

Bullet Charts in Python

How to make bullet charts 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)

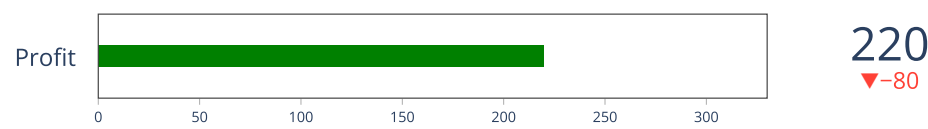
Basic Bullet Charts

Stephen Few's Bullet Chart was invented to replace dashboard gauges (https://plotly.com/python/gauge-charts/) and meters, combining both types of charts into simple bar charts with qualitative bars (steps), quantitative bar (bar) and performance line (threshold); all into one simple layout. Steps typically are broken into several values, which are defined with an array. The bar represent the actual value that a particular variable reached, and the threshold usually indicate a goal point relative to the value achieved by the bar. See [indicator page](https://plotly.com/python/gauge-charts/) (https://plotly.com/python/gauge-charts/) for more detail.

```
import plotly.graph_objects as go

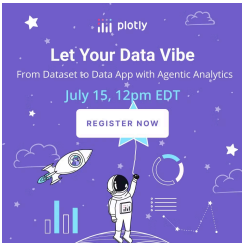
fig = go.Figure(go.Indicator(
    mode = "number+gauge+delta",
    gauge = {'shape': "bullet"},
    value = 220,
    delta = {'reference': 300},
    domain = {'x': [0, 1], 'y': [0, 1]},
    title = {'text': "Profit"}))
fig.update_layout(height = 250)

fig.show()
```



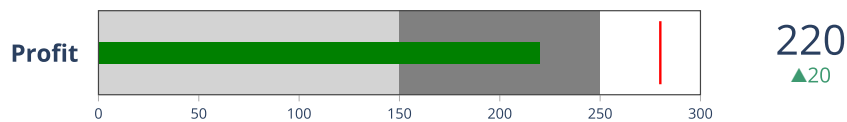
Add Steps, and Threshold

Below is the same example using "steps" attribute, which is shown as shading, and "threshold" to determine boundaries that visually alert you if the value cross a defined threshold.



```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "number+gauge+delta", value = 220,
    domain = {'x': [0.1, 1], 'y': [0, 1]},
    title = {'text' : "<b>Profit</b>"},
    delta = {'reference': 200},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "red", 'width': 2},
            'thickness': 0.75,
            'value': 280},
        'steps': [
            {'range': [0, 150], 'color': "lightgray"},
            {'range': [150, 250], 'color': "gray"}])])
fig.update_layout(height = 250)
fig.show()
```

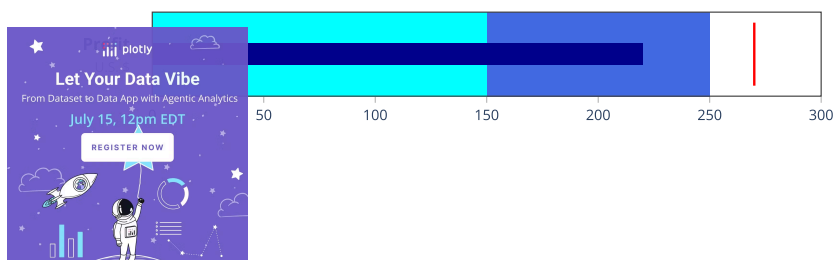


Custom Bullet

The following example shows how to customize your charts. For more information about all possible options check our [reference page](https://plotly.com/python/reference/indicator/) (<https://plotly.com/python/reference/indicator/>).

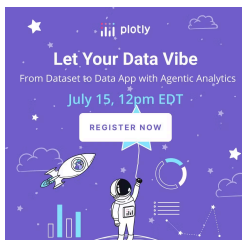
```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "number+gauge+delta", value = 220,
    domain = {'x': [0, 1], 'y': [0, 1]},
    delta = {'reference': 280, 'position': "top"},
    title = {'text': "<b>Profit</b><br><span style='color: gray; font-size: 0.8em'>U.S. $</span>", 'font': {'size': 14}},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "red", 'width': 2},
            'thickness': 0.75, 'value': 270},
        'bgcolor': "white",
        'steps': [
            {'range': [0, 150], 'color': "cyan"},
            {'range': [150, 250], 'color': "royalblue"}],
        'bar': {'color': "darkblue"}}))
fig.update_layout(height = 250)
fig.show()
```



Multi Bullet

Bullet charts can be stacked for comparing several values at once as illustrated below:



```
import plotly.graph_objects as go

fig = go.Figure()

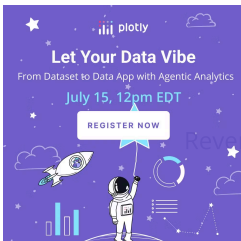
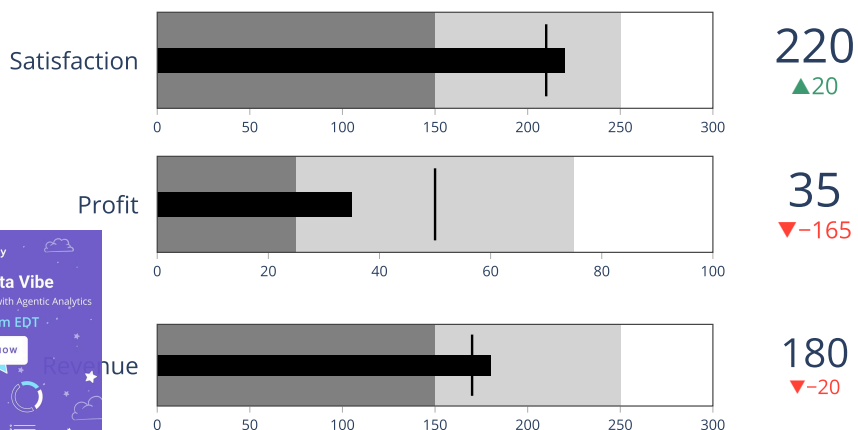
fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 180,
    delta = {'reference': 200},
    domain = {'x': [0.25, 1], 'y': [0.08, 0.25]},
    title = {'text': "Revenue"},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "black", 'width': 2},
            'thickness': 0.75,
            'value': 170},
        'steps': [
            {'range': [0, 150], 'color': "gray"},
            {'range': [150, 250], 'color': "lightgray"}],
        'bar': {'color': "black"}}))

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 35,
    delta = {'reference': 200},
    domain = {'x': [0.25, 1], 'y': [0.4, 0.6]},
    title = {'text': "Profit"},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 100]},
        'threshold': {
            'line': {'color': "black", 'width': 2},
            'thickness': 0.75,
            'value': 50},
        'steps': [
            {'range': [0, 25], 'color': "gray"},
            {'range': [25, 75], 'color': "lightgray"}],
        'bar': {'color': "black"}}))

fig.add_trace(go.Indicator(
    mode = "number+gauge+delta", value = 220,
    delta = {'reference': 200},
    domain = {'x': [0.25, 1], 'y': [0.7, 0.9]},
    title = {'text': "Satisfaction"},
    gauge = {
        'shape': "bullet",
        'axis': {'range': [None, 300]},
        'threshold': {
            'line': {'color': "black", 'width': 2},
            'thickness': 0.75,
            'value': 210},
        'steps': [
            {'range': [0, 150], 'color': "gray"},
            {'range': [150, 250], 'color': "lightgray"}],
        'bar': {'color': "black"}}))

fig.update_layout(height = 400 , margin = {'t':0, 'b':0, 'l':0})

fig.show()
```



Reference

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

My First App with Data, Graph, and Controls

pop

lifeExp

gdpPerCap

country	pop	continent	lifeExp	gdpPerCap
Afghanistan	31889923	Asia	43.828	974.5883384
Albania	3009523	Europe	76.422	5937.92520999999
Algeria	33333216	Africa	72.361	6223.367465
Angola	18228676	Africa	42.731	4797.231267
Argentina	40101927	Americas	75.32	12779.37964
Australia	20434376	Oceania	81.235	34435.367439999995
Austria	8199783	Europe	79.829	36126.4927
Bahrain	708573	Asia	75.635	29796.44834
Bangladesh	156448339	Asia	64.062	1301.253792
Belgium	10392226	Europe	79.441	33892.48588
Benin	8078334	Africa	56.728	1441.284873
Bolivia	9139532	Americas	65.554	3822.137884

(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://plotly.com/blog/](#))

Let Your Data Vibe
From Dataset to Data App with Agentic Analytics
July 15, 12pm EDT
REGISTER NOW

Products

Dash (<https://plotly.com/dash/>)

Consulting and Training (<https://plotly.com/consulting-and-oem/>)

Pricing

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

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/>)