.plot.ly/dash-core-components/graph) from the built-in dash_core_components package like this:

```
import plotly.graph_objects as go # or plotly.express as px
fig = go.Figure() # or any Plotly Express function e.g. px.bar(...)
# fig.add_trace( ... )
# fig.update_layout( ... )

from dash import Dash, dcc, html

app = Dash()
app.layout = html.Div([
    dcc.Graph(figure=fig)
])

app.run(debug=True, use_reloader=False)  # Turn off reloader if inside Jupyter
```

ts

box

y

JOIN OUR MAILING LIST

Sign up to stay in the loop with all things Plotly — from Dash Club to product updates, webinars, and more!

SUBSCRIBE (HTTPS://GO.PLOT.LY/SUBSCRIPTION)

Products

Dash (https://plotly.com/dash/)
Consulting and Training (https://plotly.com/consulting-and-oem/)

Pricing

Enterprise Pricing (https://plotly.com/get-pricing/)

About Us

Careers (https://plotly.com/careers)
Resources (https://plotly.com/resources/)
Blog (https://medium.com/@plotlygraphs)

Support

Community Support (https://community.plot.ly/)
Documentation (https://plotly.com/graphing-libraries)

Terms of Service (https://community.plotly.com/tos) Privacy Policy (https://plotly.com/privacy/)