

Radar Charts in Python

How to make radar charts 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)

A [Radar Chart](https://en.wikipedia.org/wiki/Radar_chart) (https://en.wikipedia.org/wiki/Radar_chart) (also known as a spider plot or star plot) displays multivariate data in the form of a two-dimensional chart of quantitative variables represented on axes originating from the center. The relative position and angle of the axes is typically uninformative. It is equivalent to a [parallel coordinates plot](https://plotly.com/python/parallel-coordinates-plot/) (/python/parallel-coordinates-plot/) with the axes arranged radially.

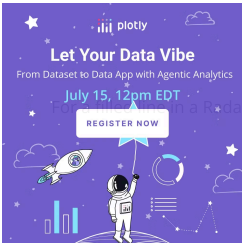
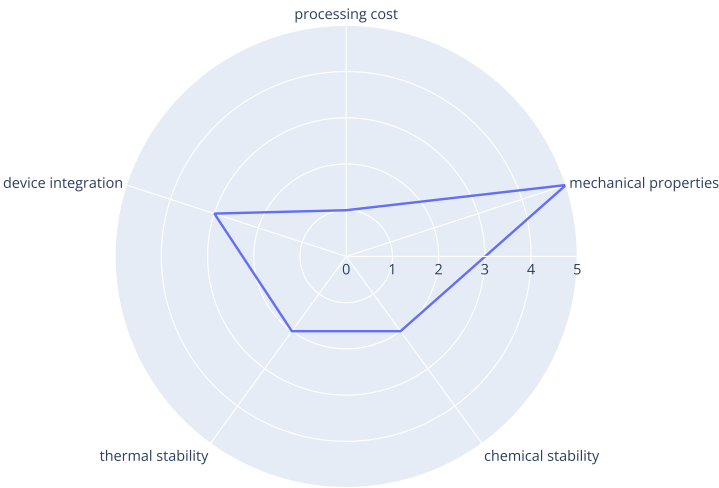
For a Radar Chart, use a [polar chart](https://plotly.com/python/polar-chart/) (/python/polar-chart/) with categorical angular variables, with px.line_polar, or with go.Scatterpolar. See [more examples of polar charts](https://plotly.com/python/polar-chart/) (/python/polar-chart/).

Radar Chart with Plotly Express

[Plotly Express](https://plotly.com/python/plotly-express/) (/python/plotly-express/) is the easy-to-use, high-level interface to Plotly, which [operates on a variety of types of data](https://plotly.com/python/px-arguments/) (/python/px-arguments/) and produces [easy-to-style figures](https://plotly.com/python/styling-plotly-express/) (/python/styling-plotly-express/).

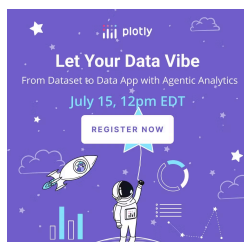
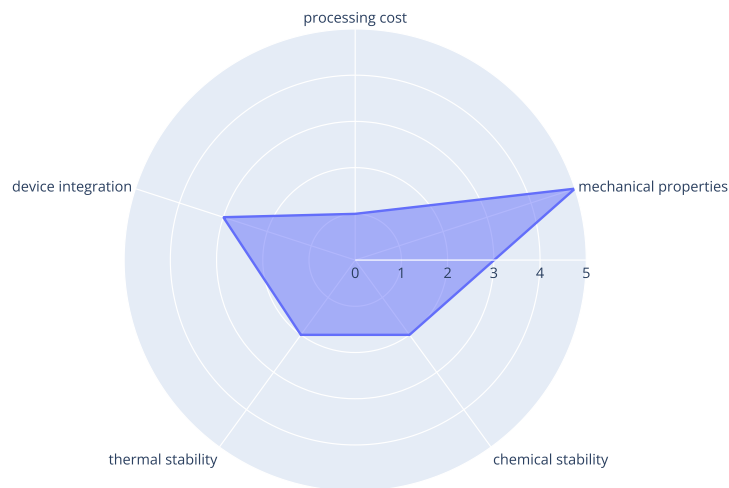
Use line_close=True for closed lines.

```
import plotly.express as px
import pandas as pd
df = pd.DataFrame(dict(
    r=[1, 5, 2, 2, 3],
    theta=['processing cost', 'mechanical properties', 'chemical stability',
           'thermal stability', 'device integration']))
fig = px.line_polar(df, r='r', theta='theta', line_close=True)
fig.show()
```



For Chart, update the figure created with px.line_polar with fig.update_traces.

```
import plotly.express as px
import pandas as pd
df = pd.DataFrame(dict(
    r=[1, 5, 2, 2, 3],
    theta=['processing cost', 'mechanical properties', 'chemical stability',
          'thermal stability', 'device integration']))
fig = px.line_polar(df, r='r', theta='theta', line_close=True)
fig.update_traces(fill='toself')
fig.show()
```



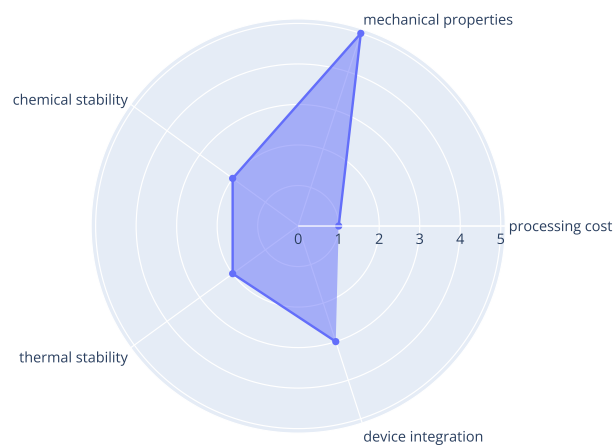
Basic Radar Chart with go.Scatterpolar

```
import plotly.graph_objects as go

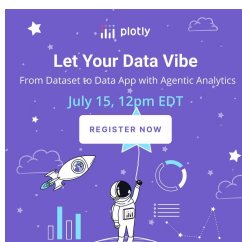
fig = go.Figure(data=go.Scatterpolar(
    r=[1, 5, 2, 2, 3],
    theta=['processing cost', 'mechanical properties', 'chemical stability', 'thermal stability',
           'device integration'],
    fill='toself'
))

fig.update_layout(
    polar=dict(
        radialaxis=dict(
            visible=True
        ),
    ),
    showlegend=False
)

fig.show()
```



Multiple Trace Radar Chart



```
import plotly.graph_objects as go

categories = ['processing cost', 'mechanical properties', 'chemical stability',
             'thermal stability', 'device integration']

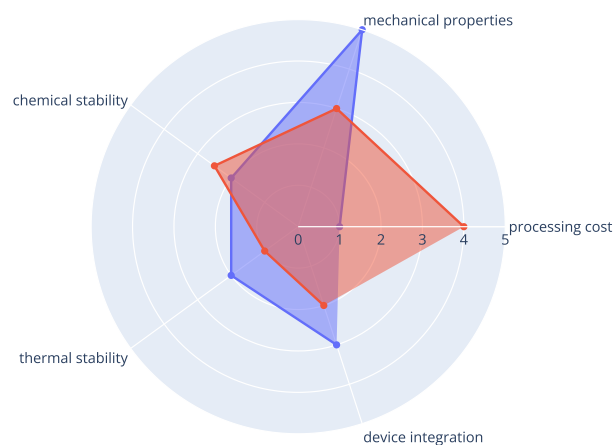
fig = go.Figure()

fig.add_trace(go.Scatterpolar(
    r=[1, 5, 2, 2, 3],
    theta=categories,
    fill='toself',
    name='Product A'
))

fig.add_trace(go.Scatterpolar(
    r=[4, 3, 2.5, 1, 2],
    theta=categories,
    fill='toself',
    name='Product B'
))

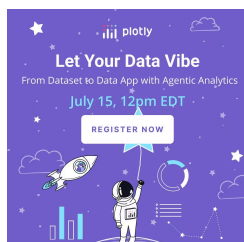
fig.update_layout(
    polar=dict(
        radialaxis=dict(
            visible=True,
            range=[0, 5]
        )
    ),
    showlegend=False
)

fig.show()
```



Reference

See [function reference for px.line_polar](https://plotly.com/python-api-reference/generated/plotly.express.line_polar/) (https://plotly.com/python-api-reference/generated/plotly.express.line_polar/) or <https://plotly.com/python/reference/scatterpolar/> (<https://plotly.com/python/reference/scatterpolar/>) 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:

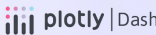
ilar

```
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	3600523	Europe	76.423	5937.829525999999
Algeria	33333216	Africa	72.381	6223.367465
Angola	12420476	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	706573	Asia	75.635	29796.04834
Bangladesh	150448339	Asia	64.062	1701.253792
Belgium	10391226	Europe	79.441	33962.04968
Benin	8878314	Africa	56.728	1441.284873
Bolivia	9139352	Americas	65.554	3822.137884



(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>)

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/>)

Support

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

Pricing

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

