2023-07-03_Interactive_03_PlotLy

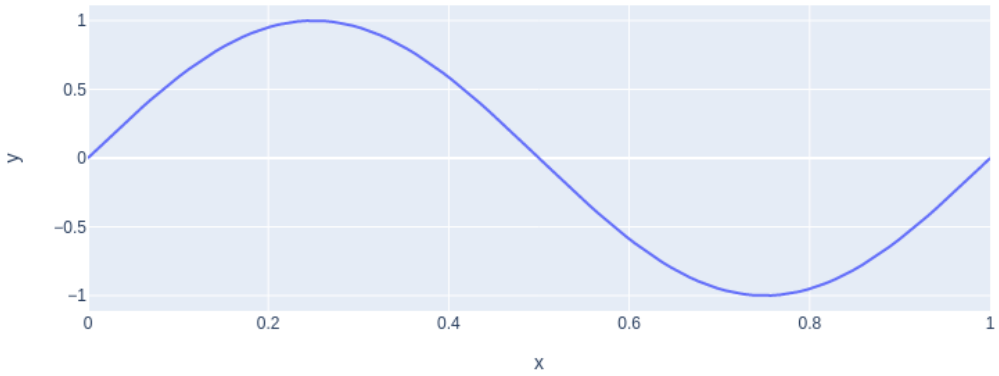
July 3, 2023

[1]:
```
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
```

1 PlotLy: Interactive

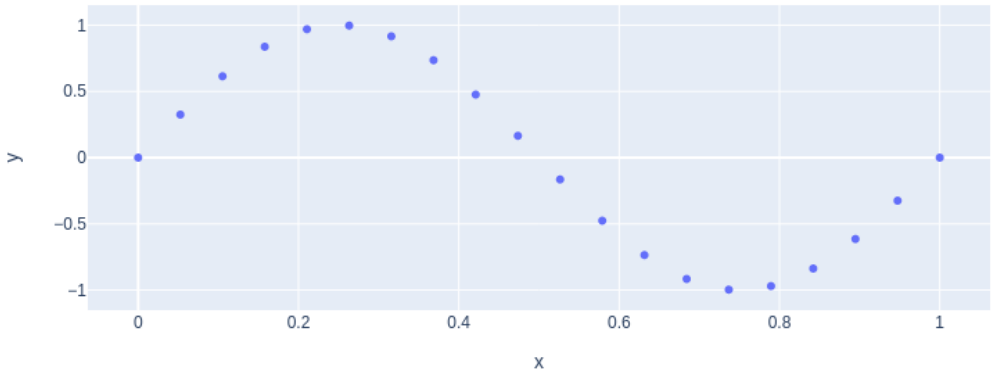
1.1 A basic line plot with PlotLy Express

[2]:
```
x=np.linspace(0,1,num=100)
y1=np.sin(2*np.pi*x)
y2=np.cos(2*np.pi*x)

fig = px.line(x=x, y=y1,width=600,height=300)
fig.show()
```



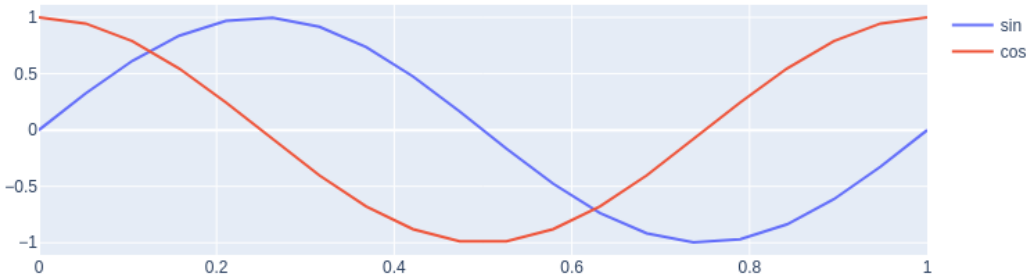
1.1.1 Add custom hover data

[3]:
```
x=np.linspace(0,1,num=20)
y1=np.sin(2*np.pi*x)
y2=["{:.03f}".format(z) for z in np.cos(2*np.pi*x)]

fig = px.scatter(x=x, y=y1,\
        hover_data={"cos":y2},width=600,height=300)
fig.show()
```



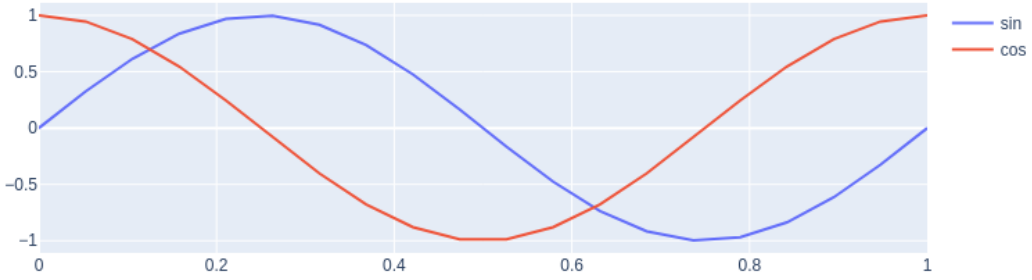
1.2 Build a simple plot from plotly.graph_objects

[4]:
```
x=np.linspace(0,1,num=20)
y1=np.sin(2*np.pi*x)
y2=np.cos(2*np.pi*x)

fig=go.Figure(layout_width=500,layout_height=300)
tr1=go.Scatter(x=x, y=y1,name="sin")
tr2=go.Scatter(x=x, y=y2,name="cos")
fig.add_trace(tr1)
fig.add_trace(tr2)
fig.show()
```



1.2.1 Add custom hover data

[5]:
```
x=np.linspace(0,1,num=20)
y1=np.sin(2*np.pi*x)
y2=np.cos(2*np.pi*x)
y3=["tan={:.03f}".format(z) for z in np.tan(2*np.pi*x)]

fig=go.Figure(layout_width=500,layout_height=300)
tr1=go.Scatter(x=x, y=y1,name="sin")
tr2=go.Scatter(x=x, y=y2,name="cos",hovertext=y3)
fig.add_trace(tr1)
fig.add_trace(tr2)
fig.show()
```



1.2.2 Exporting to HTML

Since PlotLy is at its heart a JavaScript library, export of interactive figures is a key functionality.

[6]:
```
# export a simple stand-alone html (assuming that plotly js lib is present in␣
 ↪export directory)
fig.write_html("../Web/PlotLy/example_py_1.html",include_plotlyjs="directory")
```

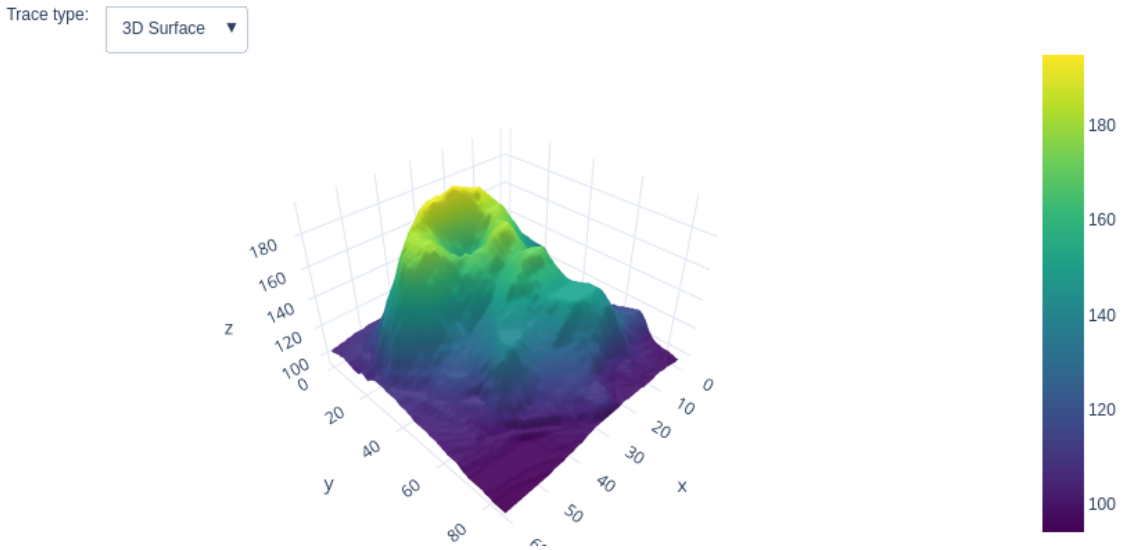
[7]:
```
# if the figure is part of a larger document that is created in a python script
# then can also write to a larger filestream
import codecs
outStream=codecs.open("../Web/PlotLy/example_py_2.html","w","utf-8")
outStream.write('<!doctype html>\n'+
    '<html>\n'+
    '<head>\n'+
    '    <meta charset="utf-8">\n'+
    '    <title>Another PlotLy Example</title>\n'+
    '    <script src="plotly-2.12.1.min.js"></script>\n'+
    '</head>\n'+
    '<body style="background:#e0e0e0">\n')

outStream.write('<p>Hello world.</p>\n')


fig.write_html(outStream,include_plotlyjs=False,full_html=False)

outStream.write('\n'+
    '</body>\n'+
    '</html>\n'
    )

outStream.close()
```

1.3 A fancy example

[8]:
```
import plotly.graph_objects as go

import pandas as pd

# load dataset
df = pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/
 ↪volcano.csv")
```

```
# create figure
fig = go.Figure()

# Add surface trace
fig.add_trace(go.Surface(z=df.values.tolist(), colorscale="Viridis"))

# Update plot sizing
fig.update_layout(
    width=800,
    height=400,
    autosize=False,
    margin=dict(t=0, b=0, l=0, r=0),
    template="plotly_white",
)

# Update 3D scene options
fig.update_scenes(
    aspectratio=dict(x=1, y=1, z=0.7),
    aspectmode="manual"
)

# Add dropdown
fig.update_layout(
    updatemenus=[
        dict(
            buttons=list([
                dict(
                    args=["type", "surface"],
                    label="3D Surface",
                    method="restyle"
                ),
                dict(
                    args=["type", "heatmap"],
                    label="Heatmap",
                    method="restyle"
                )
            ]),
            direction="down",
            pad={"r": 10, "t": 10},
            showactive=True,
            x=0.1,
            xanchor="left",
            y=1.1,
            yanchor="top"
        ),
    ]
)
```

```
# Add annotation
fig.update_layout(
    annotations=[
        dict(text="Trace type:", showarrow=False,
        x=0, y=1.085, yref="paper", align="left")
    ]
)

fig.show()
```



[ ]: