

[plotly | Graphing Libraries](https://plotly.com/graphing-libraries/) (<https://plotly.com/>) / graphing-libraries/  
 ?utm\_medium=graphing\_libraries&utm\_campaign=studio\_cloud\_launch&utm\_content=sidebar

 Python (/python) > Maps (/python/maps) > Scatter Plots on Maps Suggest an edit to this page (<https://github.com/plotly/plotly.py/edit/doc-prod/doc/python/scatter-plots-on-maps.md>)

## Scatter Plots on Maps in Python

How to make scatter plots on maps in Python. Scatter plots on maps highlight geographic areas and can be colored by value.

attergeo

Plotly Studio: Transform any dataset into an interactive data application in minutes with AI. [Sign up for early access now.](https://plotly.com/studio/?utm_medium=graphing_libraries&utm_campaign=studio_early_access&utm_content=sidebar) ([https://plotly.com/studio/?utm\\_medium=graphing\\_libraries&utm\\_campaign=studio\\_early\\_access&utm\\_content=sidebar](https://plotly.com/studio/?utm_medium=graphing_libraries&utm_campaign=studio_early_access&utm_content=sidebar))

### Base Map Configuration

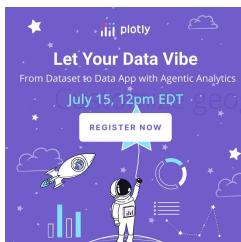
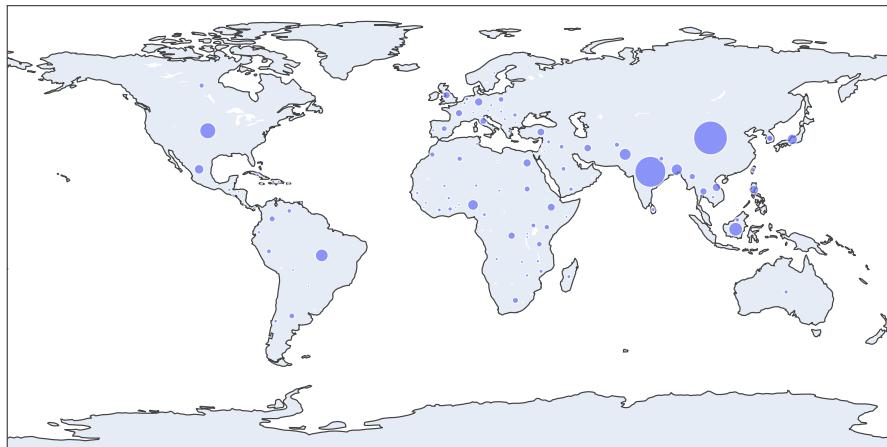
Plotly figures made with [Plotly Express \(/python/plotly-express/\)](#) px.scatter\_geo, px.line\_geo or px.choropleth functions or containing go.Choropleth or go.Scattergeo graph objects ([/python/graph-objects/](#)) have a go.layout.Geo object which can be used to [control the appearance of the base map \(/python/map-configuration/\)](#) onto which data is plotted.

### Geographical Scatter Plot with px.scatter\_geo

Here we show the [Plotly Express \(/python/plotly-express/\)](#) function px.scatter\_geo for a geographical scatter plot. The size argument is used to set the size of markers from a given column of the DataFrame.

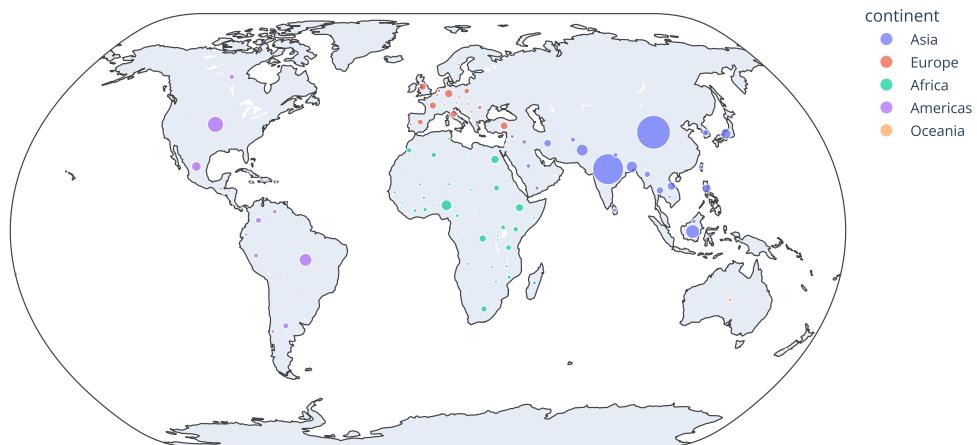
[Plotly Express \(/python/plotly-express/\)](#) is the easy-to-use, high-level interface to Plotly, which [operates on a variety of types of data \(/python/px-arguments/\)](#) and produces [easy-to-style figures \(/python/styling:plotly-express/\)](#).

```
import plotly.express as px
df = px.data.gapminder().query("year == 2007")
fig = px.scatter_geo(df, locations="iso_alpha",
                      size="pop", # size of markers, "pop" is one of the columns of gapminder
                      )
fig.show()
```



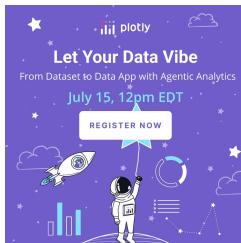
```
import plotly.express as px
df = px.data.gapminder().query("year == 2007")
fig = px.scatter_geo(df, locations="iso_alpha",
                      color="continent", # which column to use to set the color of markers
                      hover_name="country", # column added to hover information
                      size="pop", # size of markers
                      projection="natural earth")
fig.show()
```

atthergeo



## Basic Example with GeoPandas

px.scatter\_geo can work well with [GeoPandas](https://geopandas.org/) (<https://geopandas.org/>) dataframes whose geometry is of type Point.

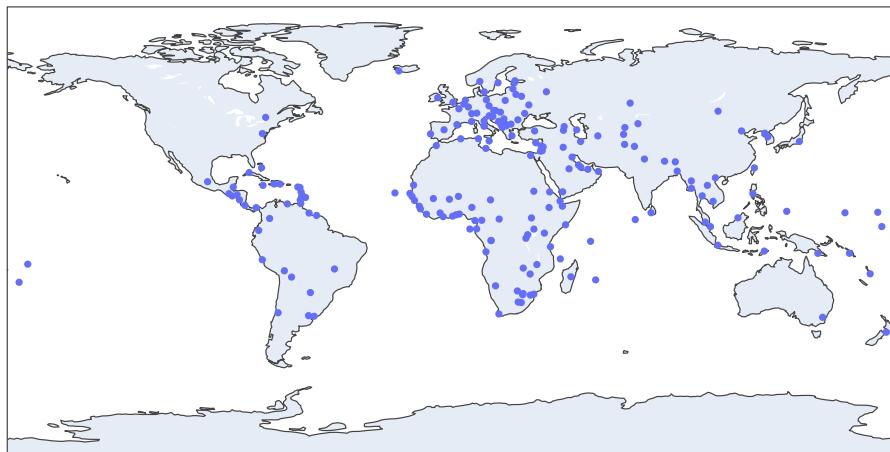


attergeo

```
import plotly.express as px
import geopandas as gpd

geo_df = gpd.read_file(gpd.datasets.get_path('naturalearth_cities'))

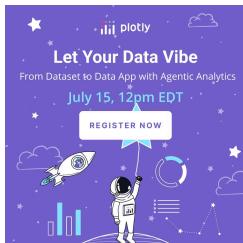
px.set_mapbox_access_token(open(".mapbox_token").read())
fig = px.scatter_geo(geo_df,
                      lat=geo_df.geometry.y,
                      lon=geo_df.geometry.x,
                      hover_name="name")
fig.show()
```



## U.S. Airports Map

Here we show how to use go.Scattergeo from `plotly.graph_objects`.

Simple U.S. Airports Map



```

import plotly.graph_objects as go

import pandas as pd

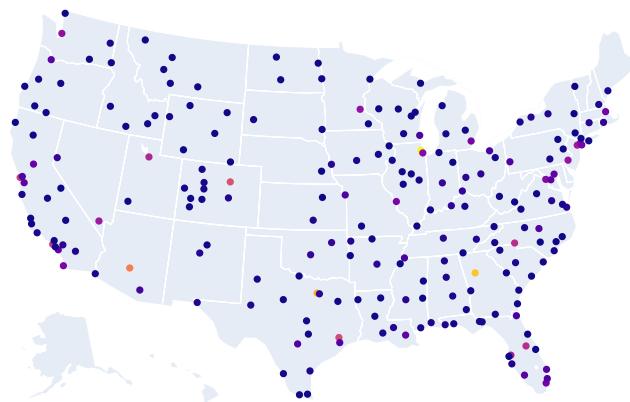
df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/2011_february_us_airport_traffic.csv')
df['text'] = df['airport'] + '' + df['city'] + ', ' + df['state'] + '' + 'Arrivals: ' + df['cnt'].astype(str)

fig = go.Figure(data=go.Scattergeo(
    lon = df['long'],
    lat = df['lat'],
    text = df['text'],
    mode = 'markers',
    marker_color = df['cnt'],
))
attergeo
:

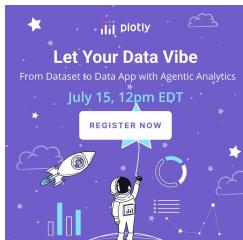
fig.update_layout(
    title = 'Most trafficked US airports<br>(Hover for airport names)',
    geo_scope='usa',
)
fig.show()

```

Most trafficked US airports  
(Hover for airport names)



Styled U.S. Airports Map



```

import plotly.graph_objects as go

import pandas as pd

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/2011_february_us_airport_traffic.csv')
df['text'] = df['airport'] + '' + df['city'] + ', ' + df['state'] + '' + 'Arrivals: ' + df['cnt'].astype(str)

fig = go.Figure(data=go.Scattergeo(
    locationmode = 'USA-states',
    lon = df['long'],
    lat = df['lat'],
    text = df['text'],
    mode = 'markers',
    marker = dict(
        size = 8,
        opacity = 0.8,
        reversescale = True,
        autocolorscale = False,
        symbol = 'square',
        line = dict(
            width=1,
            color='rgba(102, 102, 102)'
        ),
        colorscale = 'Blues',
        cmin = 0,
        color = df['cnt'],
        cmax = df['cnt'].max(),
        colorbar=dict(
            title=dict(
                text="Incoming flights<br>February 2011"
            )
        )
    )))
fig.update_layout(
    title = 'Most trafficked US airports<br>(Hover for airport names)',
    geo = dict(
        scope='usa',
        projection_type='albers usa',
        showland = True,
        landcolor = "rgb(250, 250, 250)",
        subunitcolor = "rgb(217, 217, 217)",
        countrycolor = "rgb(217, 217, 217)",
        countrywidth = 0.5,
        subunitwidth = 0.5
    ),
)
fig.show()

```

Most trafficked US airports  
(Hover for airport names)



## North American Precipitation Map

```

import plotly.graph_objects as go

import pandas as pd

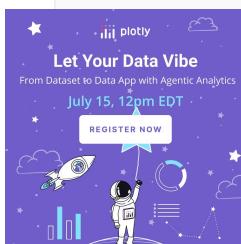
df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/2015_06_30_precipitation.csv')

atthergo
: scl = [0,"rgb(150,0,90)"],[0.125,"rgb(0, 0, 200)"],[0.25,"rgb(0, 25, 255)"],\[0.375,"rgb(0, 152, 255)"],[0.5,"rgb(44, 255, 150)"],[0.625,"rgb(151, 255, 0)"],\[0.75,"rgb(255, 234, 0)"],[0.875,"rgb(255, 111, 0)"],[1,"rgb(255, 0, 0)"]

fig = go.Figure(data=go.Scattergeo(
    lat = df['Lat'],
    lon = df['Lon'],
    text = df['Globvalue'].astype(str) + ' inches',
    marker = dict(
        color = df['Globvalue'],
        colorscale = scl,
        reversescale = True,
        opacity = 0.7,
        size = 2,
        colorbar = dict(
            title = dict(
                side="right"
            ),
            outlinecolor = "rgba(68, 68, 68, 0)",
            ticks = "outside",
            showticksuffix = "last",
            dtick = 0.1
        )
    )
))

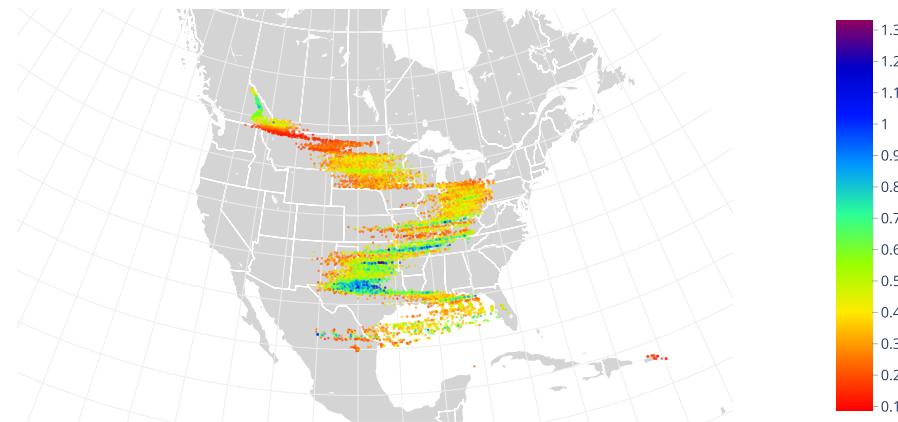
fig.update_layout(
    geo = dict(
        scope = 'north america',
        showland = True,
        landcolor = "rgb(212, 212, 212)",
        subunitcolor = "rgb(255, 255, 255)",
        countrycolor = "rgb(255, 255, 255)",
        showlakes = True,
        lakecolor = "rgb(255, 255, 255)",
        showsubunits = True,
        showcountries = True,
        resolution = 50,
        projection = dict(
            type = 'conic conformal',
            rotation_lon = -100
        ),
        lonaxis = dict(
            showgrid = True,
            gridwidth = 0.5,
            range= [ -140.0, -55.0 ],
            dtick = 5
        ),
        lataxis = dict (
            showgrid = True,
            gridwidth = 0.5,
            range= [ 20.0, 60.0 ],
            dtick = 5
        )
    ),
    title=dict(text='US Precipitation 06-30-2015<br>Source: <a href="http://water.weather.gov/precip/">NOAA</a>'),
)
fig.show()

```



US Precipitation 06-30-2015  
Source: NOAA

attergeo



## Reference

See [function reference for px.scatter\\_geo\(\)](https://plotly.com/python-api-reference/generated/plotly.express.scatter_geo.html) ([https://plotly.com/python-api-reference/generated/plotly.express.scatter\\_geo.html](https://plotly.com/python-api-reference/generated/plotly.express.scatter_geo.html)) or <https://plotly.com/python/reference/scattergeo/> (<https://plotly.com/python/reference/scattergeo.html>) and <https://plotly.com/python/reference/layout/geo/> (<https://plotly.com/python/reference/layout/geo.html>) for more information and chart attribute options!

## What About Dash?

[Dash](https://dash.plotly.com/) (<https://dash.plotly.com/>) 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.plotly/installation.html>.

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.plotly/dash-core-components/graph.html) (<https://dash.plotly/dash-core-components/graph.html>) 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
```



# Dash your way to interactive web apps.

Let Your Data Vibe

From Dataset to Data App with Agentic Analytics

July 15, 12pm EDT

REGISTER NOW

My First App with Data, Graph, and Controls

country	pop	continent	lifeExp	gdpPerCap
Afghanistan	31889923	Asia	43.828	974.5893384
Albania	2600023	Europe	76.423	5937.029225999999
Algeria	3313321	Africa	72.301	6223.367465
Angola	12420876	Africa	42.731	4797.231267
Argentina	40391927	Americas	75.32	12779.37964
Australia	20434176	Oceania	81.235	34435.367439999995
Austria	8199783	Europe	79.829	36326.4927
Bahrain	708573	Asia	75.635	29796.0488
Bangladesh	15044831	Asia	64.062	1301.255792
Belgium	1092226	Europe	79.441	33692.6956
Benin	8078314	Africa	56.728	1441.224873
Bolivia	9191952	Americas	69.554	3822.337084

Asia
Europe
Africa
Americas
Oceania

avg lifeExp
count

**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](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**

attergeo

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>)

Copyright © 2025 Plotly. All rights reserved.

[Terms of Service](https://community.plot.ly/tos) (<https://community.plot.ly/tos>)    [Privacy Policy](https://plotly.com/privacy) (<https://plotly.com/privacy>)

