



# Ternary Overlay in Python

How to make a scatter plot overlaid on ternary contour 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](https://plotly.com/studio/?utm_medium=graphing-libraries&utm_campaign=studio_early_access&utm_content=sidebar))

## Load and Process Data Files

```
import json
import pandas as pd

contour_raw_data = pd.read_json('https://raw.githubusercontent.com/plotly/datasets/master/contour_data.json')
scatter_raw_data = pd.read_json('https://raw.githubusercontent.com/plotly/datasets/master/scatter_data.json')

scatter_data = scatter_raw_data['Data']

def clean_data(data_in):
    """
    Cleans data in a format which can be conveniently
    used for drawing traces. Takes a dictionary as the
    input, and returns a List in the following format:

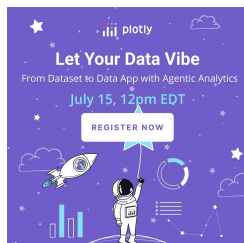
    input = {'key': ['a b c']}
    output = [key, [a, b, c]]
    """
    key = list(data_in.keys())[0]
    data_out = [key]
    for i in data_in[key]:
        data_out.append(list(map(float, i.split(' '))))

    return data_out

#Example:
print(clean_data({'L1': ['.03 0.5 0.47', '0.4 0.5 0.1']}))
```

```
['L1', [0.03, 0.5, 0.47], [0.4, 0.5, 0.1]]
```

## Create Ternary Scatter Plot:



```

import plotly.graph_objects as go

a_list = []
b_list = []
c_list = []
text = []

for raw_data in scatter_data:
    data = clean_data(raw_data)
    text.append(data[0])
    c_list.append(data[1][0])
    a_list.append(data[1][1])
    b_list.append(data[1][2])

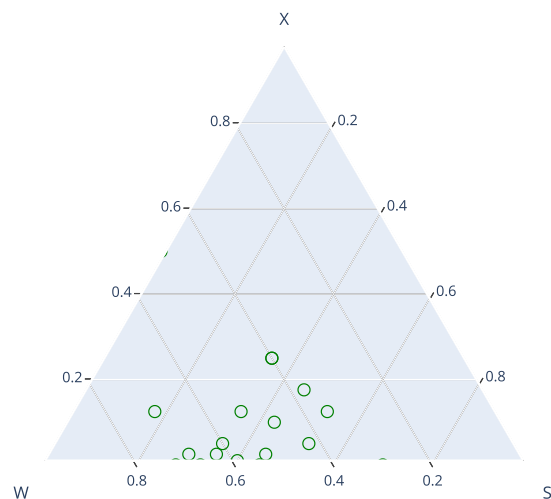
fig = go.Figure(go.Scatterternary(
    text=text,
    a=a_list,
    b=b_list,
    c=c_list,
    mode='markers',
    marker={
        'symbol': 100,
        'color': 'green',
        'size': 10},
))

fig.update_layout({
    'title': 'Ternary Scatter Plot',
    'ternary': {
        {
            'sum': 1,
            'aaxis': {'title': 'X', 'min': 0.01, 'linewidth': 2, 'ticks': 'outside' },
            'baxis': {'title': 'W', 'min': 0.01, 'linewidth': 2, 'ticks': 'outside' },
            'caxis': {'title': 'S', 'min': 0.01, 'linewidth': 2, 'ticks': 'outside' }
        },
        'showlegend': False
    })

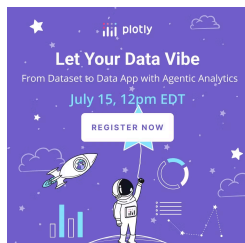
fig.show()

```

Ternary Scatter Plot



Create Ternary Contour Plot:



```
import plotly.graph_objects as go

contour_dict = contour_raw_data['Data']

# Defining a colormap:
colors = ['#8dd3c7', '#ffffb3', '#bebada',
          '#fb8072', '#80b1d3', '#fdb462',
          '#b3de69', '#fccde5', '#d9d9d9',
          '#bc80bd']
colors_iterator = iter(colors)

fig = go.Figure()

for raw_data in contour_dict:
    data = clean_data(raw_data)

    a = [inner_data[0] for inner_data in data[1:]]
    a.append(data[1][0]) # Closing the Loop

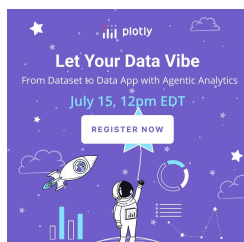
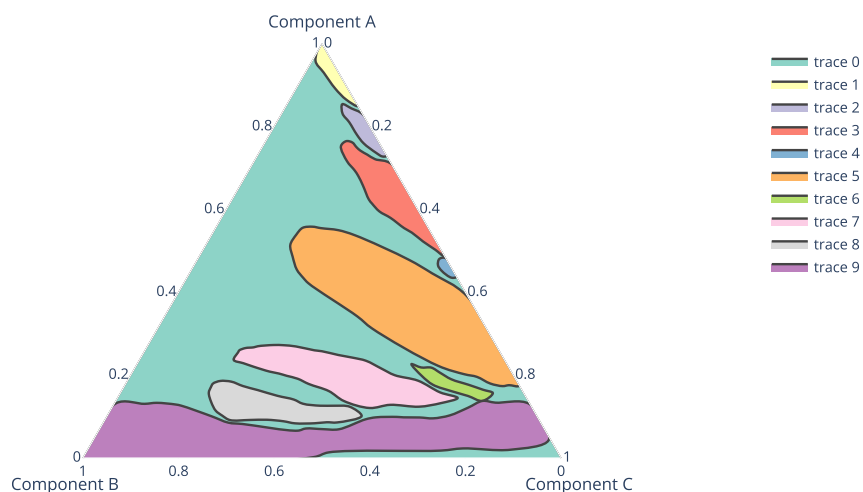
    b = [inner_data[1] for inner_data in data[1:]]
    b.append(data[1][1]) # Closing the Loop

    c = [inner_data[2] for inner_data in data[1:]]
    c.append(data[1][2]) # Closing the Loop

    fig.add_trace(go.Scatterternary(
        text = data[0],
        a=a, b=b, c=c, mode='lines',
        line=dict(color='#444', shape='spline'),
        fill='toself',
        fillcolor = colors_iterator.__next__()
    ))

fig.update_layout(title = 'Ternary Contour Plot')
fig.show()
```

Ternary Contour Plot



# 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
```



## 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	36326.4927
Bahrain	706573	Asia	75.635	29796.04834
Bangladesh	150448339	Asia	64.062	1701.253792
Belgium	10391226	Europe	79.441	33062.04908
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](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/>)

