Ternary Plots in Python

How to make Ternary plots in Python with Plotly.

Ternary Plots

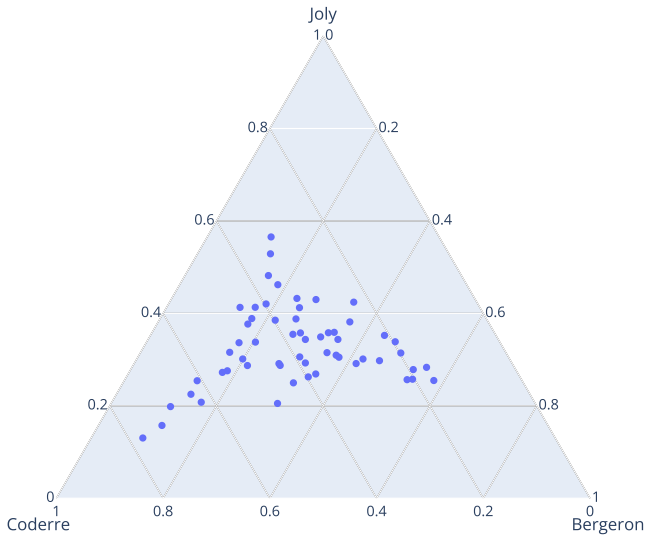
A ternary plot depicts the ratios of three variables as positions in an equilateral triangle.

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="Coderre", c="Bergeron")
fig.show()
```

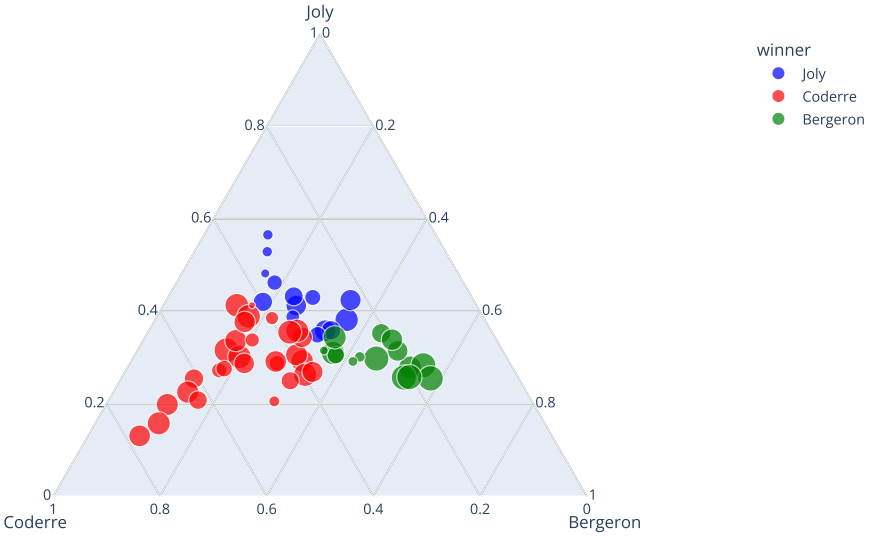


We can scale and color the markers to produce a ternary bubble chart.

```
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

ɔh

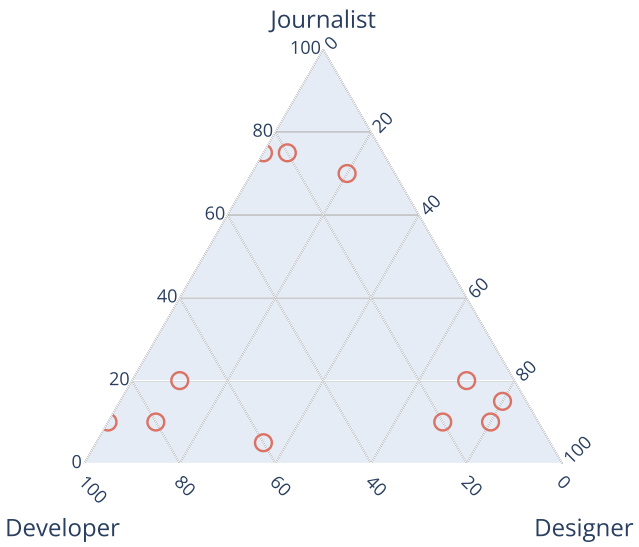


Ternary scatter plot with Plotly Graph Objects

```
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()
```

ess

ɔh



Reference

See function reference for px.(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/) 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
```

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)

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

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

Support

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

Terms of Service (https://community.plotly.com/tos) Privacy Policy (https://plotly.com/privacy/)