

a- **plotly** | Graphing Libraries (<https://plotly.com/>) / graphing-libraries/
utm_medium=graphing_libraries&utm_campaign=studio_cloud_launch&utm_content=sidebar

D **Python (/python) > Subplots (/python subplot-charts) > Map Subplots** Suggest an edit to this page (<https://github.com/plotly/plotly.py/edit/doc-prod/doc/python/map-subplots-and-small-multiples.md>)

Map Subplots in Python

How to make map subplots and map small multiples in Python.

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



US map small multiples

```

import plotly.graph_objects as go
import pandas as pd
df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/1962_2006_walmart_store_openings.csv')
df.head()

data = []
layout = dict(
    title = 'New Walmart Stores per year 1962-2006<br>\nSource: <a href="http://www.econ.umn.edu/~holmes/data/WalMart/index.html">\\
University of Minnesota</a>',
    # showlegend = False,
    autosize = False,
    width = 1000,
    height = 900,
    hovermode = False,
    legend = dict(
        x=0.7,
        y=-0.1,
        bgcolor="rgba(255, 255, 255, 0)",
        font = dict( size=11 ),
    )
)
years = df['YEAR'].unique()

for i in range(len(years)):
    geo_key = 'geo'+str(i+1) if i != 0 else 'geo'
    lons = list(df[ df['YEAR'] == years[i] ]['LON'])
    lats = list(df[ df['YEAR'] == years[i] ]['LAT'])
    # Walmart store data
    data.append(
        dict(
            type = 'scattergeo',
            showlegend=False,
            lon = lons,
            lat = lats,
            geo = geo_key,
            name = int(years[i]),
            marker = dict(
                color = "rgb(0, 0, 255)",
                opacity = 0.5
            )
        )
    )
# Year markers
data.append(
    dict(
        type = 'scattergeo',
        showlegend = False,
        lon = [-78],
        lat = [47],
        geo = geo_key,
        text = [years[i]],
        mode = 'text',
    )
)
layout[geo_key] = dict(
    scope = 'usa',
    showland = True,
    landcolor = 'rgb(229, 229, 229)',
    showcountries = False,
    domain = dict( x = [], y = [] ),
    subunitcolor = "rgb(255, 255, 255)",
)
)

```



```

        lonaxis = lonaxis,
        bgcolor = 'rgba(255,200,200,0.0)'
    )

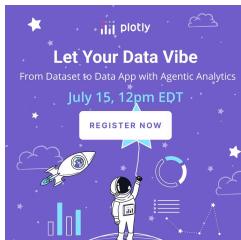
    # Stores per year sparkline
    layout['geo44'] = draw_sparkline({'x':[0.6,0.8], 'y':[0,0.15]}, \
                                    {'range':[-5.0, 30.0]}, {'range':[0.0, 40.0]} )
    data.append(
        dict(
            type = 'scattergeo',
            mode = 'lines',
            lat = list(df.groupby(by=['YEAR']).count()['storenum']/1e1),
            lon = list(range(len(df.groupby(by=['YEAR']).count()['storenum']/1e1))),
            line = dict( color = "rgb(0, 0, 255)" ),
            name = "New stores per year<br>Peak of 178 stores per year in 1990",
            geo = 'geo44',
        )
    )

    # Cumulative sum sparkline
    layout['geo45'] = draw_sparkline({'x':[0.8,1], 'y':[0,0.15]}, \
                                    {'range':[-5.0, 50.0]}, {'range':[0.0, 50.0]} )
    data.append(
        dict(
            type = 'scattergeo',
            mode = 'lines',
            lat = list(df.groupby(by=['YEAR']).count().cumsum()['storenum']/1e2),
            lon = list(range(len(df.groupby(by=['YEAR']).count()['storenum']/1e1))),
            line = dict( color = "rgb(214, 39, 40)" ),
            name ="Cumulative sum<br>3176 stores total in 2006",
            geo = 'geo45',
        )
    )

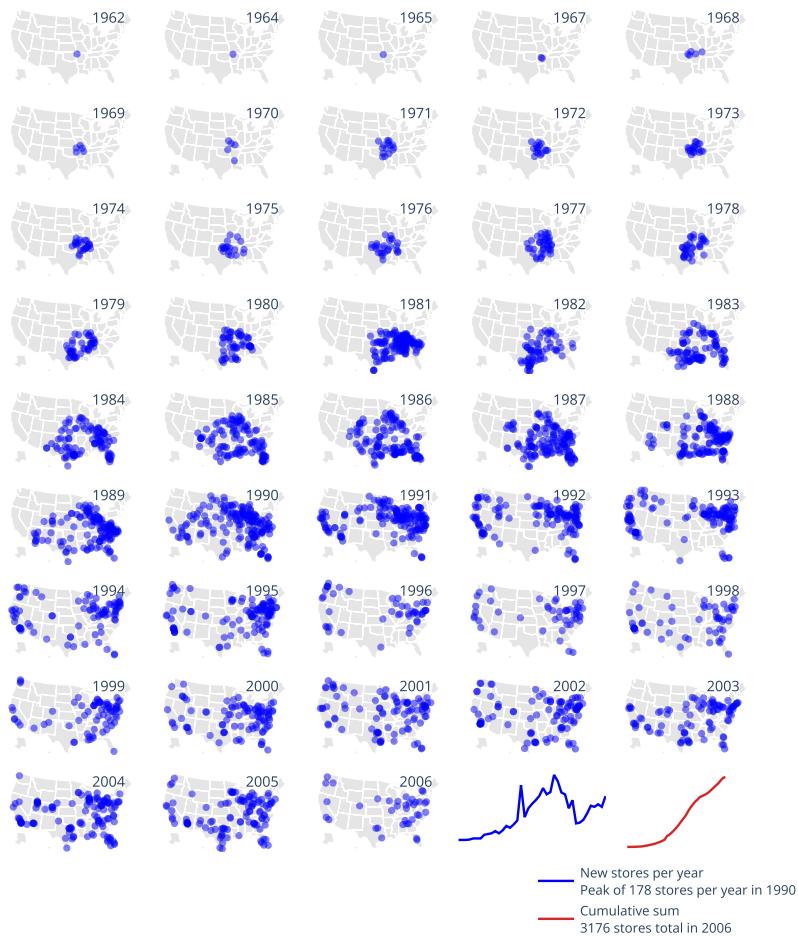
z = 0
COLS = 5
ROWS = 9
for y in reversed(range(ROWS)):
    for x in range(COLS):
        geo_key = 'geo'+str(z+1) if z != 0 else 'geo'
        layout[geo_key]['domain']['x'] = [float(x)/float(COLS), float(x+1)/float(COLS)]
        layout[geo_key]['domain']['y'] = [float(y)/float(ROWS), float(y+1)/float(ROWS)]
    z=z+1
    if z > 42:
        break

fig = go.Figure(data=data, layout=layout)
fig.update_layout(width=800)
fig.show()

```



New Walmart Stores per year 1962-2006
Source: [University of Minnesota](#)



Reference

See <https://plotly.com/python/reference/scattergeo/> (<https://plotly.com/python/reference/scattergeo/>) for more information and chart attribute options!

What About Dash?

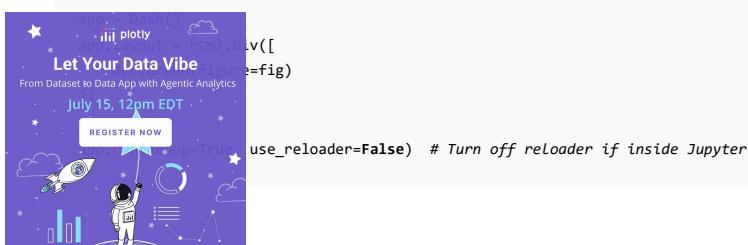
[Dash](#) (<https://dash.plot.ly/>) is an open-source framework for building analytical applications, with no Javascript required, and it is tightly integrated with the Plotly graphing library.

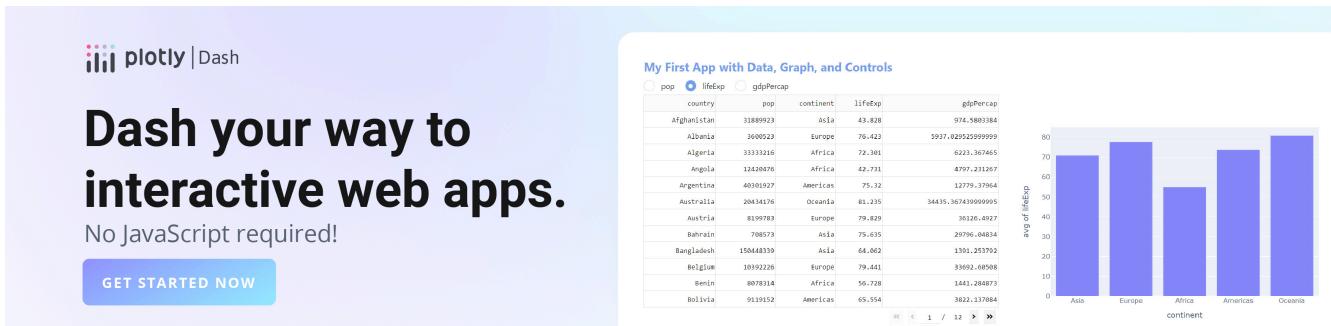
Learn about how to install Dash at <https://dash.plot.ly/installation> (<https://dash.plot.ly/installation>).

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
```





(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
\(HTTPS://GO.PLOT.LY/SUBSCRIPTION\)](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)

Copyright © 2025 Plotly. All rights reserved.

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

