

# Ternary Plots in Python

How to make Ternary 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)

## Ternary Plots

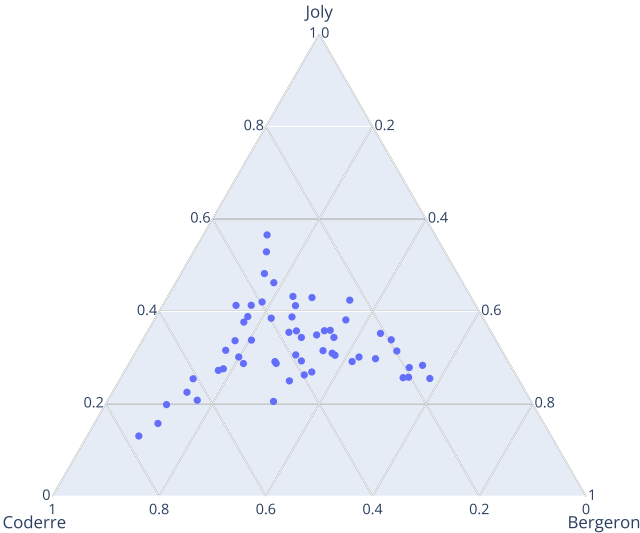
A ternary plot depicts the ratios of three variables as positions in an equilateral triangle.

## Ternary scatter plot with Plotly Express

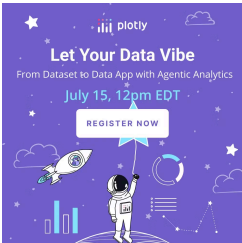
[Plotly Express \(/python/plotly-express/\)](#) is the easy-to-use, high-level interface to Plotly, which [operates on a variety of types of data \(/python/px-arguments/\)](#) and produces [easy-to-style figures \(/python/styling-plotly-express/\)](#).

Here we use `px.scatter_ternary` to visualize the three-way split between the three major candidates in a municipal election.

```
import plotly.express as px
df = px.data.election()
fig = px.scatter_ternary(df, a="Joly", b="Coderne", c="Bergeron")
fig.show()
```

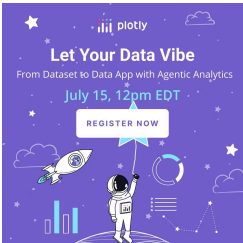
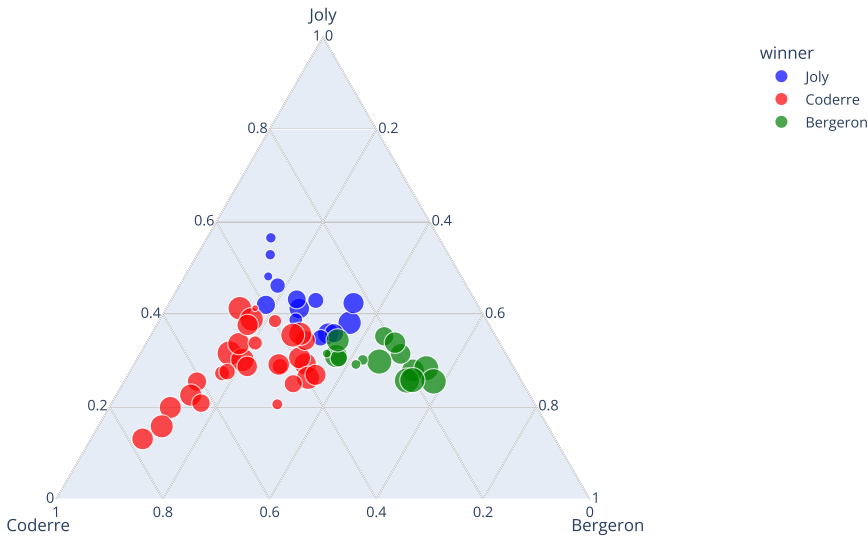


We can scale and color the markers to produce a ternary bubble chart.



```
import plotly.express as px
df = px.data.election()
fig = px.scatter_ternary(df, a="Joly", b="Coderre", c="Bergeron", hover_name="district",
    color="winner", size="total", size_max=15,
    color_discrete_map = {"Joly": "blue", "Bergeron": "green", "Coderre":"red"} )
fig.show()
```

ess  
ph



## Ternary scatter plot with Plotly Graph Objects

```
import plotly.graph_objects as go

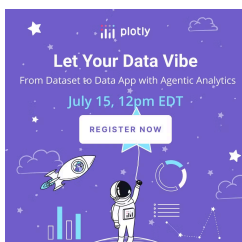
rawData = [
    {'journalist':75,'developer':25,'designer':0,'label':'point 1'},
    {'journalist':70,'developer':10,'designer':20,'label':'point 2'},
    {'journalist':75,'developer':20,'designer':5,'label':'point 3'},
    {'journalist':5,'developer':60,'designer':35,'label':'point 4'},
    {'journalist':10,'developer':80,'designer':10,'label':'point 5'},
    {'journalist':10,'developer':90,'designer':0,'label':'point 6'},
    {'journalist':20,'developer':70,'designer':10,'label':'point 7'},
    {'journalist':10,'developer':20,'designer':70,'label':'point 8'},
    {'journalist':15,'developer':5,'designer':80,'label':'point 9'},
    {'journalist':10,'developer':10,'designer':80,'label':'point 10'},
    {'journalist':20,'developer':10,'designer':70,'label':'point 11'},
];

def makeAxis(title, tickangle):
    return {
        'title': {'text': title, 'font': { 'size': 20}},
        'tickangle': tickangle,
        'tickfont': { 'size': 15 },
        'tickcolor': 'rgba(0,0,0,0)',
        'ticklen': 5,
        'showline': True,
        'showgrid': True
    }

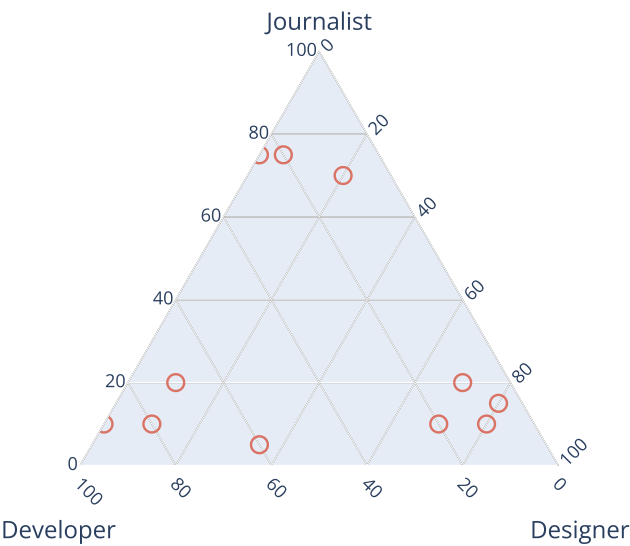
fig = go.Figure(go.Scatterternary(({
    'mode': 'markers',
    'a': [i for i in map(lambda x: x['journalist'], rawData)],
    'b': [i for i in map(lambda x: x['developer'], rawData)],
    'c': [i for i in map(lambda x: x['designer'], rawData)],
    'text': [i for i in map(lambda x: x['label'], rawData)],
    'marker': {
        'symbol': 100,
        'color': '#DB7365',
        'size': 14,
        'line': { 'width': 2 }
    }
})))

fig.update_layout({
    'ternary': {
        'sum': 100,
        'aaxis': makeAxis('Journalist', 0),
        'baxis': makeAxis('<br>Developer', 45),
        'caxis': makeAxis('<br>Designer', -45)
    },
    'annotations': [{
        'showarrow': False,
        'text': 'Simple Ternary Plot with Markers',
        'x': 0.5,
        'y': 1.3,
        'font': { 'size': 15 }
    }]
})

fig.show()
```



Simple Ternary Plot with Markers



Reference

See [function reference for px\(scatter\\_ternary\)](https://plotly.com/python-api-reference/generated/plotly.express.scatter_ternary/) ([https://plotly.com/python-api-reference/generated/plotly.express.scatter\\_ternary/](https://plotly.com/python-api-reference/generated/plotly.express.scatter_ternary/)) or <https://plotly.com/python/reference/scatterternary/> (<https://plotly.com/python/reference/scatterternary/>) 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
```

# Dash your way to interactive web apps.

Let Your Data Vibe

From Dataset to Data App with Agentic Analytics

July 15, 12pm EDT

REGISTER NOW

Plotly.com/tutorial?utm\_medium=graphing\_libraries&utm\_content=python\_footer

My First App with Data, Graph, and Controls

pop lifeExp gdpPerCap

| country     | pop      | continent | lifeExp | gdpPerCap          |
|-------------|----------|-----------|---------|--------------------|
| Afghanistan | 31889923 | Asia      | 43.828  | 974.5883384        |
| Albania     | 3606523  | Europe    | 76.423  | 5927.829525999999  |
| Algeria     | 33333236 | Africa    | 72.381  | 6223.367405        |
| Angola      | 13426476 | Africa    | 42.731  | 4707.231267        |
| Argentina   | 60005927 | American  | 75.32   | 12779.37964        |
| Australia   | 20434176 | Oceania   | 81.235  | 34435.367439999995 |
| Austria     | 8190783  | Europe    | 79.829  | 36126.6927         |
| Bahrain     | 708573   | Asia      | 75.635  | 29796.04834        |
| Bangladesh  | 15048339 | Asia      | 64.062  | 1301.253792        |
| Belgium     | 10592226 | Europe    | 79.641  | 33692.68568        |
| Benin       | 8078318  | Africa    | 56.728  | 1441.286873        |
| Bolivia     | 9139152  | American  | 65.554  | 3822.137864        |

|   |   |  |
|---|---|--|
| <div><div>JOIN OUR MAILING LIST</div><div>Sign up to stay in the loop with all things Plotly — from Dash Club to product updates, webinars, and more!</div><div><div>SUBSCRIBE</div><div>(HTTPS://GO.PLOT.LY/SUBSCRIPTION)</div></div></div>  | <div><div>Products</div><div>Dash (<a href="https://plotly.com/dash/">https://plotly.com/dash/</a>)</div><div>Consulting and Training (<a href="https://plotly.com/consulting-and-oem/">https://plotly.com/consulting-and-oem/</a>)</div></div>     | <div><div>Pricing</div><div>Enterprise Pricing (<a href="https://plotly.com/get-pricing/">https://plotly.com/get-pricing/</a>)</div></div> |
| <div><div>About Us</div><div>Careers (<a href="https://plotly.com/careers">https://plotly.com/careers</a>)</div><div>Resources (<a href="https://plotly.com/resources/">https://plotly.com/resources/</a>)</div><div>Blog (<a href="https://medium.com/@plotlygraphs">https://medium.com/@plotlygraphs</a>)</div></div> | <div><div>Support</div><div>Community Support (<a href="https://community.plot.ly/">https://community.plot.ly/</a>)</div><div>Documentation (<a href="https://plotly.com/graphing-libraries">https://plotly.com/graphing-libraries</a>)</div></div> |  |
| <div>Copyright © 2025 Plotly. All rights reserved.</div>  | <div><div>Terms of Service (<a href="https://community.plotly.com/tos">https://community.plotly.com/tos</a>)</div><div>Privacy Policy (<a href="https://plotly.com/privacy/">https://plotly.com/privacy/</a>)</div></div>                           |  |

