



Indicators in Python

How to make gauge 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)

Overview

In this tutorial we introduce a new trace named "Indicator". The purpose of "indicator" is to visualize a single value specified by the "value" attribute. Three distinct visual elements are available to represent that value: number, delta and gauge. Any combination of them can be specified via the "mode" attribute. Top-level attributes are:

1. value: the value to visualize
2. mode: which visual elements to draw
3. align: how to align number and delta (left, center, right)
4. domain: the extent of the figure

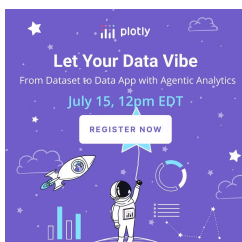
Then we can configure the 3 different visual elements via their respective container:

1. number is simply a representation of the number in text. It has attributes:
2. valueformat: to format the number
3. prefix: a string before the number
4. suffix: a string after the number
5. font.(family|size): to control the font

"delta" simply displays the difference between the value with respect to a reference. It has attributes:

1. reference: the number to compare the value with
2. relative: whether that difference is absolute or relative
3. valueformat: to format the delta
4. (increasing|decreasing).color: color to be used for positive or decreasing delta
5. (increasing|decreasing).symbol: symbol displayed on the left of the delta
6. font.(family|size): to control the font
7. position: position relative to vmber (either top, left, bottom, right)
8. prefix: a string to appear before the delta
9. suffix: a string to appear after the delta

Finally, we can have a simple title for the indicator via title with 'text' attribute which is a string, and 'align' which can be set to left, center, and right. There are two gauge types: [angular](https://plotly.com/python/gauge-charts/) (<https://plotly.com/python/gauge-charts/>) and [bullet](https://plotly.com/python/bullet-charts/) (<https://plotly.com/python/bullet-charts/>). Here is a combination of both shapes (angular, bullet), and different modes (gauge, delta, and value):



```
import plotly.graph_objects as go

fig = go.Figure()

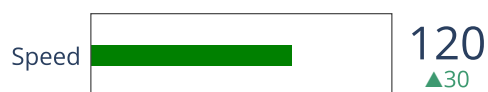
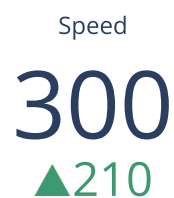
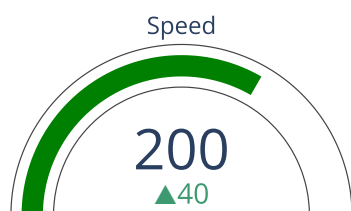
fig.add_trace(go.Indicator(
    value = 200,
    delta = {'reference': 160},
    gauge = {
        'axis': {'visible': False}},
    domain = {'row': 0, 'column': 0}))

fig.add_trace(go.Indicator(
    value = 120,
    gauge = {
        'shape': "bullet",
        'axis': {'visible': False}},
    domain = {'x': [0.05, 0.5], 'y': [0.15, 0.35]}))

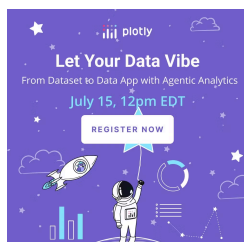
fig.add_trace(go.Indicator(
    mode = "number+delta",
    value = 300,
    domain = {'row': 0, 'column': 1}))

fig.add_trace(go.Indicator(
    mode = "delta",
    value = 40,
    domain = {'row': 1, 'column': 1}))

fig.update_layout(
    grid = {'rows': 2, 'columns': 2, 'pattern': "independent"},
    template = {'data': {'indicator': [{
        'title': {'text': "Speed"},
        'mode': "number+delta+gauge",
        'delta': {'reference': 90}}]
    })})
```



A Single Angular Gauge Chart

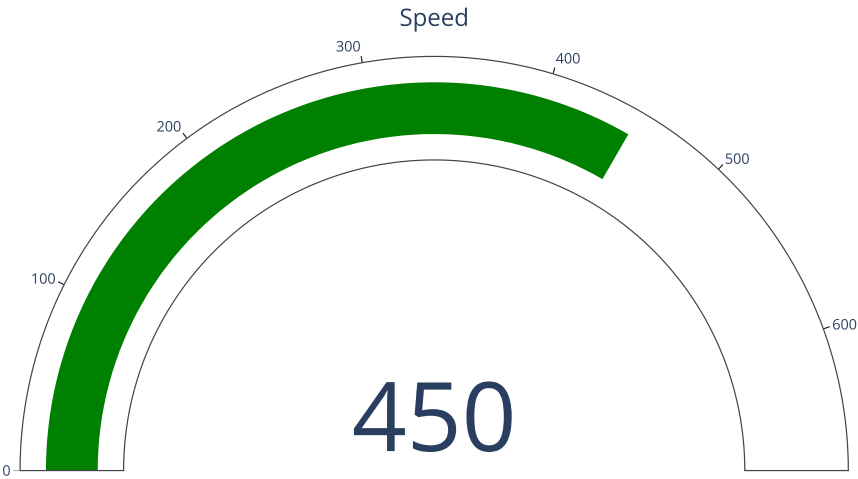


```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "gauge+number",
    value = 450,
    title = {'text': "Speed"},
    domain = {'x': [0, 1], 'y': [0, 1]}
))

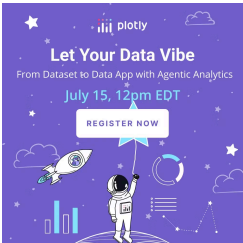
fig.show()
```

part
ers



Bullet Gauge

The equivalent of above "angular gauge":



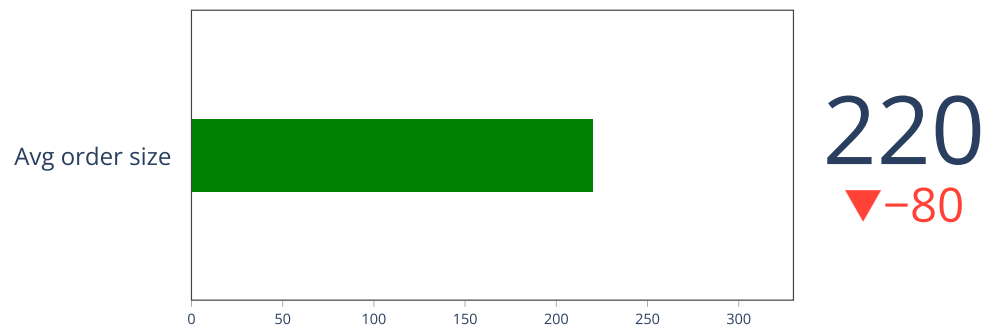
```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "number+gauge+delta",
    gauge = {'shape': "bullet"},
    delta = {'reference': 300},
    value = 220,
    domain = {'x': [0.1, 1], 'y': [0.2, 0.9]},
    title = {'text': "Avg order size"}))

fig.show()
```

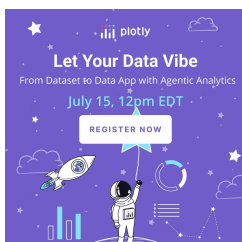
part

ers



Showing Information above Your Chart

Another interesting feature is that indicator trace sits above the other traces (even the 3d ones). This way, it can be easily used as an overlay as demonstrated below



```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "number+delta",
    value = 492,
    delta = {"reference": 512, "valueformat": ".0f"},
    title = {"text": "Users online"},
    domain = {'y': [0, 1], 'x': [0.25, 0.75]}))

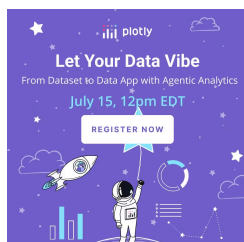
fig.add_trace(go.Scatter(
    y = [325, 324, 405, 400, 424, 404, 417, 432, 419, 394, 410, 426, 413, 419, 404, 408, 401, 377, 368, 361, 356, 359, 375, 397, 394, 418, 437, 450, 430, 442, 424, 443, 420, 418, 423, 423, 426, 440, 437, 436, 447, 460, 478, 472, 450, 456, 436, 418, 429, 412, 429, 442, 464, 447, 434, 457, 474, 480, 499, 497, 480, 502, 512, 492]))

fig.update_layout(xaxis = {'range': [0, 62]})
fig.show()
```



Data Cards / Big Numbers

Data card helps to display more contextual information about the data. Sometimes one number is all you want to see in a report, such as total sales, annual revenue, etc. This example shows how to visualize these big numbers:



```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "number+delta",
    value = 400,
    number = {'prefix': "$"},
    delta = {'position': "top", 'reference': 320},
    domain = {'x': [0, 1], 'y': [0, 1]}))

fig.update_layout(paper_bgcolor = "lightgray")

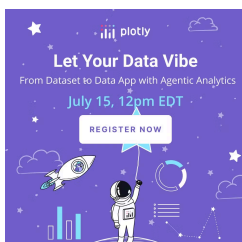
fig.show()
```

part

ers

▲80
\$400

It's possible to display several numbers



```
import plotly.graph_objects as go

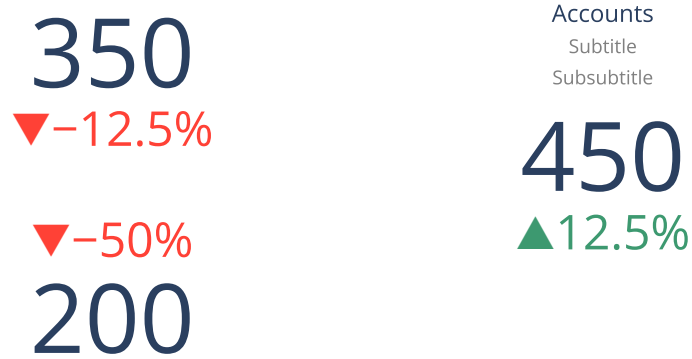
fig = go.Figure()

fig.add_trace(go.Indicator(
    mode = "number+delta",
    value = 200,
    domain = {'x': [0, 0.5], 'y': [0, 0.5]},
    delta = {'reference': 400, 'relative': True, 'position': "top"}))

fig.add_trace(go.Indicator(
    mode = "number+delta",
    value = 350,
    delta = {'reference': 400, 'relative': True},
    domain = {'x': [0, 0.5], 'y': [0.5, 1]}))

fig.add_trace(go.Indicator(
    mode = "number+delta",
    value = 450,
    title = {"text": "Accounts<br><span style='font-size:0.8em;color:gray'>Subtitle</span><br><span style='font-size:0.8em;color:gray'>Subsubtitle</span>"},
    delta = {'reference': 400, 'relative': True},
    domain = {'x': [0.6, 1], 'y': [0, 1]}))

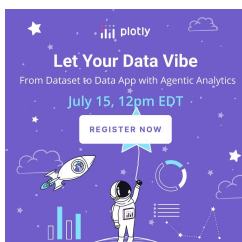
fig.show()
```



Adding a Prefix and Suffix

On both a number and a delta, you can add a string to appear before the value using prefix. You can add a string to appear after the value using suffix. In the following example, we add '\$' as a prefix and 'm' as suffix for both the number and delta.

Note: suffix and prefix on delta are new in 5.10



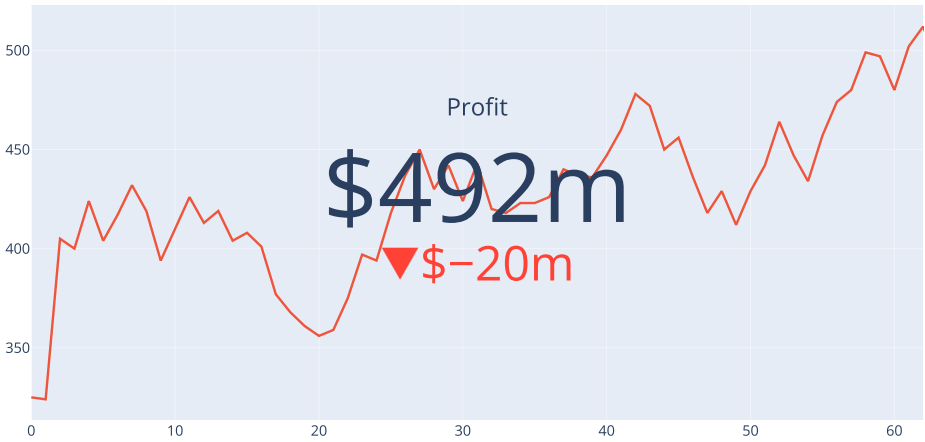
part
ers

```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "number+delta",
    value = 492,
    number = {"prefix": "$", "suffix": "m"},
    delta = {"reference": 512, "valueformat": ".0f", "prefix": "$", "suffix": "m"},
    title = {"text": "Profit"},
    domain = {'y': [0, 1], 'x': [0.25, 0.75]}))

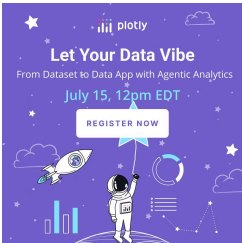
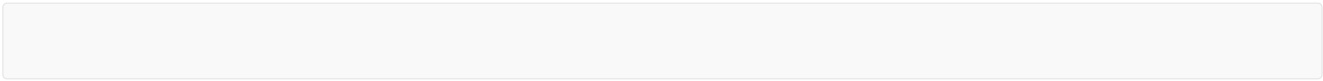
fig.add_trace(go.Scatter(
    y = [325, 324, 405, 400, 424, 404, 417, 432, 419, 394, 410, 426, 413, 419, 404, 408, 401, 377, 368, 361, 356, 359, 375, 397, 394, 418, 437, 450, 430, 442, 424, 443, 420, 418, 423, 423, 426, 440, 437, 436, 447, 460, 478, 472, 450, 456, 436, 418, 429, 412, 429, 442, 464, 447, 434, 457, 474, 480, 499, 497, 480, 502, 512, 492]))

fig.update_layout(xaxis = {'range': [0, 62]})
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/>) 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:

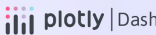
part
ers

```
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	36126.4927
Bahrain	706573	Asia	75.635	29796.04834
Bangladesh	150448339	Asia	64.062	1701.253792
Belgium	10391226	Europe	79.441	33062.48908
Benin	8878314	Africa	56.728	1441.284873
Bolivia	9119152	Americas	65.554	3821.137884



(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://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/>)

