Gauge Charts in Python

How to make gauge meter charts in Python with Plotly.

Basic Gauge

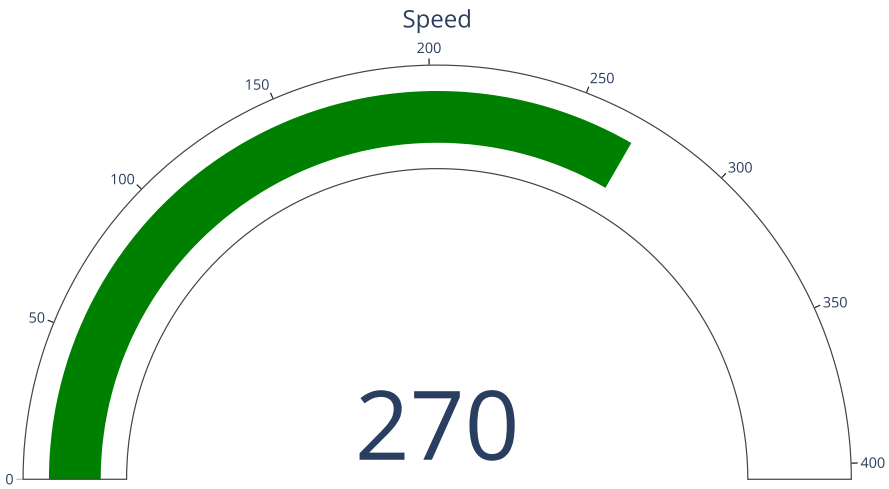
A radial gauge chart has a circular arc, which displays a single value to estimate progress toward a goal. The bar shows the target value, and the shading represents the progress toward that goal. Gauge charts, known as speedometer charts as well. This chart type is usually used to illustrate key business indicators.

The example below displays a basic gauge chart with default attributes. For more information about different added attributes check indicator (https://plotly.com/python/indicator/) tutorial.

```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "gauge+number",
    value = 270,
    domain = {'x': [0, 1], 'y': [0, 1]},
    title = {'text': "Speed"}))

fig.show()
```



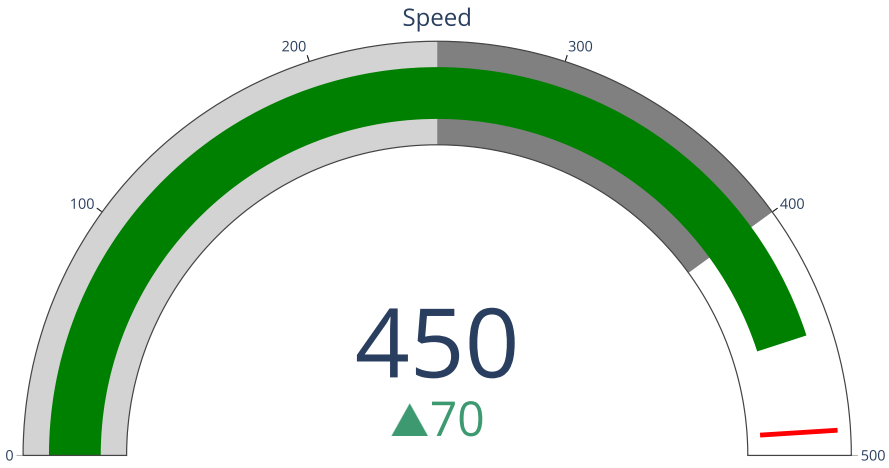
Add Steps, Threshold, and Delta

The following examples include "steps" attribute shown as shading inside the radial arc, "delta" which is the difference of the value and goal (reference - value), and [illegible] boundaries that visually alert you if the value cross a defined threshold.

```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    domain = {'x': [0, 1], 'y': [0, 1]},
    value = 450,
    mode = "gauge+number+delta",
    title = {'text': "Speed"},
    delta = {'reference': 380},
    gauge = {'axis': {'range': [None, 500]},
             'steps' : [
                 {'range': [0, 250], 'color': "lightgray"},
                 {'range': [250, 400], 'color': "gray"}],
             'threshold' : {'line': {'color': "red", 'width': 4}, 'thickness': 0.75, 'value': 490}}))

fig.show()
```



Custom Gauge Chart

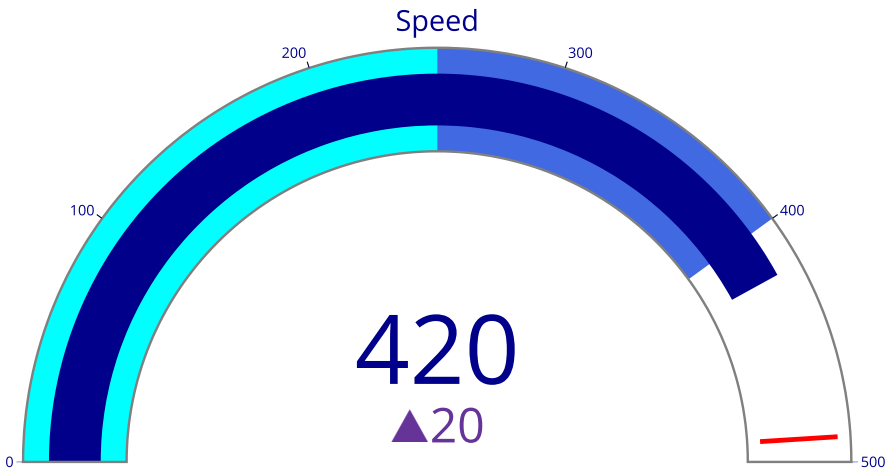
The following example shows how to style your gauge charts. For more information about all possible options check our reference page (https://plotly.com/python/reference/indicator/).

```
import plotly.graph_objects as go

fig = go.Figure(go.Indicator(
    mode = "gauge+number+delta",
    value = 420,
    domain = {'x': [0, 1], 'y': [0, 1]},
    title = {'text': "Speed", 'font': {'size': 24}},
    delta = {'reference': 400, 'increasing': {'color': "RebeccaPurple"}},
    gauge = {
        'axis': {'range': [None, 500], 'tickwidth': 1, 'tickcolor': "darkblue"},
        'bar': {'color': "darkblue"},
        'bgcolor': "white",
        'borderwidth': 2,
        'bordercolor': "gray",
        'steps': [
            {'range': [0, 250], 'color': 'cyan'},
            {'range': [250, 400], 'color': 'royalblue'}],
        'threshold': {
            'line': {'color': "red", 'width': 4},
            'thickness': 0.75,
            'value': 490}}))

fig.update_layout(paper_bgcolor = "lavender", font = {'color': "darkblue", 'family': "Arial"})

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:

```
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
```