



Carpet Scatter Plot in Python

How to make carpet scatter plots in Python with Plotly.

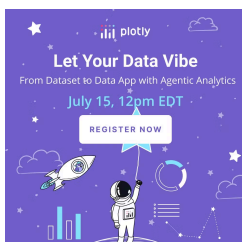
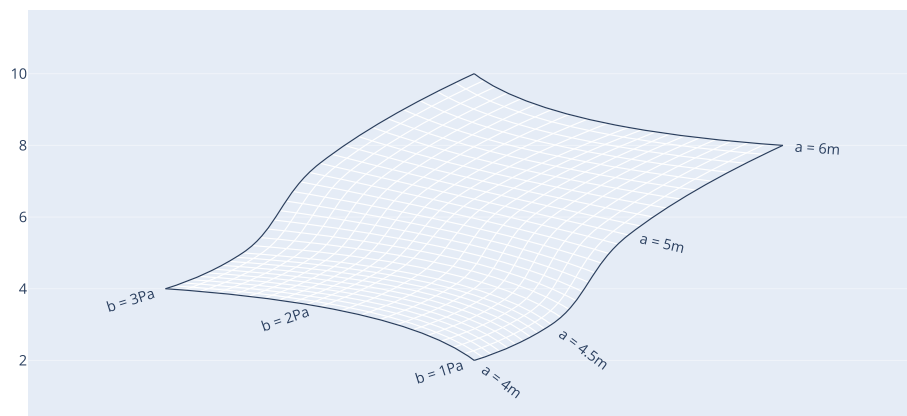
Plotly Studio: Transform any dataset into an interactive data application in minutes with AI. [Sign up for early access now.](https://plotly.com/studio/?utm_medium=graphing-libraries&utm_campaign=studio_early_access&utm_content=sidebar) (https://plotly.com/studio/?utm_medium=graphing-libraries&utm_campaign=studio_early_access&utm_content=sidebar)

Basic Carpet Plot

```
import plotly.graph_objects as go

fig = go.Figure(go.Carpet(
    a = [4, 4, 4, 4.5, 4.5, 4.5, 5, 5, 5, 6, 6, 6],
    b = [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3],
    y = [2, 3.5, 4, 3, 4.5, 5, 5.5, 6.5, 7.5, 8, 8.5, 10],
    aaxis = dict(
        tickprefix = 'a = ',
        ticksuffix = 'm',
        smoothing = 1,
        minorgridcount = 9
    ),
    baxis = dict(
        tickprefix = 'b = ',
        ticksuffix = 'Pa',
        smoothing = 1,
        minorgridcount = 9
    )
))

fig.show()
```



Add Carpet Scatter Trace

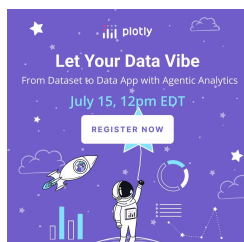
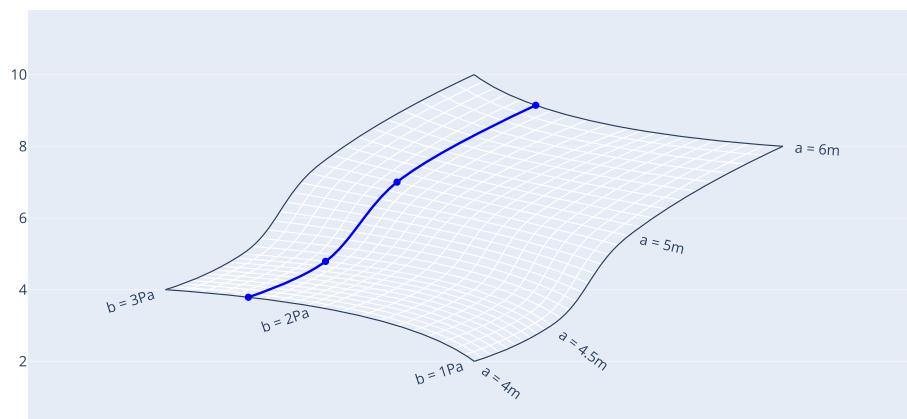
```
import plotly.graph_objects as go

fig = go.Figure()

fig.add_trace(go.Carpet(
    a = [4, 4, 4, 4.5, 4.5, 4.5, 5, 5, 5, 5, 6, 6, 6],
    b = [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3],
    y = [2, 3.5, 4, 3, 4.5, 5, 5.5, 6.5, 7.5, 8, 8.5, 10],
    aaxis = dict(
        tickprefix = 'a = ',
        ticksuffix = 'm',
        smoothing = 1,
        minorgridcount = 9
    ),
    baxis = dict(
        tickprefix = 'b = ',
        ticksuffix = 'Pa',
        smoothing = 1,
        minorgridcount = 9
    )
))

fig.add_trace(go.Scattercarpet(
    a = [4, 4.5, 5, 6],
    b = [2.5, 2.5, 2.5, 2.5],
    line = dict(
        shape = 'spline',
        smoothing = 1,
        color = 'blue'
    )
))

fig.show()
```



Add Multiple Scatter Traces

```
import plotly.graph_objects as go

fig = go.Figure()

fig.add_trace(go.Carpet(
    a = [0.1, 0.2, 0.3],
    b = [1, 2, 3],
    y = [[1, 2.2, 3], [1.5, 2.7, 3.5], [1.7, 2.9, 3.7]],
    cheaterslope = 1,
    aaxis = dict(
        title = "a",
        tickmode = "linear",
        dtick = 0.05
    ),
    baxis = dict(
        title = "b",
        tickmode = "linear",
        dtick = 0.05
    )
))

fig.add_trace(go.Scattercarpet(
    name = "b = 1.5",
    a = [0.05, 0.15, 0.25, 0.35],
    b = [1.5, 1.5, 1.5, 1.5]
))

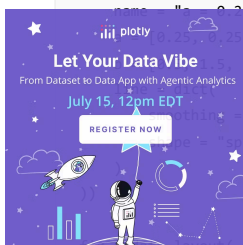
fig.add_trace(go.Scattercarpet(
    name = "b = 2",
    a = [0.05, 0.15, 0.25, 0.35],
    b = [2, 2, 2, 2]
))

fig.add_trace(go.Scattercarpet(
    name = "b = 2.5",
    a = [0.05, 0.15, 0.25, 0.35],
    b = [2.5, 2.5, 2.5, 2.5]
))

fig.add_trace(go.Scattercarpet(
    name = "a = 0.15",
    a = [0.15, 0.15, 0.15, 0.15],
    b = [0.5, 1.5, 2.5, 3.5],
    line = dict(
        smoothing = 1,
        shape = "spline"
    )
))

fig.add_trace(go.Scattercarpet(
    name = "a = 0.2",
    a = [0.2, 0.2, 0.2, 0.2],
    b = [0.5, 1.5, 2.5, 3.5],
    line = dict(
        smoothing = 1,
        shape = "spline"
    ),
    marker = dict(
        size = [10, 20, 30, 40],
        color = ["#000", "#f00", "#ff0", "#fff"]
    )
))

fig.add_trace(go.Scattercarpet(
    name = "a = 0.25",
    a = [0.25, 0.25, 0.25, 0.25],
    b = [0.5, 1.5, 2.5, 3.5],
    line = dict(
        smoothing = 1,
        shape = "spline"
    )
))
```



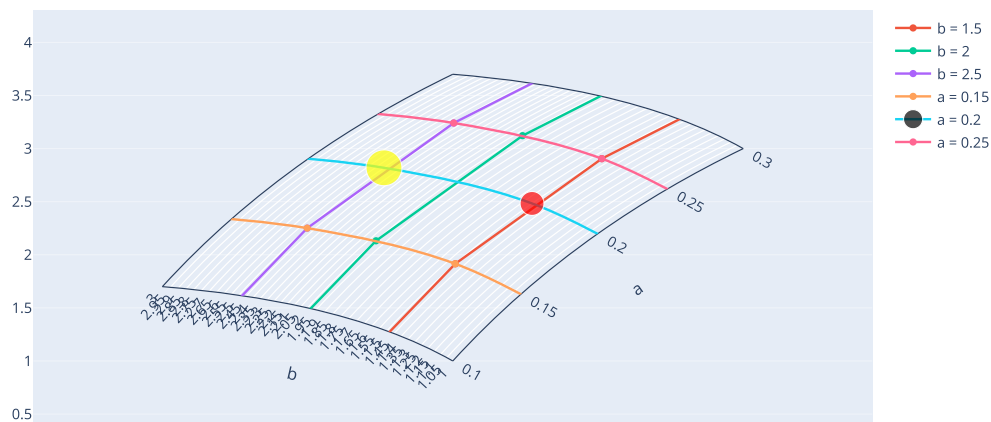
```

title = "scattercarpet extrapolation, clipping, and smoothing",
hovermode = "closest"
)

fig.show()

```

scattercarpet extrapolation, clipping, and smoothing



Reference

See <https://plotly.com/python/reference/scattercarpet/> (<https://plotly.com/python/reference/scattercarpet/>) 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:

```

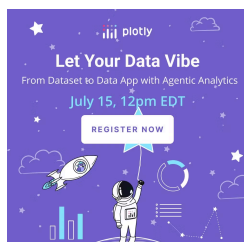
import plotly.graph_objects as go # or plotly.express as px
fig = go.Figure() # or any Plotly Express function e.g. px.bar(...)
# fig.add_trace( ... )
# fig.update_layout( ... )


from dash import Dash, dcc, html

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.

No JavaScript required!

GET STARTED NOW

My First App with Data, Graph, and Controls


pop

lifeExp

gdpPerCap

country	pop	continent	lifeExp	gdpPerCap
Afghanistan	31889923	Asia	43.828	974.5883384
Albania	2600522	Europe	76.422	5937.625525999999
Algeria	33333216	Africa	72.361	6223.367465
Angola	12420676	Africa	42.731	4707.231267
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	1501.253792
Belgium	10392226	Europe	79.441	33692.04908
Benin	8078314	Africa	56.728	1441.284873
Bolivia	9119152	Americas	65.554	3822.137884

1 / 12



continent	avg lifeExp
Asia	~65
Europe	~75
Africa	~55
Americas	~70
Oceania	~78

(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))

About Us

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

Products

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

Pricing

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

Support

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

