plotly | Graphing Libraries (https://plotly.com/)(/graphing-libraries/)

utm_campaign=studio_cloud_launch&utm_content=sidebar)

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

Scatter Plots on Tile Maps in Python

How to make scatter plots on tile maps in Python.

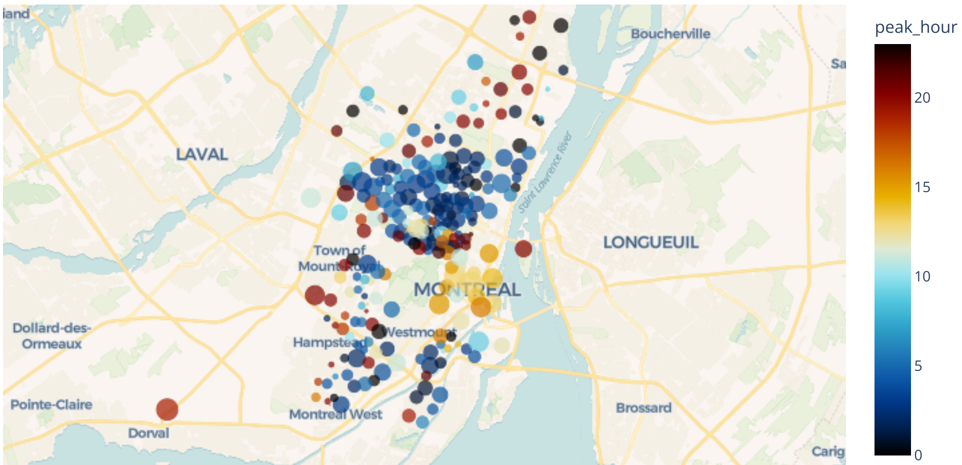
> Plotly Studio: Transform any dataset into an interactive data application in minutes with AI. Sign up for early access now. (https://plotly.com/studio/?utm_medium=graphing_libraries&utm_campaign=studio_early_access&utm_content=sidebar)

Basic example with Plotly Express

Here we show the Plotly Express (/python/plotly-express/) function px.scatter_map for a scatter plot on a tile map.

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/).

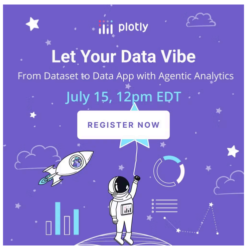
```
import plotly.express as px
df = px.data.carshare()
fig = px.scatter_map(df, lat="centroid_lat", lon="centroid_lon",     color="peak_hour", size="car_hours",
                  color_continuous_scale=px.colors.cyclical.IceFire, size_max=15, zoom=10)
fig.show()
```



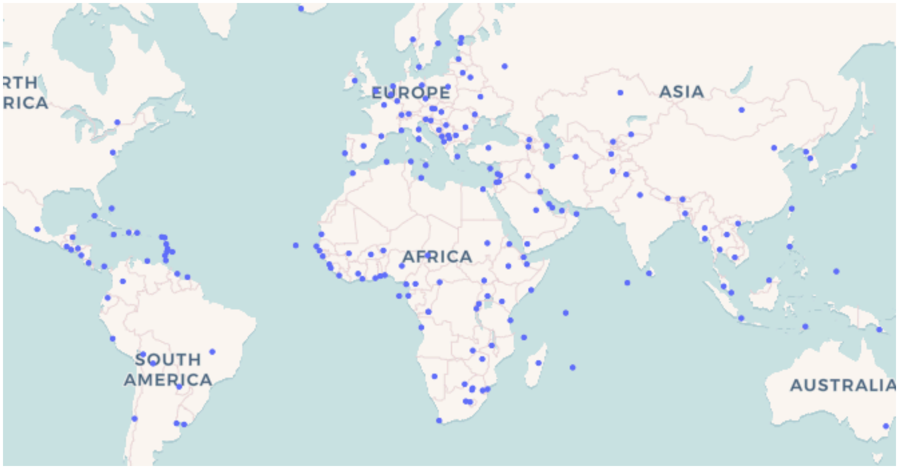
© CARTO (https://carto.com/about-carto/), © OpenStreetMap (http://www.openstreetmap.org/about/) contributors

Basic Example with GeoPandas

px.scatter_map can work well with GeoPandas (https://geopandas.org/) dataframes whose geometry is of type Point.

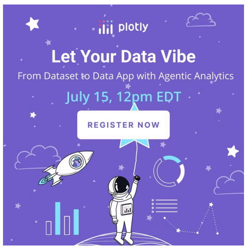


```
import plotly.express as px
import geopandas as gpd

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

fig = px.scatter_map(geo_df,
                        lat=geo_df.geometry.y,
                        lon=geo_df.geometry.x,
                        hover_name="name",
                        zoom=1)
fig.show()
```

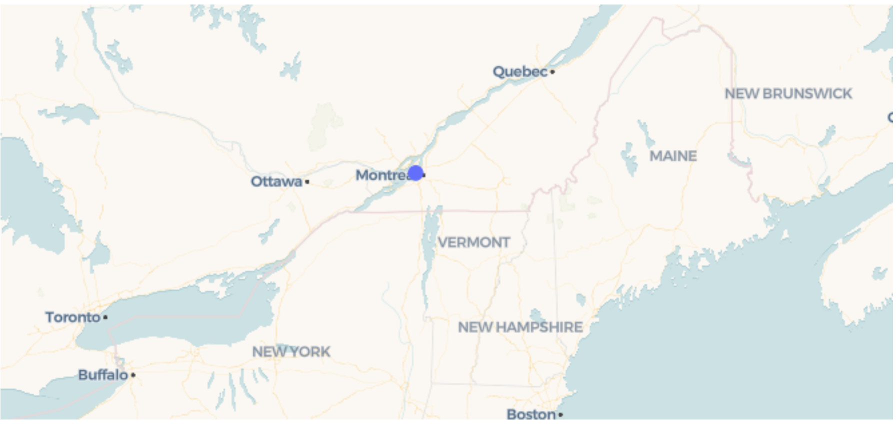


© CARTO (https://carto.com/about-carto/), © OpenStreetMap (http://www.openstreetmap.org/about/) contributors

Basic Example



```
import plotly.graph_objects as go

fig = go.Figure(go.Scattermap(
        lat=['45.5017'],
        lon=['-73.5673'],
        mode='markers',
        marker=go.scattermap.Marker(
            size=14
        ),
        text=['Montreal'],
    ))

fig.update_layout(
    hovermode='closest',
    map=dict(
        bearing=0,
        center=go.layout.map.Center(
            lat=45,
            lon=-73
        ),
        pitch=0,
        zoom=5
    )
)

fig.show()
```



© CARTO (https://carto.com/about-carto/), © OpenStreetMap (http://www.openstreetmap.org/about/) contributors

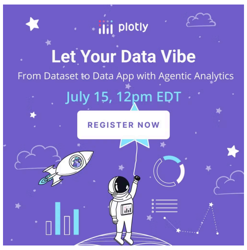
Multiple Markers

```
import plotly.graph_objects as go

fig = go.Figure(go.Scattermap(
        lat=['38.91427','38.91538','38.91458',
             '38.92239','38.93222','38.90842',
             '38.91931','38.93260','38.91368',
             '38.88516','38.921894','38.93206',
             '38.91275'],
        lon=['-77.02827','-77.02013','-77.03155',
             '-77.04227','-77.02854','-77.02419',
             '-77.02518','-77.03304','-77.04509',
             '-76.99656','-77.042438','-77.02821',
             '-77.01239'],
        mode='markers',
        marker=go.scattermap.Marker(
            size=9
        ),
        text=["The coffee bar","Bistro Bohem","Black Cat",
             "Snap","Columbia Heights Coffee","Azi's Cafe",
             "Blind Dog Cafe","Le Caprice","Filter",
             "Peregrine","Tryst","The Coupe",
             "Big Bear Cafe"],
    ))

fig.update_layout(
    autosize=True,
    hovermode='closest',
    map=dict(
        bearing=0,
        center=dict(
            lat=38.92,
            lon=-77.07
        ),
        pitch=0,
        zoom=10
    ),
)

fig.show()
```

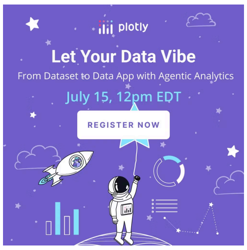
Great Falls
Adelphi
Greenbelt
BETHESDA
Takoma Park
COLLEGE PARK
Seabrook
Falls Crossing
Chillum
HYATTSVILLE
Brookmont
Bladensburg
Wolf Trap
McLean
Tysons
Vienna
Idylwood
WASHINGTON
Summerfield
Oakton
Merrifield
Falls Church
Seven Corners
Walker Mill
Largo
Bailey's Crossroads
FAIRFAX
Suitland
Forestville
Wakefield
Annandale
Hillcrest Heights

© CARTO (https://carto.com/about-carto/), © OpenStreetMap (http://www.openstreetmap.org/about/) contributors

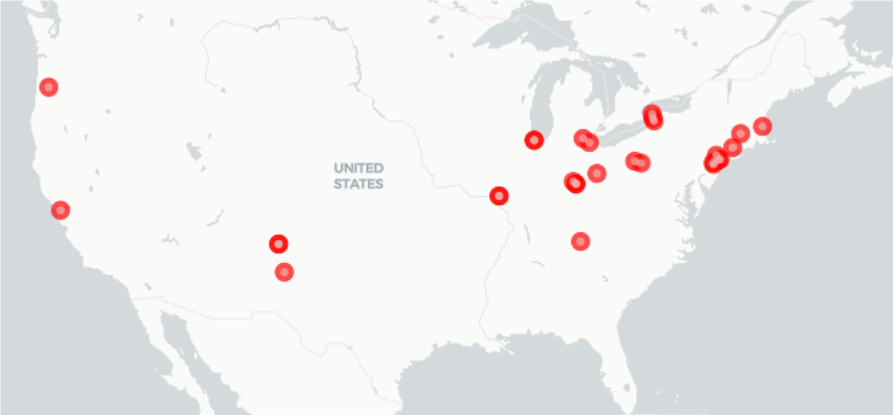
Nuclear Waste Sites on Campuses



```
import plotly.graph_objects as go
import pandas as pd

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/Nuclear%20Waste%20Sites%20on%20American%20Campuses.csv')
site_lat = df.lat
site_lon = df.lon
locations_name = df.text

fig = go.Figure()

fig.add_trace(go.Scattermap(
        lat=site_lat,
        lon=site_lon,
        mode='markers',
        marker=go.scattermap.Marker(
            size=17,
            color='rgb(255, 0, 0)',
            opacity=0.7
        ),
        text=locations_name,
        hoverinfo='text'
    ))

fig.add_trace(go.Scattermap(
        lat=site_lat,
        lon=site_lon,
        mode='markers',
        marker=go.scattermap.Marker(
            size=8,
            color='rgb(242, 177, 172)',
            opacity=0.7
        ),
        hoverinfo='none'
    ))

fig.update_layout(
    title=dict(text='Nuclear Waste Sites on Campus'),
    autosize=True,
    hovermode='closest',
    showlegend=False,
    map=dict(
        bearing=0,
        center=dict(
            lat=38,
            lon=-94
        ),
        pitch=0,
        zoom=3,
        style='light'
    ),
)

fig.show()
```



Nuclear Waste Sites on Campus

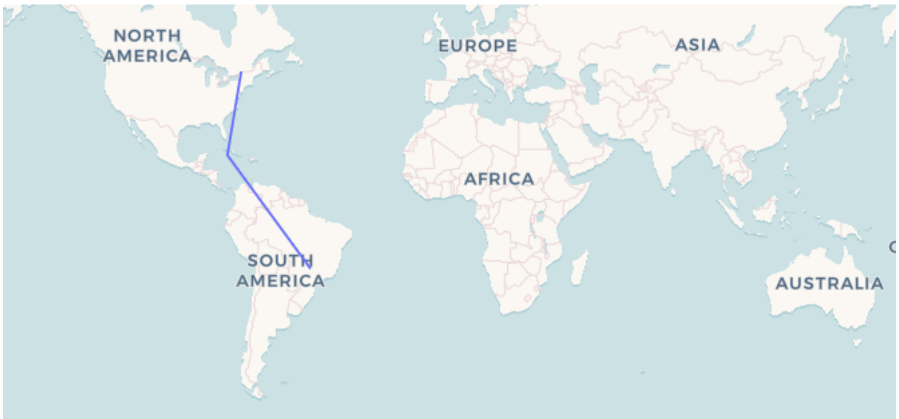


© CARTO (https://carto.com/about-carto/), © OpenStreetMap (http://www.openstreetmap.org/about/) contributors

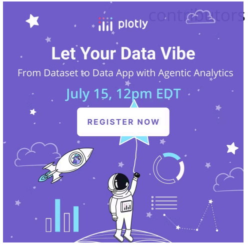
Set Marker Symbols

You can define the symbol on your map by setting symbol (https://plotly.com/python/reference/scattermap/#scattermap-marker-symbol) attribute.

```
import plotly.graph_objects as go

fig = go.Figure(go.Scattermap(
    mode = "markers+text+lines",
    lon = [-75, -80, -50], lat = [45, 20, -20],
    marker = {'size': 20, 'symbol': ["bus", "harbor", "airport"]},
    text = ["Bus", "Harbor", "airport"],textposition = "bottom right"))

fig.update_layout(
    map = {
        'style': "outdoors", 'zoom': 0.7},
    showlegend = False)

fig.show()
```



© CARTO (https://carto.com/about-carto/), © OpenStreetMap (http://www.openstreetmap.org/about/) contributors



Add Clusters

New in 5.11

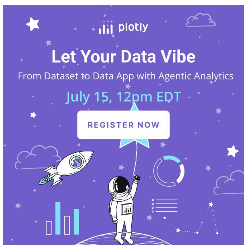
Display clusters of data points by setting cluster. Here, we enable clusters with enabled=True. You can also enable clusters by setting other cluster properties. Other available properties include color (for setting the color of the clusters), size (for setting the size of a cluster step), and step (for configuring how many points it takes to create a cluster or advance to the next cluster step).

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

df = pd.read_csv(
    "https://raw.githubusercontent.com/plotly/datasets/master/2011_february_us_airport_traffic.csv"
)
fig = px.scatter_map(df, lat="lat", lon="long", size="cnt", zoom=3)
fig.update_traces(cluster=dict(enabled=True))
fig.show()
```

© CARTO (https://carto.com/about-carto/), © OpenStreetMap (http://www.openstreetmap.org/about/) contributors

Font Customization

You can customize the font on go.Scattermap traces with textfont. For example, you can set the font family.



```
import plotly.graph_objects as go

fig = go.Figure(go.Scattermap(
    mode = "markers+text+lines",
    lon = [-75, -80, -50], lat = [45, 20, -20],
    marker = {'size': 20, 'symbol': ["bus", "harbor", "airport"]},
    text = ["Bus", "Harbor", "airport"], textposition = "bottom right",
    textfont = dict(size=18, color="black", family="Open Sans Bold")
    ))

fig.update_layout(
    map = {
        'style': "outdoors", 'zoom': 0.7},
    showlegend = False,)

fig.show()
```

© CARTO (https://carto.com/about-carto/), © OpenStreetMap (http://www.openstreetmap.org/about/) contributors

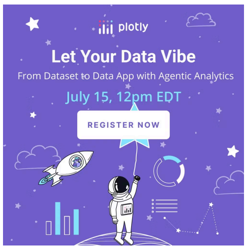
go.Scattermap supports the following values for textfont.family:

'Metropolis Black Italic', 'Metropolis Black', 'Metropolis Bold Italic', 'Metropolis Bold', 'Metropolis Extra Bold Italic', 'Metropolis Extra Bold', 'Metropolis Extra Light Italic', 'Metropolis Extra Light', 'Metropolis Light Italic', 'Metropolis Light', 'Metropolis Medium Italic', 'Metropolis Medium', 'Metropolis Regular Italic', 'Metropolis Regular', 'Metropolis Semi Bold Italic', 'Metropolis Semi Bold', 'Metropolis Thin Italic', 'Metropolis Thin', 'Open Sans Bold Italic', 'Open Sans Bold', 'Open Sans Extrabold Italic', 'Open Sans Extrabold', 'Open Sans Italic', 'Open Sans Light Italic', 'Open Sans Light', 'Open Sans Regular', 'Open Sans Semibold Italic', 'Open Sans Semibold', 'Klokantech Noto Sans Bold', 'Klokantech Noto Sans CJK Bold', 'Klokantech Noto Sans CJK Regular', 'Klokantech Noto Sans Italic', and 'Klokantech Noto Sans Regular'.

Font Weight

New in 5.23

You can specify a numeric font weight on go.Scattermap with textfont.weight.



```
import plotly.graph_objects as go

fig = go.Figure(go.Scattermap(
    mode = "markers+text+lines",
    lon = [-75, -80, -50], lat = [45, 20, -20],
    marker = dict(size=20, symbol=["bus", "harbor", "airport"]),
    text = ["Bus", "Harbor", "airport"], textposition = "bottom right",
    textfont = dict(size=18, color="black", weight=900)
    ))

fig.update_layout(
    map = dict(
        style="outdoors", zoom=0.7),
    showlegend = False,)

fig.show()
```

NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
AUSTRALIA

© CARTO (https://carto.com/about-carto/), © OpenStreetMap (http://www.openstreetmap.org/about/) contributors

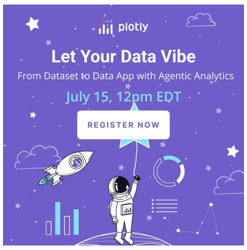
Mapbox Maps

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

The earlier examples using px.scatter_map and go.Scattermap use Maplibre (https://maplibre.org/maplibre-gl-js/docs/) for rendering. These traces were introduced in Plotly.py 5.24 and are now the recommended way to create scatter plots on tile-based maps. There are also traces that use Mapbox (https://docs.mapbox.com): px.scatter_mapbox and go.Scattermapbox

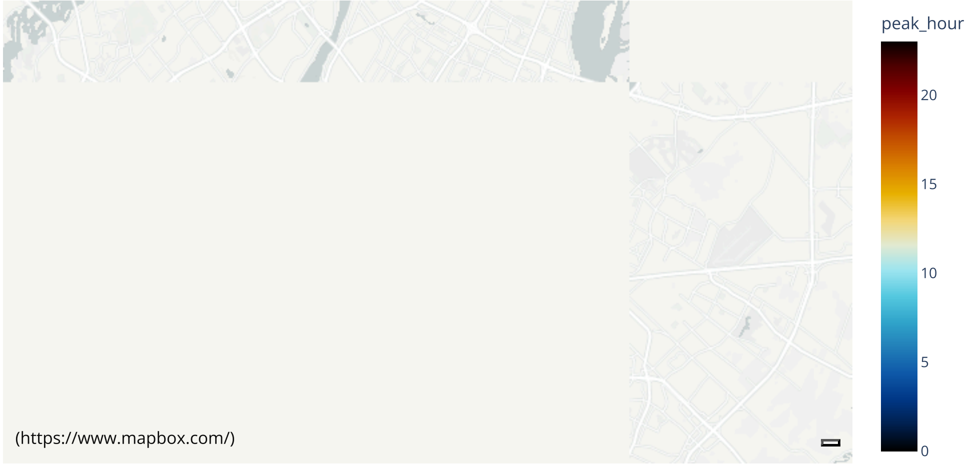
To plot on Mapbox maps with Plotly you *may* need a Mapbox account and a public Mapbox Access Token (https://www.mapbox.com/studio). See our Mapbox Map Layers (/python/mapbox-layers/) documentation for more information.

Here's the first example rewritten to use px.scatter_mapbox.

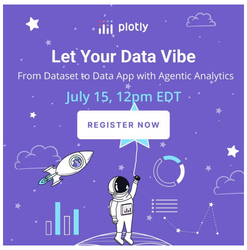


```
import plotly.express as px
px.set_mapbox_access_token(open(".mapbox_token").read())
df = px.data.carshare()
fig = px.scatter_mapbox(df, lat="centroid_lat", lon="centroid_lon",     color="peak_hour", size="car_hours",
                  color_continuous_scale=px.colors.cyclical.IceFire, size_max=15, zoom=10)
fig.show()
```

```
/tmp/ipykernel_17752/3814773356.py:4: DeprecationWarning:

*scatter_mapbox* is deprecated! Use *scatter_map* instead. Learn more at: https://plotly.com/python/mapbox-to-maplibre/
```

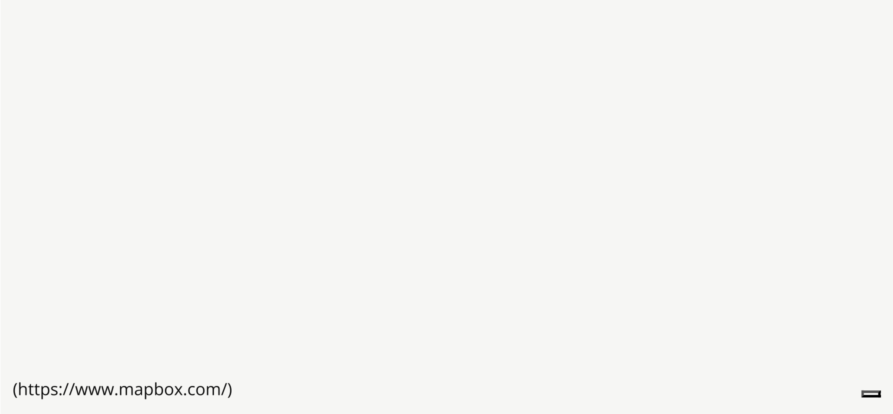


And here's an example using Graph Objects:



```
import plotly.graph_objects as go

mapbox_access_token = open(".mapbox_token").read()

fig = go.Figure(go.Scattermapbox(
        lat=['45.5017'],
        lon=['-73.5673'],
        mode='markers',
        marker=go.scattermapbox.Marker(
            size=14
        ),
        text=['Montreal'],
    ))

fig.update_layout(
    hovermode='closest',
    mapbox=dict(
        accesstoken=mapbox_access_token,
        bearing=0,
        center=go.layout.mapbox.Center(
            lat=45,
            lon=-73
        ),
        pitch=0,
        zoom=5
    )
)

fig.show()
```

```
/tmp/ipykernel_17752/3799354564.py:5: DeprecationWarning:

*scattermapbox* is deprecated! Use *scattermap* instead. Learn more at: https://plotly.com/python/mapbox-to-maplibre/
```



Reference

See function reference for px.scatter_map (https://plotly.com/python-api-reference/generated/plotly.express.scatter_map) or https://plotly.com/python/reference/scattermap/ (https://plotly.com/python/reference/scattermap/) for more information about the attributes available.

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

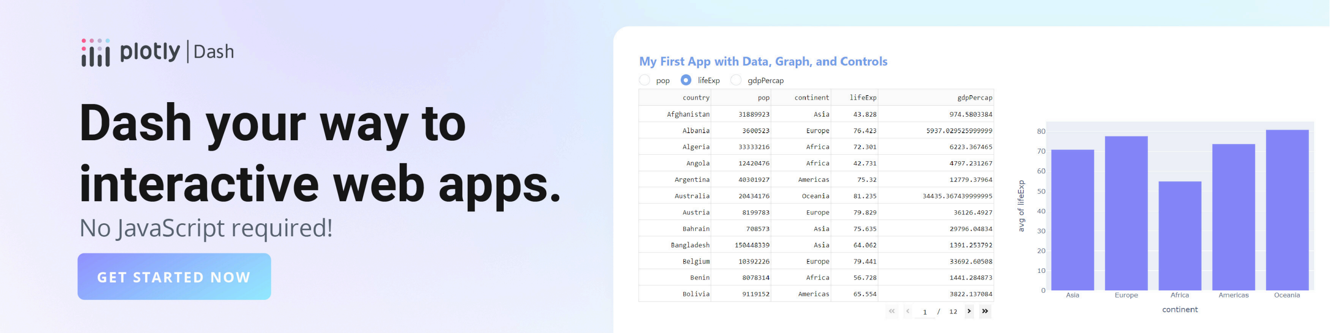
What About Dash?

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
```



My First App with Data, Graph, and Controls

○ pop ● lifeExp ○ gdpPercap

country	pop	continent	lifeExp	gdpPercap
Afghanistan	31889923	Asia	43.828	974.5803384
Albania	3600523	Europe	76.423	5937.029525999999
Algeria	33333216	Africa	72.301	6223.367465
Angola	12420476	Africa	42.731	4797.231267
Argentina	40301927	Americas	75.32	12779.37964
Australia	20434176	Oceania	81.235	34435.367439999995
Austria	8199783	Europe	79.829	36126.4927
Bahrain	708573	Asia	75.635	29796.04834
Bangladesh	150448339	Asia	64.062	1391.253792
Belgium	10392226	Europe	79.441	33692.60508
Benin	8078314	Africa	56.728	1441.284873
Bolivia	9119152	Americas	65.554	3822.137084

« ‹ 1 / 12 › »

(https://dash.plotly.com/tutorial?utm_medium=graphing_libraries&utm_content=python_footer)

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/)