



Carpet Plots in Python

How to make carpet 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)

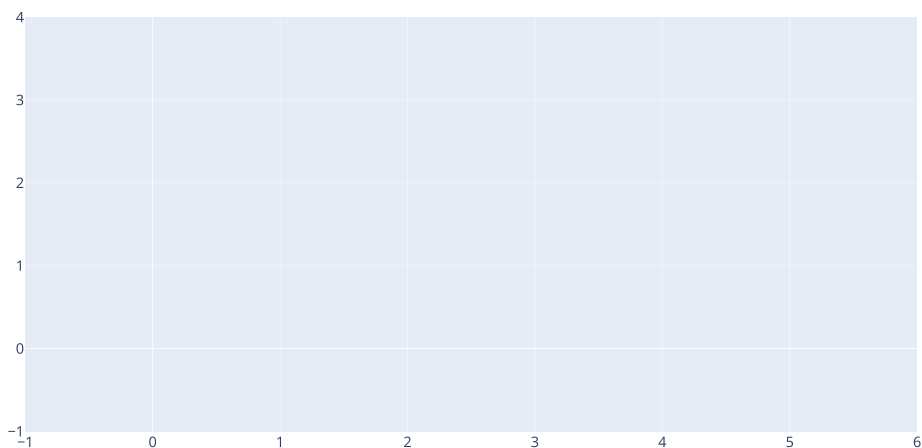
Set X and Y Coordinates

To set the x and y coordinates use x and y attributes. If x coordinate values are omitted a cheater plot will be created. The plot below has a y array specified but requires a and b parameter values before an axis may be plotted.

```
import plotly.graph_objects as go

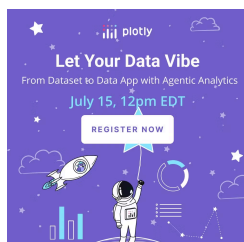
fig = go.Figure(go.Carpet(
    y = [2, 3.5, 4, 3, 4.5, 5, 5.5, 6.5, 7.5, 8, 8.5, 10]
))

fig.show()
```



Add Parameter Values

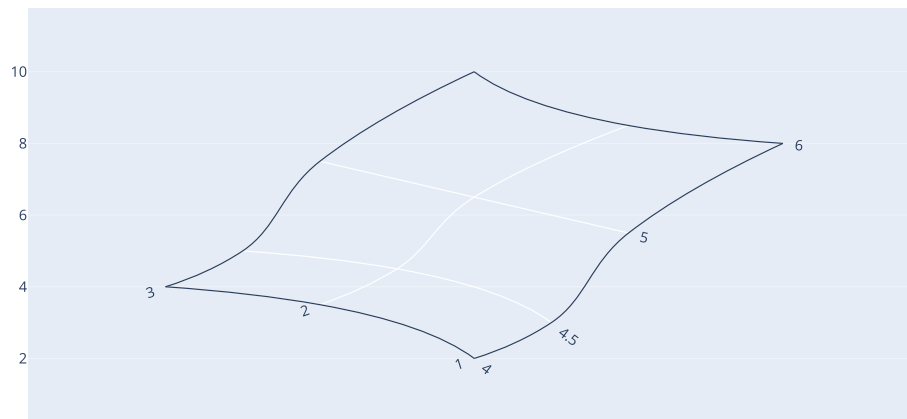
To save parameter values use the a and b attributes.



```
import plotly.graph_objects as go

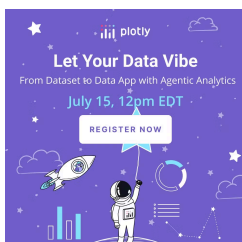
fig = go.Figure(go.Carpet(
    a = [4, 4, 4, 4.5, 4.5, 5, 5, 5, 6, 6, 6],
    b = [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3],
    y = [2, 3.5, 4, 3, 4.5, 5, 5.5, 6.5, 7.5, 8, 8.5, 10]
))

fig.show()
```



Add A and B axis

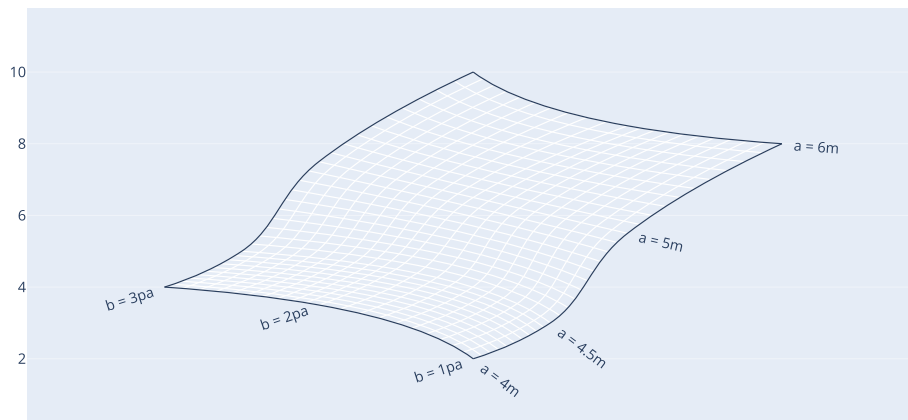
Use `aaxis` or `baxis` list to make changes to the axes. For a more detailed list of attributes refer to [R reference \(https://plotly.com/r/reference/carpet/#carpet-aaxis\)](https://plotly.com/r/reference/carpet/#carpet-aaxis).



```
import plotly.graph_objects as go

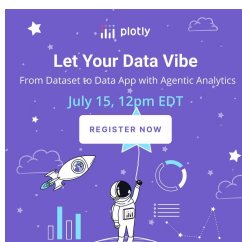
fig = go.Figure(go.Carpet(
    a = [4, 4, 4, 4.5, 4.5, 5, 5, 5, 6, 6, 6],
    b = [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3],
    y = [2, 3.5, 4, 3, 4.5, 5, 5.5, 6.5, 7.5, 8, 8.5, 10],
    aaxis = dict(
        tickprefix = 'a = ',
        ticksuffix = 'm',
        smoothing = 1,
        minorgridcount = 9,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

fig.show()
```



Alternate input format

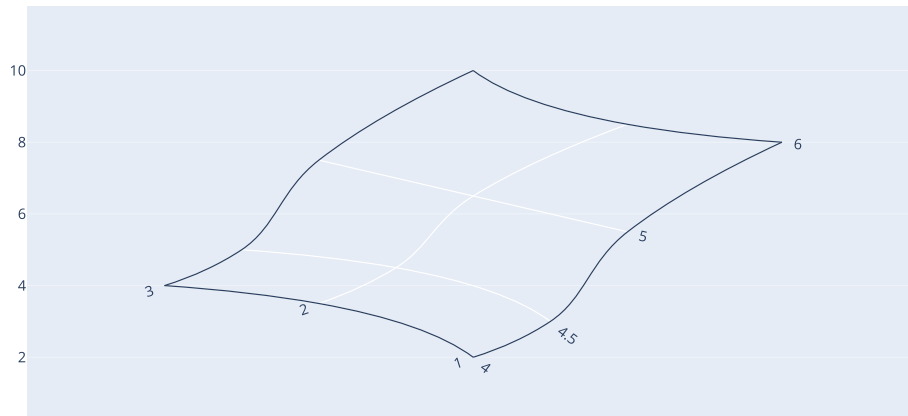
The data arrays x , y may either be specified as one-dimensional arrays of data or as arrays of arrays. If one-dimensional, then x , y , a , and b should all be the same length. If x and y are arrays of arrays, then the length of a should match the inner dimension and the length of b the outer dimension. The plot below represents the same plot as those above.



```
import plotly.graph_objects as go

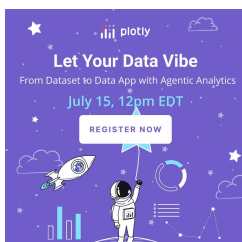
fig = go.Figure(go.Carpet(
    a = [4, 4.5, 5, 6],
    b = [1, 2, 3],
    y = [[2, 3, 5.5, 8],
         [3.5, 4.5, 6.5, 8.5],
         [4, 5, 7.5, 10]]
))

fig.show()
```



Cheater plot layout

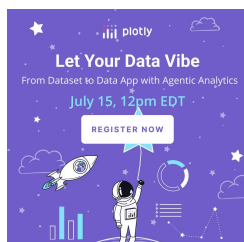
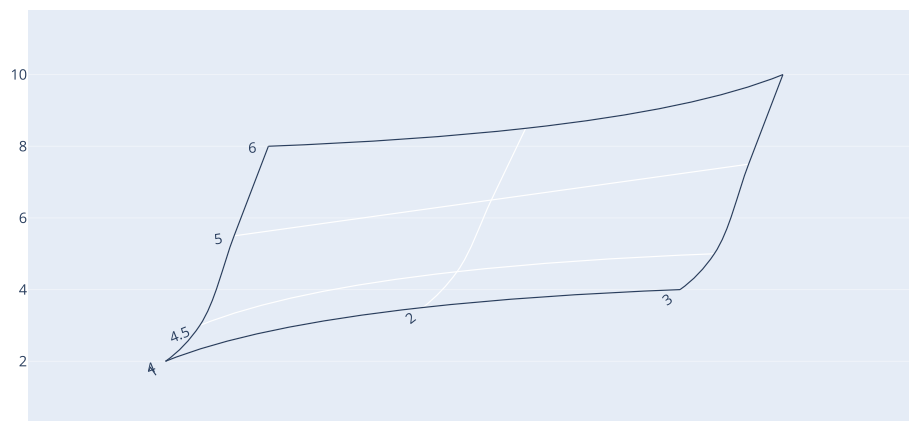
The layout of cheater plots is not unique and depends upon the `cheaterslope` and `axis cheatertype` parameters. If `x` is not specified, each row of the `x` array is constructed based on the formula $a + \text{cheaterslope} * b$, where `a` and `b` are either the value or the integer index of `a` and `b` respectively, depending on the corresponding axis `cheatertype`. Although the layout of the axis below is different than the plots above, it represents the same data as the axes above.



```
import plotly.graph_objects as go

fig = go.Figure(go.Carpet(
    a = [4, 4.5, 5, 6],
    b = [1, 2, 3],
    y = [[2, 3, 5.5, 8],
         [3.5, 4.5, 6.5, 8.5],
         [4, 5, 7.5, 10]],
    cheaterslope = -5,
    aaxis = dict(cheatertype = 'index'),
    baxis = dict(cheatertype = 'value')
))

fig.show()
```



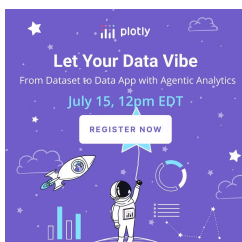
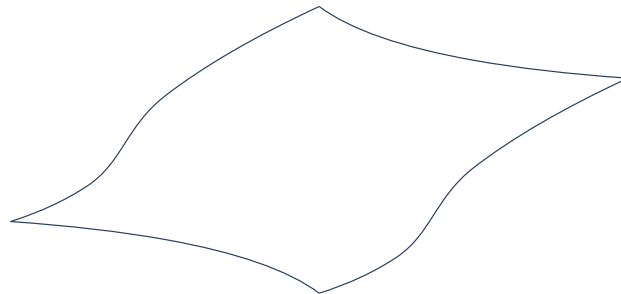
Style A and B axis

```
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,
        minorgridwidth = 0.6,
        minorgridcolor = 'white',
        gridcolor = 'white',
        color = 'white'
    ),
    baxis = dict(
        ticksuffix = 'Pa',
        smoothing = 1,
        minorgridcount = 9,
        minorgridwidth = 0.6,
        gridcolor = 'white',
        minorgridcolor = 'white',
        color = 'white'
    )
))

fig.update_layout(
    plot_bgcolor = 'black',
    paper_bgcolor = 'black',
    xaxis = dict(
        showgrid = False,
        showticklabels = False
    ),
    yaxis = dict(
        showgrid = False,
        showticklabels = False
    )
)

fig.show()
```



Add Points and Contours

To add points and lines see [Carpet Scatter Plots \(https://plotly.com/python/carpet-scatter\)](https://plotly.com/python/carpet-scatter) or to add contours see [Carpet Contour Plots \(https://plotly.com/python/carpet-contour\)](https://plotly.com/python/carpet-contour)

Reference

See [https://plotly.com/python/reference/carpet/ \(https://plotly.com/python/reference/carpet/\)](https://plotly.com/python/reference/carpet/) for more information and chart attribute options!

What About Dash?

[Dash \(https://dash.plot.ly/\)](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 \(https://dash.plot.ly/installation\)](https://dash.plot.ly/installation).

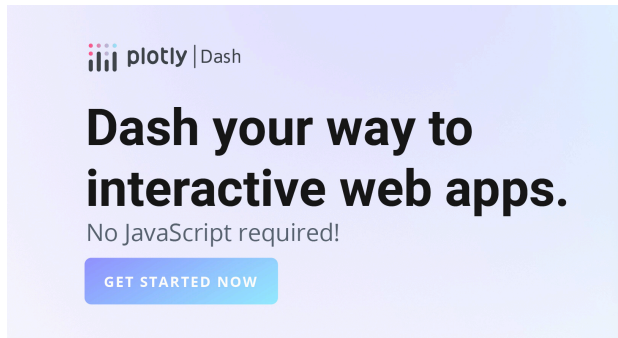
Everywhere in this page that you see `fig.show()`, you can display the same figure in a Dash application by passing it to the figure argument of the [Graph component \(https://dash.plot.ly/dash-core-components/graph\)](https://dash.plot.ly/dash-core-components/graph) from the built-in `dash_core_components` 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

country	pop	continent	lifeExp	gdpPercap
Afghanistan	33889923	Asia	43.828	975.5883384
Albania	3000523	Europe	76.423	5927.629525999999
Algeria	33333216	Africa	72.361	6223.367465
Angola	22828476	Africa	42.731	4707.211267
Argentina	40301927	Americas	75.32	12779.37964
Australia	20434176	Oceania	81.235	34435.367450000005
Austria	8199783	Europe	79.829	35126.4927
Bahrain	708573	Asia	75.635	29796.04654
Bangladesh	150448339	Asia	64.862	1391.253792
Belgium	10392229	Europe	79.441	33892.48588
Benin	8878314	Africa	56.728	1441.284873
Bolivia	9139152	Americas	65.554	3822.137884

avg of lifeExp

continent

(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)

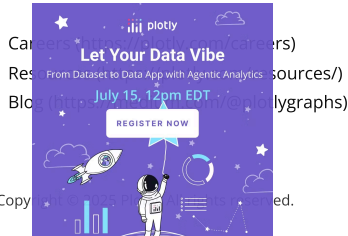
Products

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

Pricing

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

About Us



Let Your Data Vibe
From Dataset to Data App with Agentic Analytics
July 15, 12pm EDT
[REGISTER NOW](#)

Support

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