

plotly

August 14, 2024

1 Plotly

Plotly is a Data Visualization library by the company Plotly based out of Canadaa with support in languages such as Python, JS, Julia, etc.

1.0.1 Advantages over other Data Viz libraries

- Multi language support
- Lot's of graphs
- Interactive plots
- Beautiful plots

1.0.2 Disadvantages

- Does not work with live Data streams, However Dash can be implemented for that.

1.0.3 Main Components of Plotly

1. Plotly GO
2. Plotly Express
3. Dash

1.1 1. Plotly Graph Objects(GO)

```
[122]: import plotly.graph_objects as go
import plotly.express as px
import numpy as np
import pandas as pd
```

```
[123]: # Datasets
tips = px.data.tips()
iris = px.data.iris()
gap = px.data.gapminder()
```

```
[124]: gap.head()
```

```
[124]:      country continent  year  lifeExp      pop  gdpPercap iso_alpha \
0  Afghanistan      Asia  1952   28.801  8425333  779.445314      AFG
1  Afghanistan      Asia  1957   30.332  9240934  820.853030      AFG
```

2	Afghanistan	Asia	1962	31.997	10267083	853.100710	AFG
3	Afghanistan	Asia	1967	34.020	11537966	836.197138	AFG
4	Afghanistan	Asia	1972	36.088	13079460	739.981106	AFG

	iso_num
0	4
1	4
2	4
3	4
4	4

1.1.1 Scatter Plot using Plotly GO

```
[125]: temp = gap[gap['year'] == 2007]
temp
```

```
[125]:
```

	country	continent	year	lifeExp	pop	gdpPercap	\
11	Afghanistan	Asia	2007	43.828	31889923	974.580338	
23	Albania	Europe	2007	76.423	3600523	5937.029526	
35	Algeria	Africa	2007	72.301	33333216	6223.367465	
47	Angola	Africa	2007	42.731	12420476	4797.231267	
59	Argentina	Americas	2007	75.320	40301927	12779.379640	
...	
1655	Vietnam	Asia	2007	74.249	85262356	2441.576404	
1667	West Bank and Gaza	Asia	2007	73.422	4018332	3025.349798	
1679	Yemen, Rep.	Asia	2007	62.698	22211743	2280.769906	
1691	Zambia	Africa	2007	42.384	11746035	1271.211593	
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298	

	iso_alpha	iso_num
11	AFG	4
23	ALB	8
35	DZA	12
47	AGO	24
59	ARG	32
...
1655	VNM	704
1667	PSE	275
1679	YEM	887
1691	ZMB	894
1703	ZWE	716

[142 rows x 8 columns]

```
[126]: trace1 = go.Scatter(x=temp['lifeExp'], y=temp['gdpPercap'], mode='markers')
trace2 = go.Scatter(x=[0, 1, 2], y=[0, 90, 30000], mode='lines')
data = [trace1, trace2]
```

```

layout = go.Layout(title='Life Exp Vs GDP per Capita for 2007', xaxis={'title':
    ↪'Life Exp'}, yaxis={'title': 'GDP'})
fig = go.Figure(data, layout)
fig.show()

```

1.2 2. Plotly Express(PX)

```

[127]: px.scatter(temp, x='lifeExp', y='gdpPercap', color='continent', size='pop',
    ↪size_max=70, hover_name='country')

```

1.2.1 Animations

```

[128]: px.scatter(gap, x='lifeExp', y='gdpPercap',
    color='continent', size='pop',
    size_max=70, hover_name='country',
    range_x=[30, 95],
    animation_frame='year', animation_group='country')

```

1.2.2 Line plot

Population of India over the years

```

[129]: temp = gap[gap['country'] == 'India']
temp

```

```

[129]:
country continent  year  lifeExp      pop  gdpPercap iso_alpha \
696   India      Asia  1952   37.373  372000000  546.565749      IND
697   India      Asia  1957   40.249  409000000  590.061996      IND
698   India      Asia  1962   43.605  454000000  658.347151      IND
699   India      Asia  1967   47.193  506000000  700.770611      IND
700   India      Asia  1972   50.651  567000000  724.032527      IND
701   India      Asia  1977   54.208  634000000  813.337323      IND
702   India      Asia  1982   56.596  708000000  855.723538      IND
703   India      Asia  1987   58.553  788000000  976.512676      IND
704   India      Asia  1992   60.223  872000000  1164.406809      IND
705   India      Asia  1997   61.765  959000000  1458.817442      IND
706   India      Asia  2002   62.879  1034172547  1746.769454      IND
707   India      Asia  2007   64.698  1110396331  2452.210407      IND

iso_num
696      356
697      356
698      356
699      356
700      356
701      356

```

```

702      356
703      356
704      356
705      356
706      356
707      356

```

```
[130]: px.line(temp, x='year', y='pop', title='India Population over years')
```

Plotting population of India, China, Pak over the years

```
[131]: temp = gap[gap['country'].isin(['India', 'China', 'Pakistan'])].
        ↪pivot(index='year', columns='country', values='lifeExp')
temp
```

```
[131]: country      China      India      Pakistan
year
1952      44.00000  37.373      43.436
1957      50.54896  40.249      45.557
1962      44.50136  43.605      47.670
1967      58.38112  47.193      49.800
1972      63.11888  50.651      51.929
1977      63.96736  54.208      54.043
1982      65.52500  56.596      56.158
1987      67.27400  58.553      58.245
1992      68.69000  60.223      60.838
1997      70.42600  61.765      61.818
2002      72.02800  62.879      63.610
2007      72.96100  64.698      65.483
```

```
[132]: px.line(temp, x=temp.index, y=temp.columns)
```

1.2.3 Bar Chart

India's population over the years

```
[133]: temp = gap[gap['country'] == 'India']
        px.bar(temp, x='year', y='pop', title="India's population over the years")
```

1.2.4 Grouped Bar Chart

Population of three countries

```
[134]: temp = gap[gap['country'].isin(['India', 'China', 'Pakistan'])].
        ↪pivot(index='year', columns='country', values='pop')
temp
```

```
[134]: country      China      India      Pakistan
year
1952      556263527      372000000      41346560
1957      637408000      409000000      46679944
1962      665770000      454000000      53100671
1967      754550000      506000000      60641899
1972      862030000      567000000      69325921
1977      943455000      634000000      78152686
1982     1000281000      708000000      91462088
1987     1084035000      788000000     105186881
1992     1164970000      872000000     120065004
1997     1230075000      959000000     135564834
2002     1280400000     1034172547     153403524
2007     1318683096     1110396331     169270617
```

```
[135]: px.bar(temp, x=temp.index, y=temp.columns, barmode='group', log_y=True,
↳text_auto=True)
```

1.2.5 Stacked Bar Chart

pop contribution per country to a continents pop

```
[136]: temp = gap[gap['year'] == 2007]
px.bar(temp, x='continent', y='pop', color='country', log_y=True)
```

1.2.6 Animation

```
[137]: px.bar(gap, x='continent', y='pop', color='country', animation_frame='year',
↳animation_group='country', range_y=[0, 4000000000])
```

1.2.7 Histogram

Histogram of life Exp of all countries in 2007

```
[138]: temp = gap[gap['year'] == 2007]
px.histogram(temp, x='lifeExp', nbins=10, text_auto=True)
```

Histogram of Sepal length of all iris species

```
[139]: px.histogram(iris, x='sepal_length', color='species')
```

1.2.8 Pie Chart

pie chart of pop of european countries in 2007

```
[140]: temp = gap[(gap['continent'] == 'Europe') & (gap['year'] == 2007)]
px.pie(temp, values='pop', names='country', hover_name='country')
```

Pie Chart of World pop in 1952 continent wise

```
[141]: temp_ser = gap[gap['year'] == 1952].groupby('continent')['pop'].sum()
px.pie(temp_ser, values=temp_ser.values, names=temp_ser.index)
```

1.2.9 Sunburst Plot

Sunburst plots visualize hierarchical data spanning outwards radially from root to leaves

```
[142]: temp = gap[gap['year'] == 2007]
px.sunburst(temp, path=['continent', 'country'], values='pop', color='lifeExp')
```

```
[143]: px.sunburst(tips, path=['sex', 'smoker', 'time', 'day'], values='total_bill',
    ↪color='size')
```

1.2.10 Treemap

```
[144]: temp = gap[gap['year'] == 2007]
px.treemap(temp, path=[px.Constant('World'), 'continent', 'country'],
    ↪values='pop', color='lifeExp')
```

1.2.11 Heatmap

Heatmap of all continents with year on avg life exp

```
[145]: temp = gap.pivot_table(index='year', columns='continent', values='lifeExp',
    ↪aggfunc='mean')
px.imshow(temp)
```

1.2.12 3D Scatter Plot

3D scatter plot of all country data for 2007

```
[146]: temp = gap[gap['year'] == 2007]
px.scatter_3d(temp, x='lifeExp', y='pop', z='gdpPercap', log_y=True,
    ↪color='continent', hover_name='country')
```

```
[147]: px.scatter_3d(iris, x='sepal_length', y='sepal_width', z='petal_length',
    ↪color='species')
```

1.2.13 Scatter Matrix

```
[148]: px.scatter_matrix(iris, dimensions=['sepal_length', 'sepal_width',
    ↪'petal_length', 'petal_width'], color='species')
```

```
[149]: px.scatter(tips, x='total_bill', y='tip', facet_col='smoker', facet_row='sex',
    ↪color='time')
```

1.2.14 3D Surface Plot

3D Surface plots can not be created using Plotly express, Plotly graph objects can be used

```
[150]: x = np.linspace(-10, 10, 100)
y = np.linspace(-10, 10, 100)

xx, yy = np.meshgrid(x,y)

z = xx**2 + yy**2

trace = go.Surface(x=x, y=y, z=z)

data = [trace]
layout = go.Layout(title='3D Surface Plot')

fig = go.Figure(data, layout)

fig.show()
```

1.2.15 Countour Plots

```
[151]: x = np.linspace(-10, 10, 100)
y = np.linspace(-10, 10, 100)

xx, yy = np.meshgrid(x,y)

z = xx**2 + yy**2

trace = go.Contour(x=x, y=y, z=z)

data = [trace]
layout = go.Layout(title='3D Surface Plot')

fig = go.Figure(data, layout)

fig.show()
```

1.2.16 Facet Plot

```
[152]: px.histogram(tips, x='total_bill', facet_row='sex')
```

```
[153]: px.scatter(gap, x='lifeExp', y='gdpPercap', facet_col='year', facet_col_wrap=2)
```

1.2.17 Subplots

can't use plotly express functionalities.

```
[154]: from plotly.subplots import make_subplots
```

```
[155]: fig = make_subplots(rows=2, cols=2)
```

```
[156]: fig.add_trace(  
    go.Scatter(x=[1, 9, 5], y=[2, 10, 1]),  
    row = 1,  
    col = 1  
)  
fig.add_trace(  
    go.Histogram(x=[1,9,5,22,109,134,56,78,12,34,89]),  
    row = 1,  
    col = 2  
)  
fig.add_trace(  
    go.Scatter(x=[1, 9, 5], y=[2, 10, 1]),  
    row = 2,  
    col = 1  
)  
fig.add_trace(  
    go.Histogram(x=[1,9,5,22,109,134,56,78,12,34,89]),  
    row = 2,  
    col = 2  
)  
  
fig.update_layout(title='Subplots')  
fig.show()
```