Python (/python) > Maps (/python/maps) > Lines on Tile Maps

Suggest an edit to this page (https://github.com/plotly/plotly.py/edit/doc-prod/doc/python/lines-on-tile-maps.md)

Lines on Tile Maps in Python

ess

How to draw a line on tile-based maps in Python with Plotly.

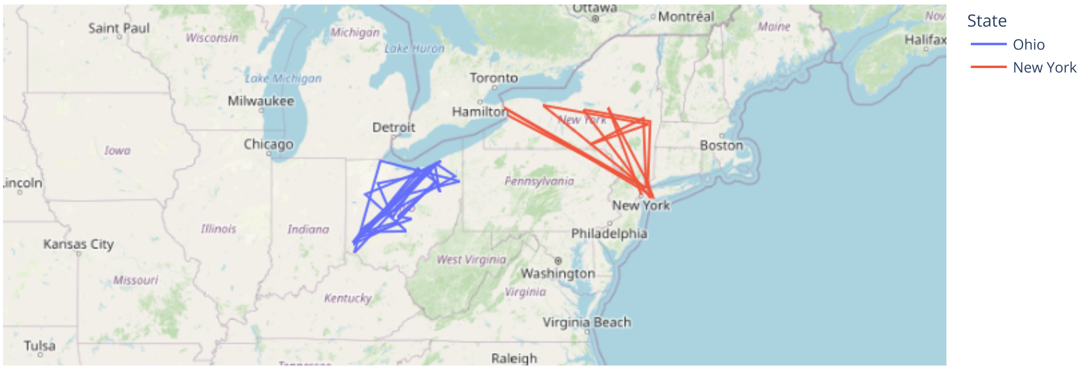
ices

> 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)

Lines on tile maps using Plotly Express

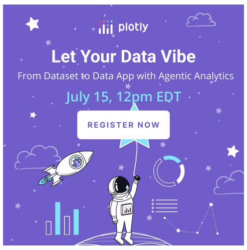
To draw a line on a map, you either can use px.line_map in Plotly Express, or go.Scattermap in Plotly Graph Objects. Here's an example of drawing a line on a tile-based map using Plotly Express.

```
import pandas as pd

us_cities = pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/us-cities-top-1k.csv")
us_cities = us_cities.query("State in ['New York', 'Ohio']")

import plotly.express as px

fig = px.line_map(us_cities, lat="lat", lon="lon", color="State", zoom=3, height=300)

fig.update_layout(map_style="open-street-map", map_zoom=4, map_center_lat = 41,
    margin={"r":0,"t":0,"l":0,"b":0})

fig.show()
```



Lines on maps from GeoPandas

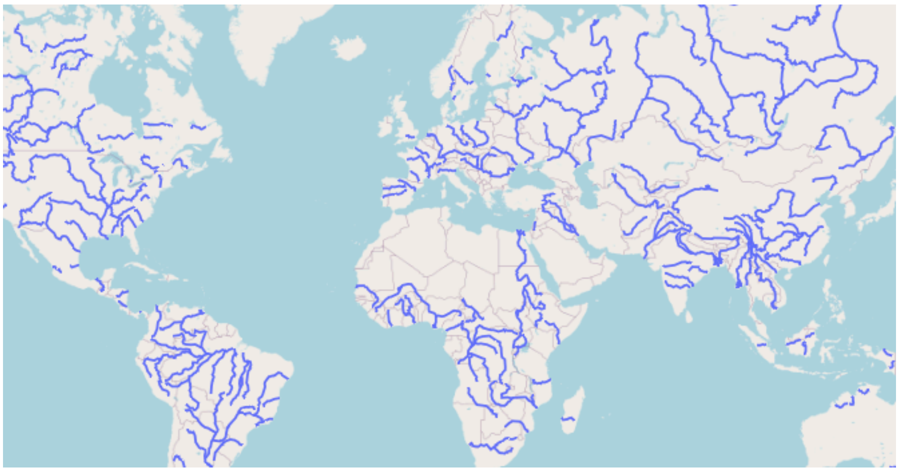
Given a GeoPandas geo-data frame with linestring or multilinestring features, one can extra point data and use px.line_map.



ess

ices

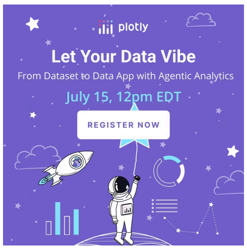
```
import plotly.express as px
import geopandas as gpd
import shapely.geometry
import numpy as np
import wget

# download a zipped shapefile
wget.download("https://plotly.github.io/datasets/ne_50m_rivers_lake_centerlines.zip")

# open a zipped shapefile with the zip:// pseudo-protocol
geo_df = gpd.read_file("zip://ne_50m_rivers_lake_centerlines.zip")

lats = []
lons = []
names = []

for feature, name in zip(geo_df.geometry, geo_df.name):
    if isinstance(feature, shapely.geometry.linestring.LineString):
        linestrings = [feature]
    elif isinstance(feature, shapely.geometry.multilinestring.MultiLineString):
        linestrings = feature.geoms
    else:
        continue
    for linestring in linestrings:
        x, y = linestring.xy
        lats = np.append(lats, y)
        lons = np.append(lons, x)
        names = np.append(names, [name]*len(y))
        lats = np.append(lats, None)
        lons = np.append(lons, None)
        names = np.append(names, None)

fig = px.line_map(lat=lats, lon=lons, hover_name=names,
                     map_style="open-street-map", zoom=1)
fig.show()
```



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

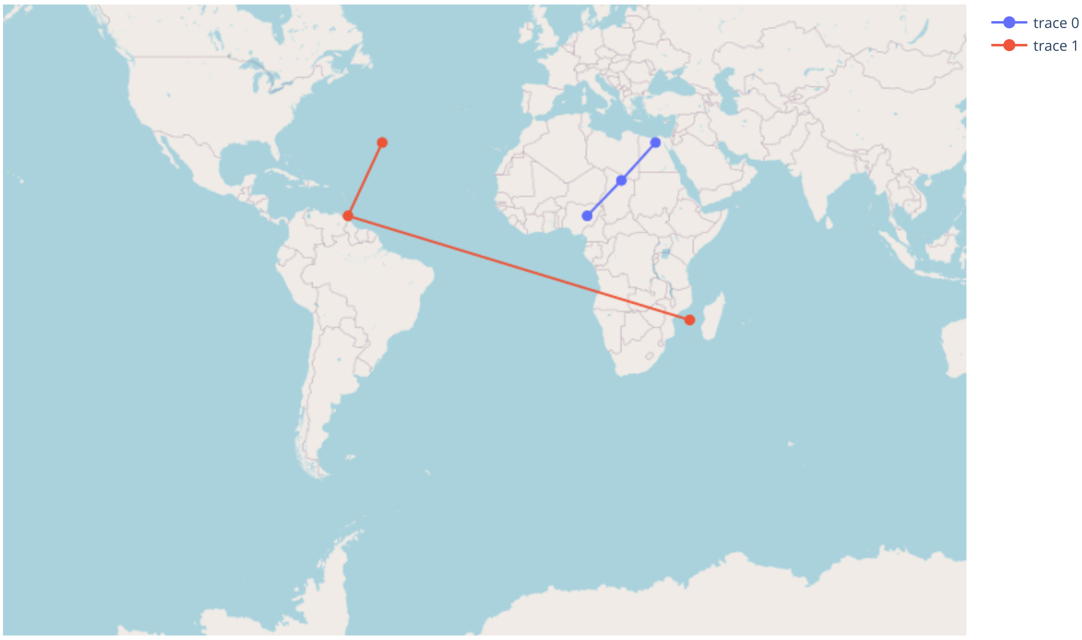
Lines on maps using Scattermap traces

This example uses go.Scattermap and sets the mode (https://plotly.com/python/reference/scattermapbox/#scattermap-mode) attribute to a combination of markers and line.



ess
ices

```
import plotly.graph_objects as go

fig = go.Figure(go.Scattermap(
    mode = "markers+lines",
    lon = [10, 20, 30],
    lat = [10, 20,30],
    marker = {'size': 10}))

fig.add_trace(go.Scattermap(
    mode = "markers+lines",
    lon = [-50, -60,40],
    lat = [30, 10, -20],
    marker = {'size': 10}))

fig.update_layout(
    margin ={'l':0,'t':0,'b':0,'r':0},
    map = {
        'center': {'lon': 10, 'lat': 10},
        'style': "open-street-map",
        'center': {'lon': -20, 'lat': -20},
        'zoom': 1})

fig.show()
```



Mapbox Maps

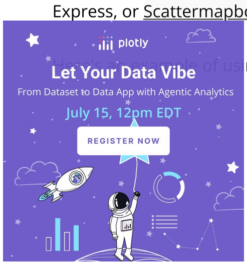
Mapbox traces are deprecated and may be removed in a future version of Plotly.py.

The earlier examples using px.line_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 draw lines on tile-based maps. There are also traces that use Mapbox (https://docs.mapbox.com): px.line_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.

To draw a line on your map, you either can use px.line_mapbox (https://plotly.com/python-api-reference/generated/plotly.express.line_mapbox.html) in Plotly Express, or Scattermapbox (https://plotly.com/python/reference/scattermapbox/) traces. Below we show you how to draw a line on Mapbox using Plotly Express.

of using Scattermapbox.

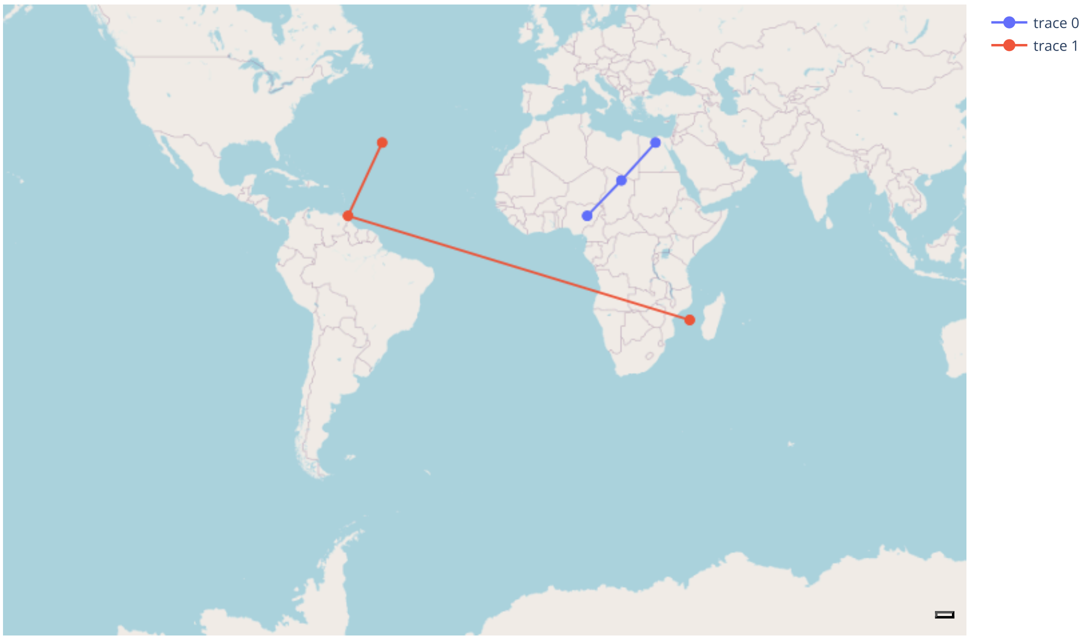


```
import plotly.graph_objects as go

fig = go.Figure(go.Scattermapbox(
    mode = "markers+lines",
    lon = [10, 20, 30],
    lat = [10, 20,30],
    marker = {'size': 10}))

fig.add_trace(go.Scattermapbox(
    mode = "markers+lines",
    lon = [-50, -60,40],
    lat = [30, 10, -20],
    marker = {'size': 10}))

fig.update_layout(
    margin ={'l':0,'t':0,'b':0,'r':0},
    mapbox = {
        'center': {'lon': 10, 'lat': 10},
        'style': "open-street-map",
        'center': {'lon': -20, 'lat': -20},
        'zoom': 1})

fig.show()
```

ess

ices

```
/tmp/ipykernel_11054/3823995118.py:3: DeprecationWarning:

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

/tmp/ipykernel_11054/3823995118.py:9: DeprecationWarning:

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



Reference

See function reference for px.line_map (https://plotly.com/python-api-reference/generated/plotly.express.line_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.line_mapbox (https://plotly.com/python-api-reference/generated/plotly.express.line_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.

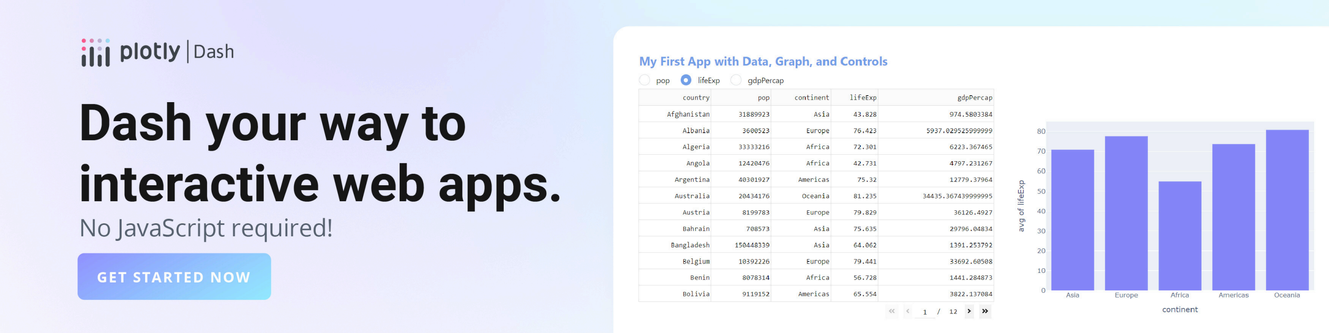
Learn about how to install Dash at https://dash.plot.ly/installation (https://dash.plot.ly/installation).

ess

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:

ices

```
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/)