



Figure Factory Subplots in Python

Subplots with Plotly Figure Factory Charts

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)

Plotly's Figure Factory Module

Plotly's Python API contains a [figure factory module \(/python/figure-factories/\)](#) which includes many wrapper functions that create unique chart types that are not yet included in [plotly.js \(https://github.com/plotly/plotly.js\)](#), Plotly's open-source graphing library. The figure factory functions create a full figure, so some Plotly features, such as subplotting, should be implemented slightly differently with these charts.

Vertical Figure Factory Charts

First create the figures that you'd like to appear in the subplot:

```
import plotly.figure_factory as ff
import plotly.graph_objects as go

import numpy as np

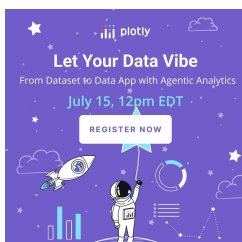
## Create first figure
x1,y1 = np.meshgrid(np.arange(0, 2, .2), np.arange(0, 2, .2))
u1 = np.cos(x1)*y1
v1 = np.sin(x1)*y1

fig1 = ff.create_quiver(x1, y1, u1, v1, name='Quiver')

## Create second figure
x = np.linspace(-3, 3, 100)
y = np.linspace(-3, 3, 100)
Y, X = np.meshgrid(x, y)
u = -1 - X**2 + Y
v = 1 + X - Y**2

fig2 = ff.create_streamline(x, y, u, v, arrow_scale=.1, name='Streamline')
```

Edit the figures' x and y axes attributes to create subplots:



```

for i in range(len(fig1.data)):
    fig1.data[i].xaxis='x1'
    fig1.data[i].yaxis='y1'

fig1.layout.xaxis1.update({'anchor': 'y1'})
fig1.layout.yaxis1.update({'anchor': 'x1', 'domain': [.55, 1]})

for i in range(len(fig2.data)):
    fig2.data[i].xaxis='x2'
    fig2.data[i].yaxis='y2'

# initialize xaxis2 and yaxis2
fig2['layout']['xaxis2'] = {}
fig2['layout']['yaxis2'] = {}

fig2.layout.xaxis2.update({'anchor': 'y2'})
fig2.layout.yaxis2.update({'anchor': 'x2', 'domain': [0, .45]})

```

```

layout.YAxis({
    'anchor': 'x2', 'domain': [0, 0.45]
})

```

Combine the data and layout objects to create a figure

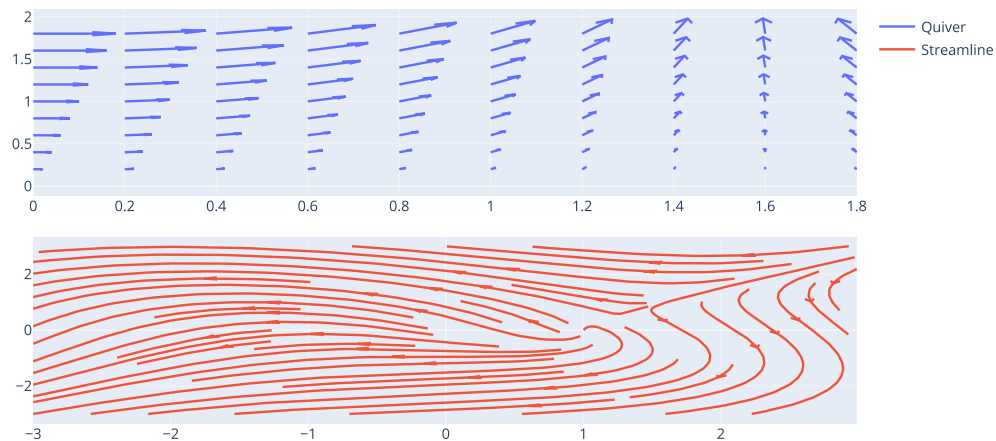
```

fig = go.Figure()
fig.add_traces([fig1.data[0], fig2.data[0]])

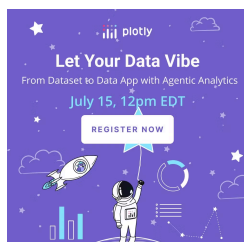
fig.layout.update(fig1.layout)
fig.layout.update(fig2.layout)

fig.show()

```



Horizontal Table and Chart



```

import plotly.graph_objects as go
import plotly.figure_factory as ff

table_data = [['Team', 'Wins', 'Losses', 'Ties'],
               ['Montréal<br>Canadiens', 18, 4, 0],
               ['Dallas Stars', 18, 5, 0],
               ['NY Rangers', 16, 5, 0],
               ['Boston<br>Bruins', 13, 8, 0],
               ['Chicago<br>Blackhawks', 13, 8, 0],
               ['LA Kings', 13, 8, 0],
               ['Ottawa<br>Senators', 12, 5, 0]]

fig = ff.create_table(table_data, height_constant=60)

teams = ['Montréal Canadiens', 'Dallas Stars', 'NY Rangers',
         'Boston Bruins', 'Chicago Blackhawks', 'LA Kings', 'Ottawa Senators']
GFGP = [3.54, 3.48, 3.0, 3.27, 2.83, 2.45, 3.18]
GAPG = [2.17, 2.57, 2.0, 2.91, 2.57, 2.14, 2.77]

trace1 = go.Scatter(x=teams, y=GFGP,
                    marker=dict(color='#0099ff'),
                    name='Goals For<br>Per Game',
                    xaxis='x2', yaxis='y2')
trace2 = go.Scatter(x=teams, y=GAPG,
                    marker=dict(color='#404040'),
                    name='Goals Against<br>Per Game',
                    xaxis='x2', yaxis='y2')

fig.add_traces([trace1, trace2])

# initialize xaxis2 and yaxis2
fig['layout']['xaxis2'] = {}
fig['layout']['yaxis2'] = {}

# Edit layout for subplots
fig.layout.xaxis.update({'domain': [0, .5]})
fig.layout.xaxis2.update({'domain': [0.6, 1.]})

# The graph's yaxis MUST BE anchored to the graph's xaxis
fig.layout.yaxis2.update({'anchor': 'x2'})
fig.layout.yaxis2.update({'title': 'Goals'})

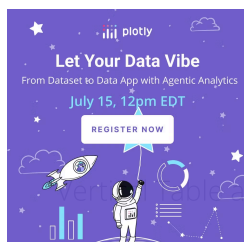
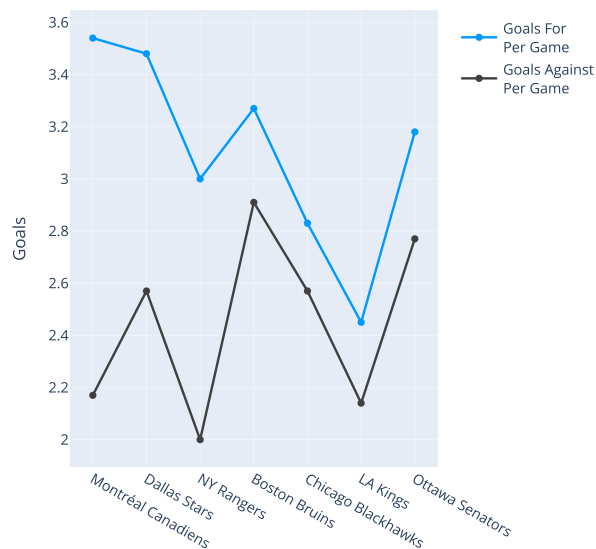
# Update the margins to add a title and see graph x-Labels.
fig.layout.margin.update({'t':50, 'b':100})
fig.layout.update({'title': '2016 Hockey Stats'})

fig.show()

```

2016 Hockey Stats

Team	Wins	Losses	Ties
Montréal Canadiens	18	4	0
Dallas Stars	18	5	0
NY Rangers	16	5	0
Boston Bruins	13	8	0
Chicago Blackhawks	13	8	0
LA Kings	13	8	0
Ottawa Senators	12	5	0



and Chart

```

import plotly.graph_objects as go
import plotly.figure_factory as ff

# Add table data
table_data = [['Team', 'Wins', 'Losses', 'Ties'],
               ['Montréal<br>Canadiens', 18, 4, 0],
               ['Dallas Stars', 18, 5, 0],
               ['NY Rangers', 16, 5, 0],
               ['Boston<br>Bruins', 13, 8, 0],
               ['Chicago<br>Blackhawks', 13, 8, 0],
               ['Ottawa<br>Senators', 12, 5, 0]]

# Initialize a figure with ff.create_table(table_data)
fig = ff.create_table(table_data, height_constant=60)

# Add graph data
teams = ['Montréal Canadiens', 'Dallas Stars', 'NY Rangers',
         'Boston Bruins', 'Chicago Blackhawks', 'Ottawa Senators']
GFGP = [3.54, 3.48, 3.0, 3.27, 2.83, 3.18]
GAPG = [2.17, 2.57, 2.0, 2.91, 2.57, 2.77]

# Make traces for graph
trace1 = go.Bar(x=teams, y=GFGP, xaxis='x2', yaxis='y2',
                 marker=dict(color='#0099ff'),
                 name='Goals For<br>Per Game')
trace2 = go.Bar(x=teams, y=GAPG, xaxis='x2', yaxis='y2',
                 marker=dict(color='#404040'),
                 name='Goals Against<br>Per Game')

# Add trace data to figure
fig.add_traces([trace1, trace2])

# initialize xaxis2 and yaxis2
fig['layout']['xaxis2'] = {}
fig['layout']['yaxis2'] = {}

# Edit layout for subplots
fig.layout.yaxis.update({'domain': [0, .45]})
fig.layout.yaxis2.update({'domain': [.6, 1]})

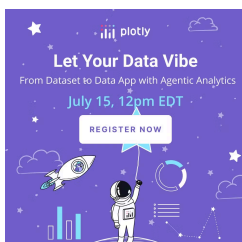
# The graph's yaxis2 MUST BE anchored to the graph's xaxis2 and vice versa
fig.layout.yaxis2.update({'anchor': 'x2'})
fig.layout.xaxis2.update({'anchor': 'y2'})
fig.layout.yaxis2.update({'title': 'Goals'})

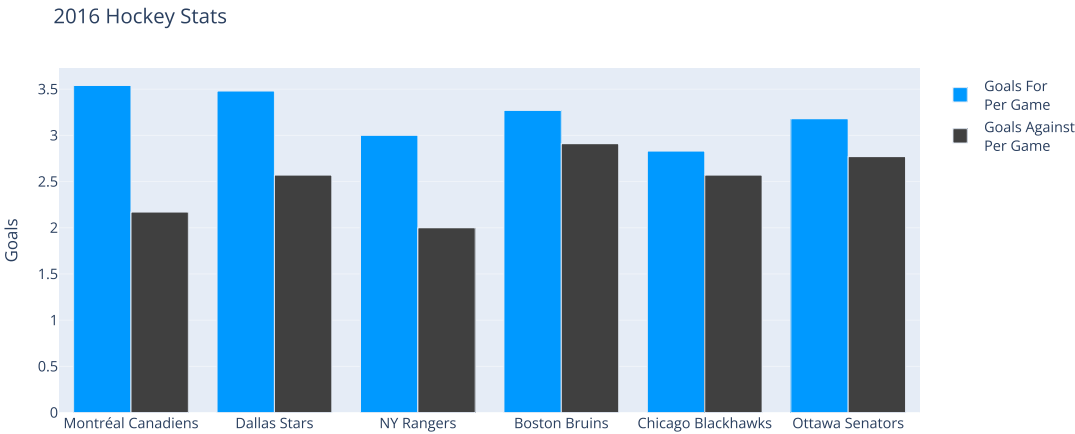
# Update the margins to add a title and see graph x-Labels.
fig.layout.margin.update({'t':75, 'l':50})
fig.layout.update({'title': '2016 Hockey Stats'})

# Update the height because adding a graph vertically will interact with
# the plot height calculated for the table
fig.layout.update({'height':800})

# Plot!
fig.show()

```





Team	Wins	Losses	Ties
Montréal Canadiens	18	4	0
Dallas Stars	18	5	0
NY Rangers	16	5	0
Boston Bruins	13	8	0
Chicago Blackhawks	13	8	0
Ottawa Senators	12	5	0

Reference

See <https://plotly.com/python/subplots/> (<https://plotly.com/python/subplots/>) for more information on working with subplots. See <https://plotly.com/python/reference/> (<https://plotly.com/python/reference/>) for more information regarding chart attributes!

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>).

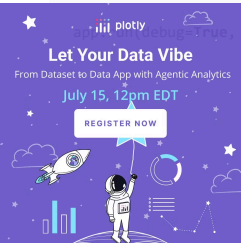
Everywhere in this page that you see `fig.show()`, you can display the same figure in a Dash application by passing it to the figure argument of the [Graph component](https://dash.plot.ly/dash-core-components/graph) (<https://dash.plot.ly/dash-core-components/graph>) from the built-in `dash_core_components`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

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	2600522	Europe	76.422	5937.029525999999
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	29796.04834
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 / 12



(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/>)

