

2D Histogram Contour in Python

Contours
s
unt

How to make 2D Histogram Contour 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)

2D Histogram Contours or Density Contours

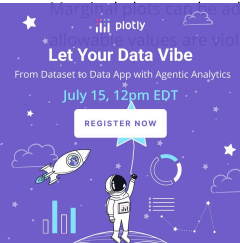
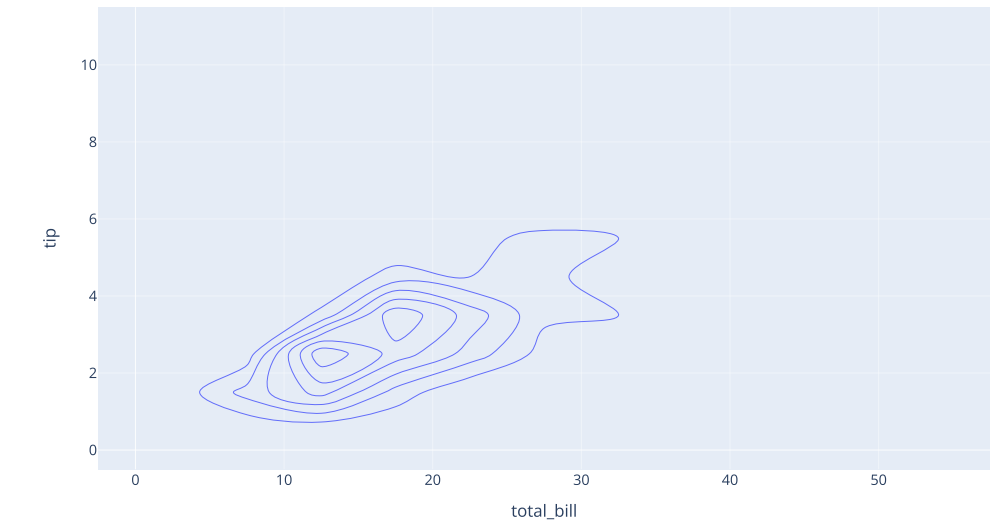
A 2D histogram contour plot, also known as a density contour plot, is a 2-dimensional generalization of a [histogram \(/python/histograms/\)](#) which resembles a [contour plot \(/python/contour-plots/\)](#) but is computed by grouping a set of points specified by their x and y coordinates into bins, and applying an aggregation function such as count or sum (if z is provided) to compute the value to be used to compute contours. This kind of visualization (and the related [2D histogram, or density heatmap \(/python/2d-histogram/\)](#)) is often used to manage over-plotting, or situations where showing large data sets as [scatter plots \(/python/line-and-scatter/\)](#) would result in points overlapping each other and hiding patterns.

Density Contours with Plotly Express

[Plotly Express \(/python/plotly-express/\)](#) is the easy-to-use, high-level interface to Plotly, which [operates on a variety of types of data \(/python/px-arguments/\)](#) and produces [easy-to-style figures \(/python/styling-plotly-express/\)](#). The Plotly Express function `density_contour()` can be used to produce density contours.

```
import plotly.express as px
df = px.data.tips()

fig = px.density_contour(df, x="total_bill", y="tip")
fig.show()
```

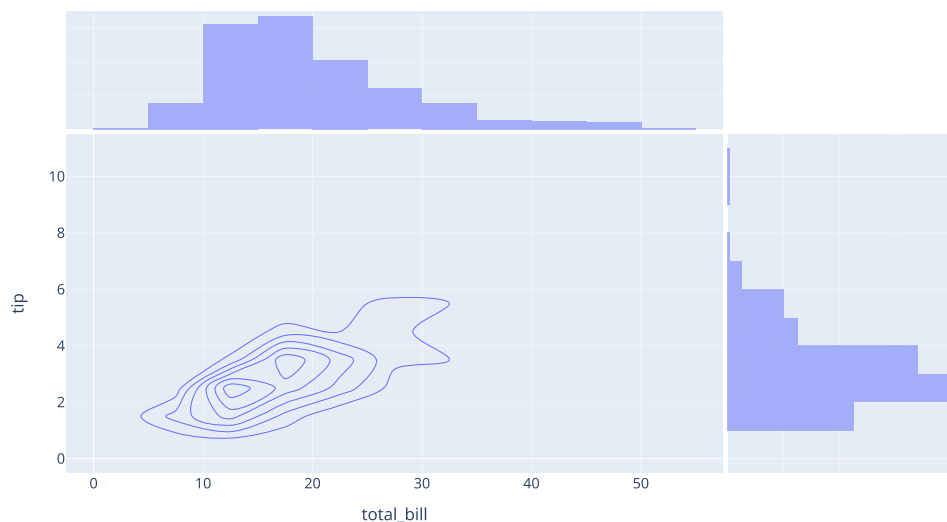


ded to visualize the 1-dimensional distributions of the two variables. Here we use a marginal [histogram \(/python/histograms/\)](#). Other
in, box and rug.

```
import plotly.express as px
df = px.data.tips()

fig = px.density_contour(df, x="total_bill", y="tip", marginal_x="histogram", marginal_y="histogram")
fig.show()
```

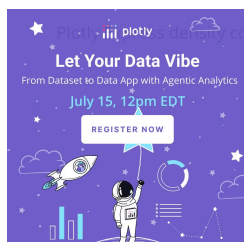
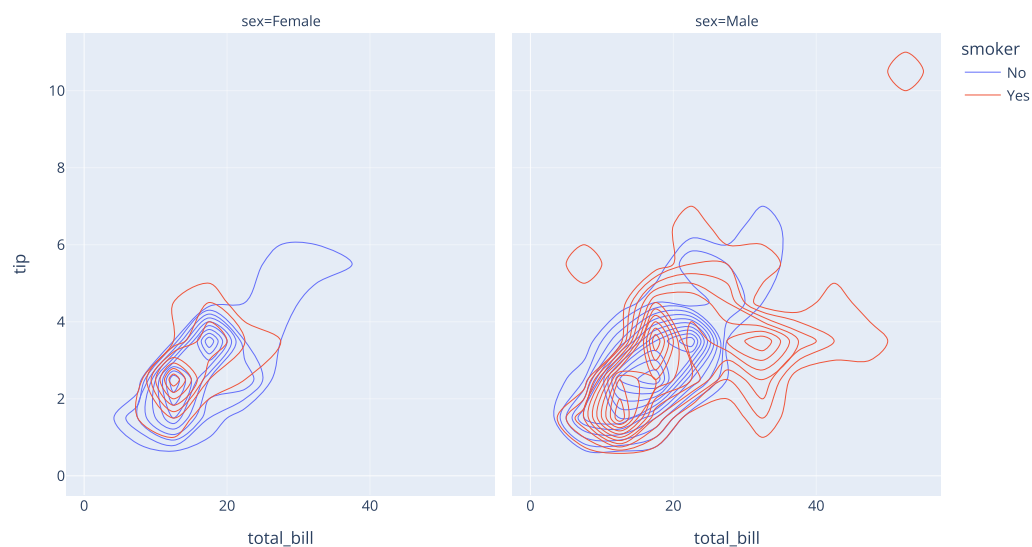
Contours
s
ount



Density contours can also be [faceted](#) ([/python/facet-plots/](#)) and [discretely colored](#) ([/python/discrete-color/](#)):

```
import plotly.express as px
df = px.data.tips()

fig = px.density_contour(df, x="total_bill", y="tip", facet_col="sex", color="smoker")
fig.show()
```



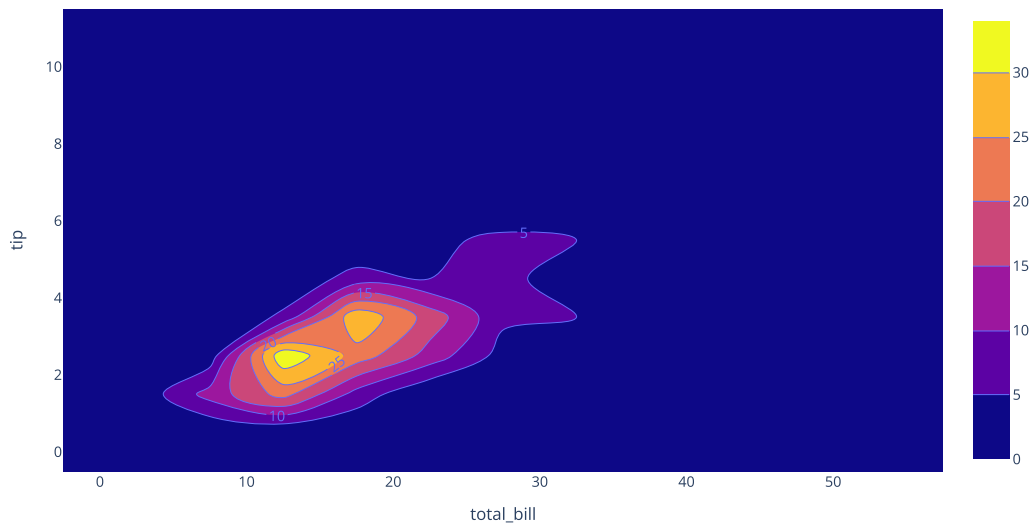
```
import plotly.express as px
df = px.data.tips()

fig = px.density_contour(df, x="total_bill", y="tip")
fig.update_traces(contours_coloring="fill", contours_showlabels = True)
fig.show()
```

Contours

s

ount

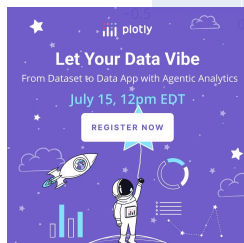
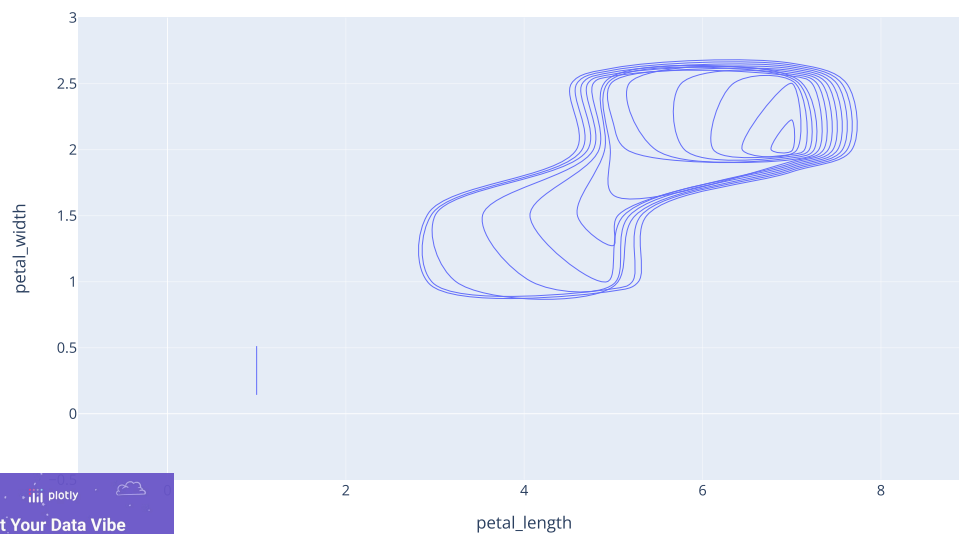


Other aggregation functions than count

By passing in a `z` value and a `histfunc`, density contours can perform basic aggregation operations. Here we show average Sepal Length grouped by Petal Length and Petal Width for the Iris dataset.

```
import plotly.express as px
df = px.data.iris()

fig = px.density_contour(df, x="petal_length", y="petal_width", z="sepal_length", histfunc="avg")
fig.show()
```



2D Histograms with Graph Objects

To build this kind of figure with [graph objects](https://plotly.com/python/graph-objects/) ([python/graph-objects/](https://plotly.com/python/graph-objects/)) without using Plotly Express, we can use the `go.Histogram2dContour` class.

Basic 2D Histogram Contour

Contours
s
ount

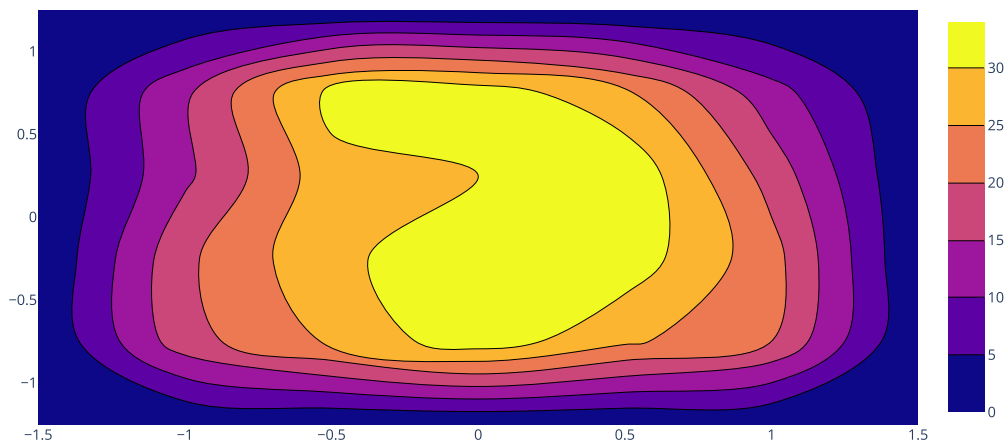
```
import plotly.graph_objects as go

import numpy as np
np.random.seed(1)

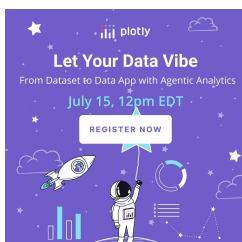
x = np.random.uniform(-1, 1, size=500)
y = np.random.uniform(-1, 1, size=500)

fig = go.Figure(go.Histogram2dContour(
    x = x,
    y = y
))

fig.show()
```



2D Histogram Contour Colorscale



```
import plotly.graph_objects as go

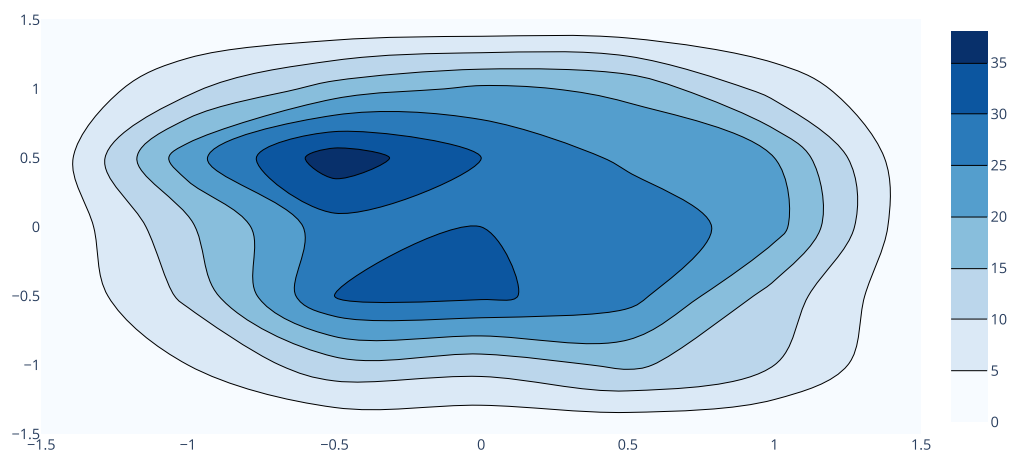
import numpy as np

x = np.random.uniform(-1, 1, size=500)
y = np.random.uniform(-1, 1, size=500)

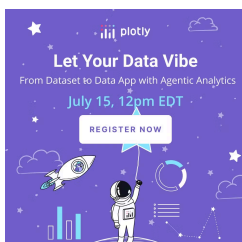
fig = go.Figure(go.Histogram2dContour(
    x = x,
    y = y,
    colorscale = 'Blues'
))

fig.show()
```

Contours
s
ount



2D Histogram Contour Styled



Contours
s
ount

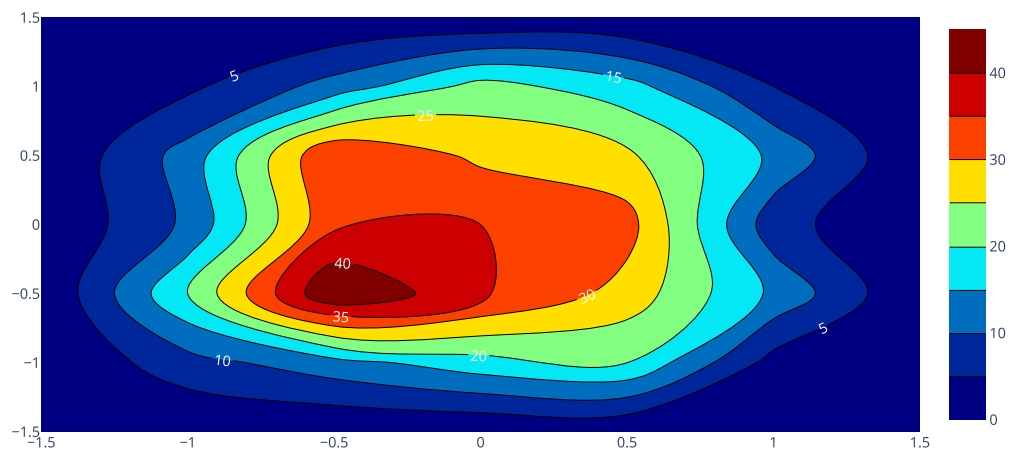
```
import plotly.graph_objects as go

import numpy as np

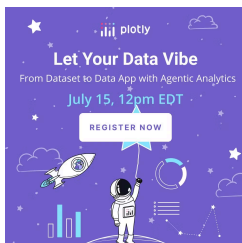
x = np.random.uniform(-1, 1, size=500)
y = np.random.uniform(-1, 1, size=500)

fig = go.Figure(go.Histogram2dContour(
    x = x,
    y = y,
    colorscale = 'Jet',
    contours = dict(
        showlabels = True,
        labelfont = dict(
            family = 'Raleway',
            color = 'white'
        )
    ),
    hoverlabel = dict(
        bgcolor = 'white',
        bordercolor = 'black',
        font = dict(
            family = 'Raleway',
            color = 'black'
        )
    )
))

fig.show()
```



2D Histogram Contour Subplot



Contours
s
ount

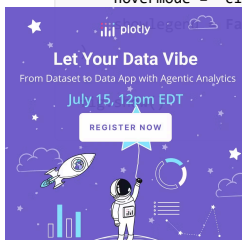
```
import plotly.graph_objects as go

import numpy as np

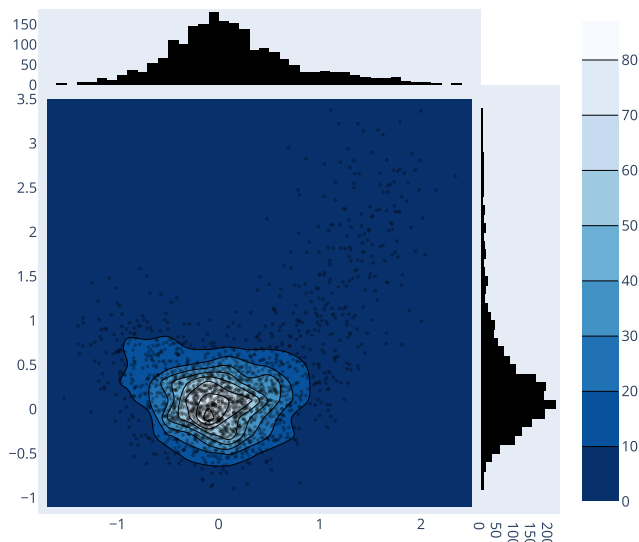
t = np.linspace(-1, 1.2, 2000)
x = (t**3) + (0.3 * np.random.randn(2000))
y = (t**6) + (0.3 * np.random.randn(2000))

fig = go.Figure()
fig.add_trace(go.Histogram2dContour(
    x = x,
    y = y,
    colorscale = 'Blues',
    reversescale = True,
    xaxis = 'x',
    yaxis = 'y'
))
fig.add_trace(go.Scatter(
    x = x,
    y = y,
    xaxis = 'x',
    yaxis = 'y',
    mode = 'markers',
    marker = dict(
        color = 'rgba(0,0,0,0.3)',
        size = 3
    )
))
fig.add_trace(go.Histogram(
    y = y,
    xaxis = 'x2',
    marker = dict(
        color = 'rgba(0,0,0,1)'
    )
))
fig.add_trace(go.Histogram(
    x = x,
    yaxis = 'y2',
    marker = dict(
        color = 'rgba(0,0,0,1)'
    )
))

fig.update_layout(
    autosize = False,
    xaxis = dict(
        zeroline = False,
        domain = [0,0.85],
        showgrid = False
    ),
    yaxis = dict(
        zeroline = False,
        domain = [0,0.85],
        showgrid = False
    ),
    xaxis2 = dict(
        zeroline = False,
        domain = [0.85,1],
        showgrid = False
    ),
    yaxis2 = dict(
        zeroline = False,
        domain = [0.85,1],
        showgrid = False
    ),
    height = 600,
    width = 600,
    bargap = 0,
    hovermode = 'closest',
    use_bgcolor = False
)
```



Contours
s
ount



Reference

See <https://plotly.com/python/reference/histogram2dcontour/> (<https://plotly.com/python/reference/histogram2dcontour/>) 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>).

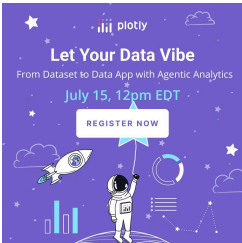
Everywhere in this page that you see `fig.show()`, you can display the same figure in a Dash application by passing it to the `figure` argument of the [Graph component](https://dash.plot.ly/dash-core-components/graph) (<https://dash.plot.ly/dash-core-components/graph>) from the built-in `dash_core_components`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



Contours
s
ount



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 | 5937.629525999999 |
| 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 | 29796.04854 |
| Bangladesh | 150448339 | Asia | 64.062 | 1501.253792 |
| Belgium | 10592226 | Europe | 79.441 | 33692.04908 |
| Benin | 8078314 | Africa | 56.728 | 1441.284873 |
| Bolivia | 9119152 | Americas | 65.554 | 3822.137884 |

1

2

3



| 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/>)

Support

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

Pricing

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

