Python (/python) > 3D Charts (/python/3d-charts) > 3D Bubble Charts

Suggest an edit to this page (https://github.com/plotly/plotly.py/edit/doc-prod/doc/python/3d-bubble-charts.md)

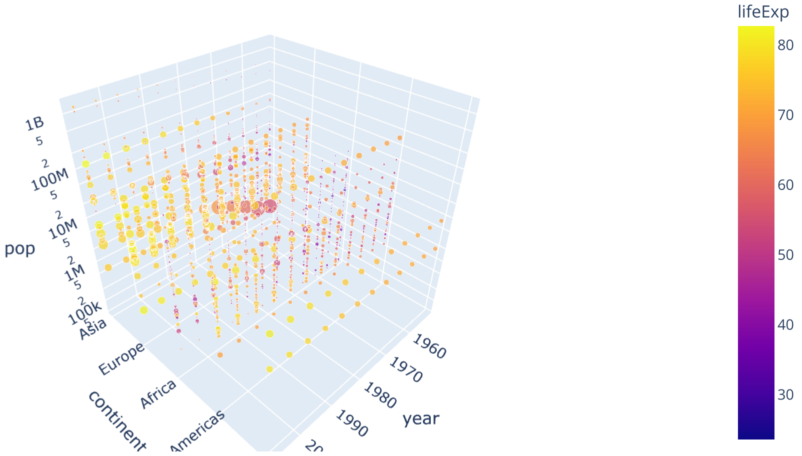
3D Bubble Charts in Python

How to make 3D Bubble Charts in Python with Plotly. Three examples of 3D Bubble 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)

3d Bubble chart with Plotly Express

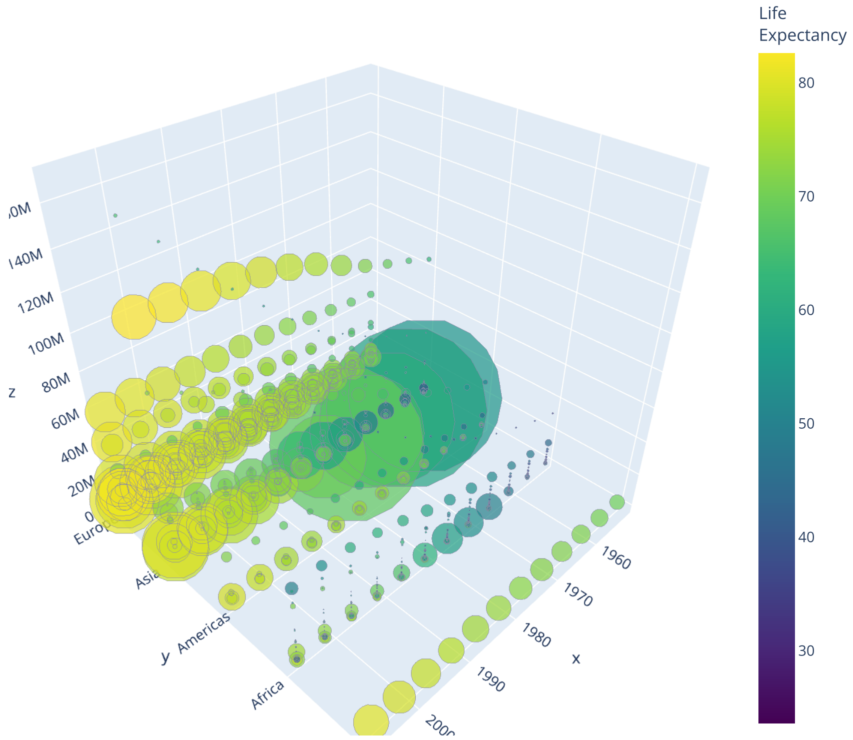
```
import plotly.express as px
import numpy as np
df = px.data.gapminder()
fig = px.scatter_3d(df, x='year', y='continent', z='pop', size='gdpPercap', color='lifeExp',
                    hover_data=['country'])
fig.update_layout(scene_zaxis_type="log")
fig.show()
```



Simple Bubble Chart

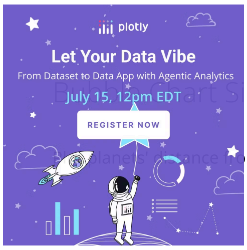
```
import plotly.graph_objects as go

import pandas as pd

# Get Data: this ex will only use part of it (i.e. rows 750-1500)
df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/gapminderDataFiveYear.csv')

start, end = 750, 1500

fig = go.Figure(data=go.Scatter3d(
    x=df['year'][start:end],
    y=df['continent'][start:end],
    z=df['pop'][start:end],
    text=df['country'][start:end],
    mode='markers',
    marker=dict(
        sizemode='diameter',
        sizeref=750,
        size=df['gdpPercap'][start:end],
        color = df['lifeExp'][start:end],
        colorscale = 'Viridis',
        colorbar_title = 'Life<br>Expectancy',
        line_color='rgb(140, 140, 170)'
    )
))


fig.update_layout(height=800, width=800,
                  title=dict(text='Examining Population and Life Expectancy Over Time'))

fig.show()
```

Examining Population and Life Expectancy Over Time

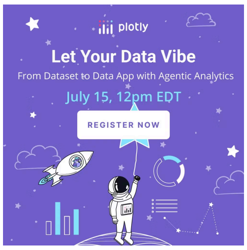


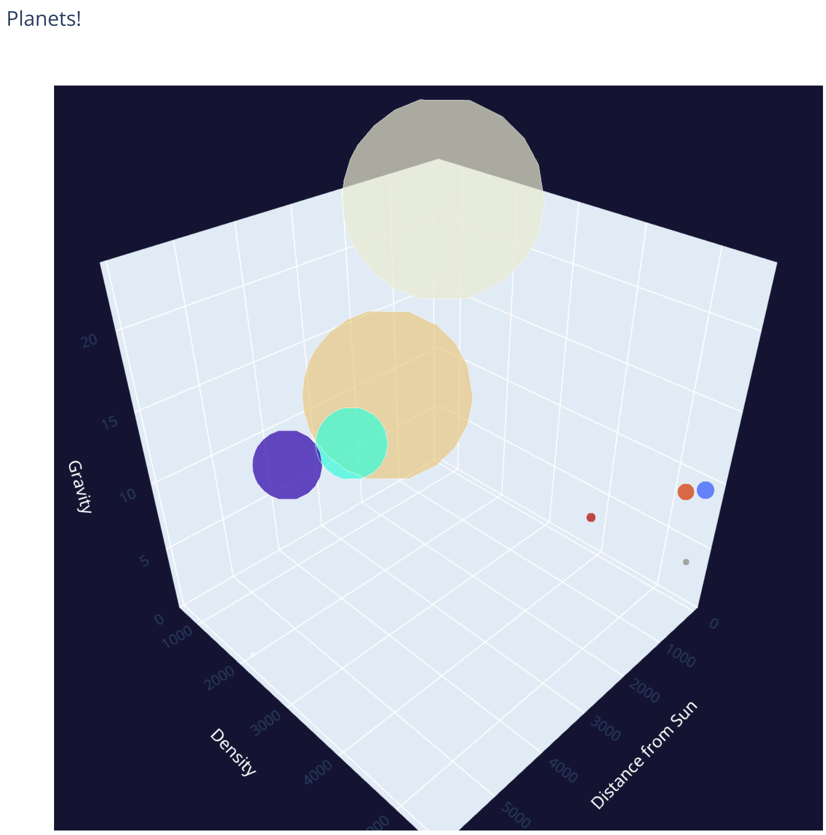
zed by a Variable

m sun, density, and gravity with bubble size based on planet size



```python
import plotly.graph_objects as go

planets = ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune', 'Pluto']
planet_colors = ['rgb(135, 135, 125)', 'rgb(210, 50, 0)', 'rgb(50, 90, 255)',
                 'rgb(178, 0, 0)', 'rgb(235, 235, 210)', 'rgb(235, 205, 130)',
                 'rgb(55, 255, 217)', 'rgb(38, 0, 171)', 'rgb(255, 255, 255)']
distance_from_sun = [57.9, 108.2, 149.6, 227.9, 778.6, 1433.5, 2872.5, 4495.1, 5906.4]
density = [5427, 5243, 5514, 3933, 1326, 687, 1271, 1638, 2095]
gravity = [3.7, 8.9, 9.8, 3.7, 23.1, 9.0, 8.7, 11.0, 0.7]
planet_diameter = [4879, 12104, 12756, 6792, 142984, 120536, 51118, 49528, 2370]

# Create trace, sizing bubbles by planet diameter
fig = go.Figure(data=go.Scatter3d(
    x = distance_from_sun,
    y = density,
    z = gravity,
    text = planets,
    mode = 'markers',
    marker = dict(
        sizemode = 'diameter',
        sizeref = 750, # info on sizeref: https://plotly.com/python/reference/scatter/#scatter-marker-sizeref
        size = planet_diameter,
        color = planet_colors,
        )
))

fig.update_layout(
    width=800,
    height=800,
    title=dict(text="Planets!"),
    scene=dict(
        xaxis=dict(
            title=dict(
                text="Distance from Sun",
                font=dict(
                    color="white"
                )
            )
        ),
        yaxis=dict(
            title=dict(
                text="Density",
                font=dict(
                    color="white"
                )
            )
        ),
        zaxis=dict(
            title=dict(
                text="Gravity",
                font=dict(
                    color="white"
                )
            )
        ),
        bgcolor="rgb(20, 24, 54)"
    )
)

fig.show()
```

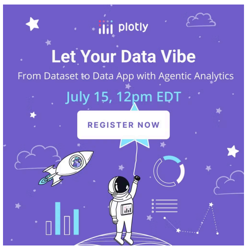


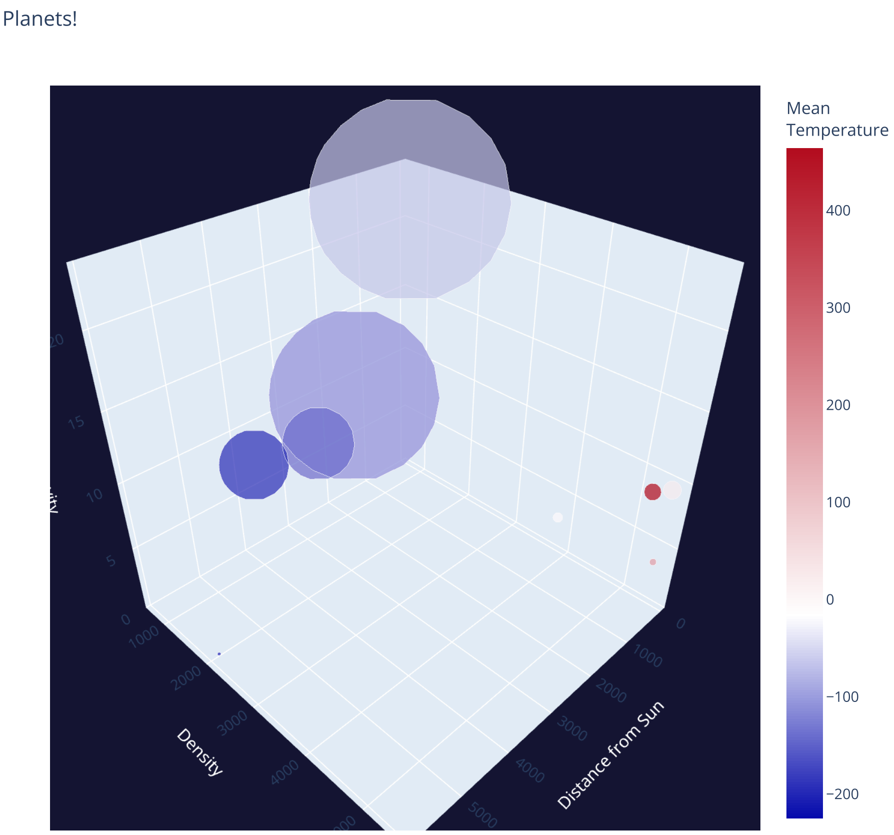


Edit the Colorbar

Plot planets' distance from sun, density, and gravity with bubble size based on planet size

```
import plotly.graph_objects as go

planets = ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune', 'Pluto']
temperatures = [167, 464, 15, -20, -65, -110, -140, -195, -200, -225]
distance_from_sun = [57.9, 108.2, 149.6, 227.9, 778.6, 1433.5, 2872.5, 4495.1, 5906.4]
density = [5427, 5243, 5514, 3933, 1326, 687, 1271, 1638, 2095]
gravity = [3.7, 8.9, 9.8, 3.7, 23.1, 9.0, 8.7, 11.0, 0.7]
planet_diameter = [4879, 12104, 12756, 6792, 142984, 120536, 51118, 49528, 2370]

# Create trace, sizing bubbles by planet diameter
fig = go.Figure(go.Scatter3d(
    x = distance_from_sun,
    y = density,
    z = gravity,
    text = planets,
    mode = 'markers',
    marker = dict(
        sizemode = 'diameter',
        sizeref = 750, # info on sizeref: https://plotly.com/python/reference/scatter/#scatter-marker-sizeref
        size = planet_diameter,
        color = temperatures,
        colorbar_title = 'Mean<br>Temperature',
        colorscale=[[0, 'rgb(5, 10, 172)'], [.3, 'rgb(255, 255, 255)'], [1, 'rgb(178, 10, 28)']]
        )
))

fig.update_layout(
    width=800,
    height=800,
    title=dict(text="Planets!"),
    scene=dict(
        xaxis=dict(
            title=dict(
                text="Distance from Sun",
                font=dict(
                    color="white"
                )
            )
        ),
        yaxis=dict(
            title=dict(
                text="Density",
                font=dict(
                    color="white"
                )
            )
        ),
        zaxis=dict(
            title=dict(
                text="Gravity",
                font=dict(
                    color="white"
                )
            )
        ),
        bgcolor="rgb(20, 24, 54)"
    )
)

fig.show()
```





Reference

See https://plotly.com/python/reference/scatter3d/ (https://plotly.com/python/reference/scatter3d/) and https://plotly.com/python/reference/scatter/#scatter-marker-sizeref (https://plotly.com/python/reference/scatter/#scatter-marker-sizeref)
for more information and chart attribute options!

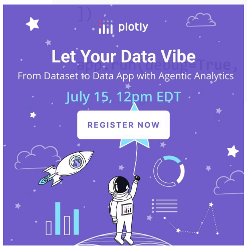
What About Dash?

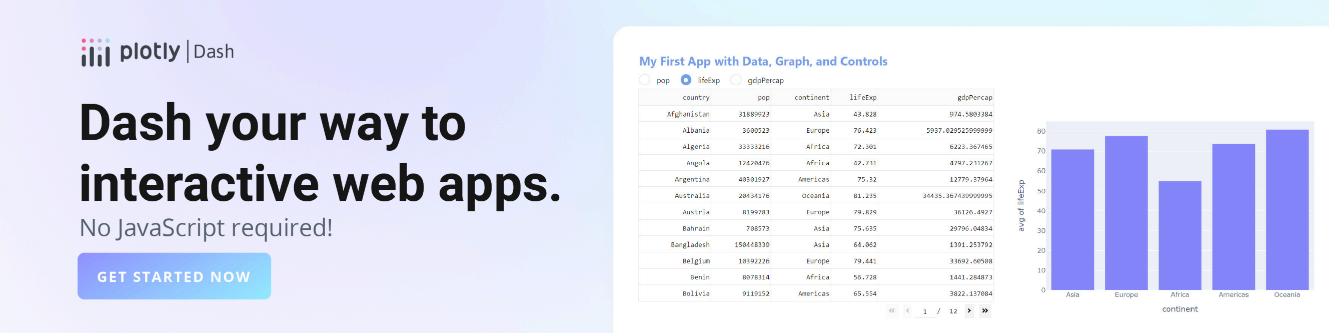
Dash (https://dash.plot.ly/) is an open-source framework for building analytical applications, with no Javascript required, and it is tightly integrated with the Plotly graphing library.

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
```





My First App with Data, Graph, and Controls

○ pop ● lifeExp ○ gdpPercap

country	pop	continent	lifeExp	gdpPercap
Afghanistan	31889923	Asia	43.828	974.5803384
Albania	3600523	Europe	76.423	5937.029525999999
Algeria	33333216	Africa	72.301	6223.367465
Angola	12420476	Africa	42.731	4797.231267
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	1391.253792
Belgium	10392226	Europe	79.441	33692.60508
Benin	8078314	Africa	56.728	1441.284873
Bolivia	9119152	Americas	65.554	3822.137084

(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)

Products

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

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

Pricing

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

About Us

Careers (https://plotly.com/careers)

Resources (https://plotly.com/resources/)

Blog (https://medium.com/@plotlygraphs)

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/)