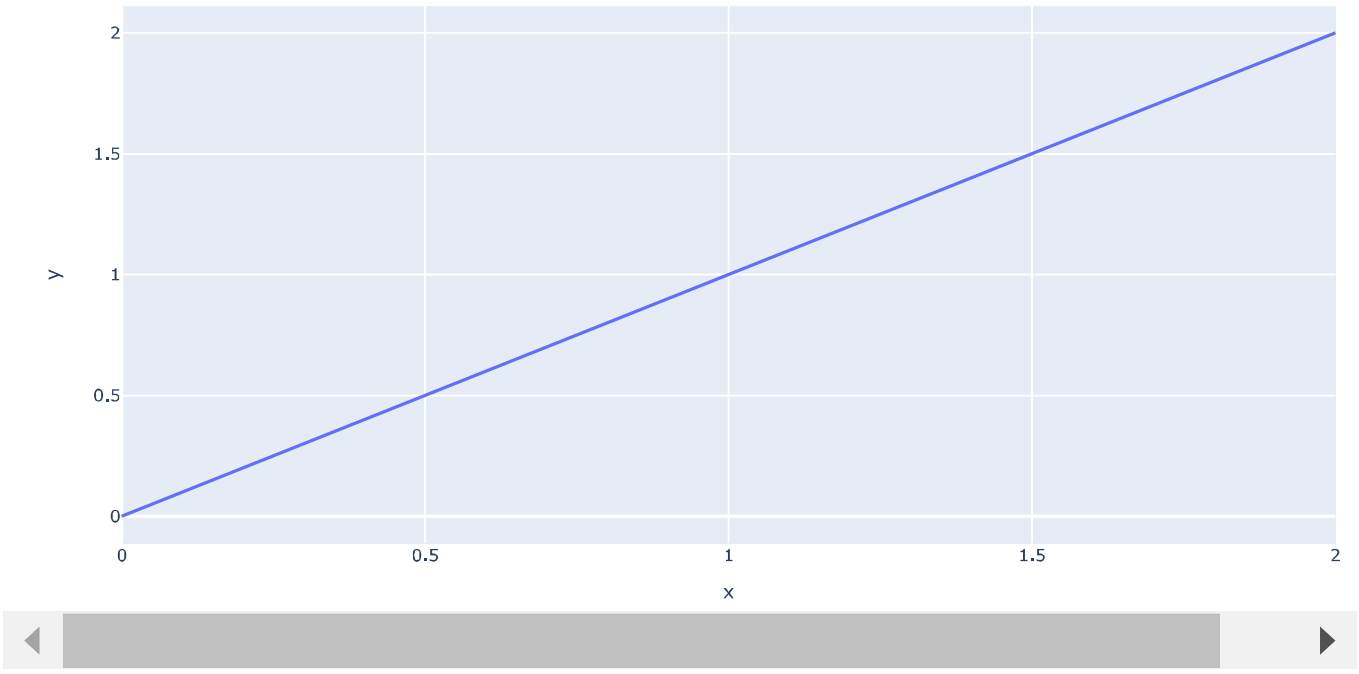
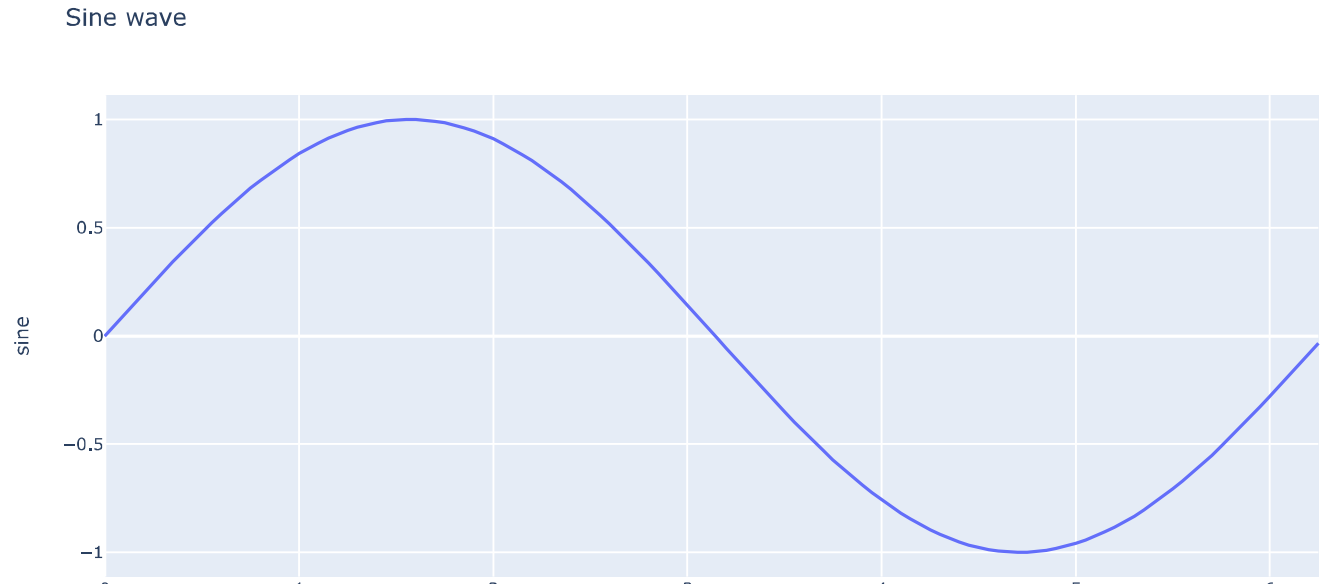
```
#SINE AND COS WAVE
import plotly.express as px

fig = px.line(x=[0,1,2], y=[0,1,2])
fig.show
```

```
<bound method BaseFigure.show of Figure({
    'data': [{'hovertemplate': 'x=%{x}<br>y=%{y}<extra></extra>',
              'legendgroup': '',
              'line': {'color': '#636efa', 'dash': 'solid'},
              'marker': {'symbol': 'circle'},
              'mode': 'lines',
              'name': '',
              'orientation': 'v',
              'showlegend': False,
              'type': 'scatter',
              'x': array([0, 1, 2]),
              'xaxis': 'x',
              'y': array([0, 1, 2]),
              'yaxis': 'y'}],
    'layout': {'legend': {'tracegroupgap': 0},
               'margin': {'t': 60},
               'template': '...',
               'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'x'}},
               'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'y'}}}
})>
```

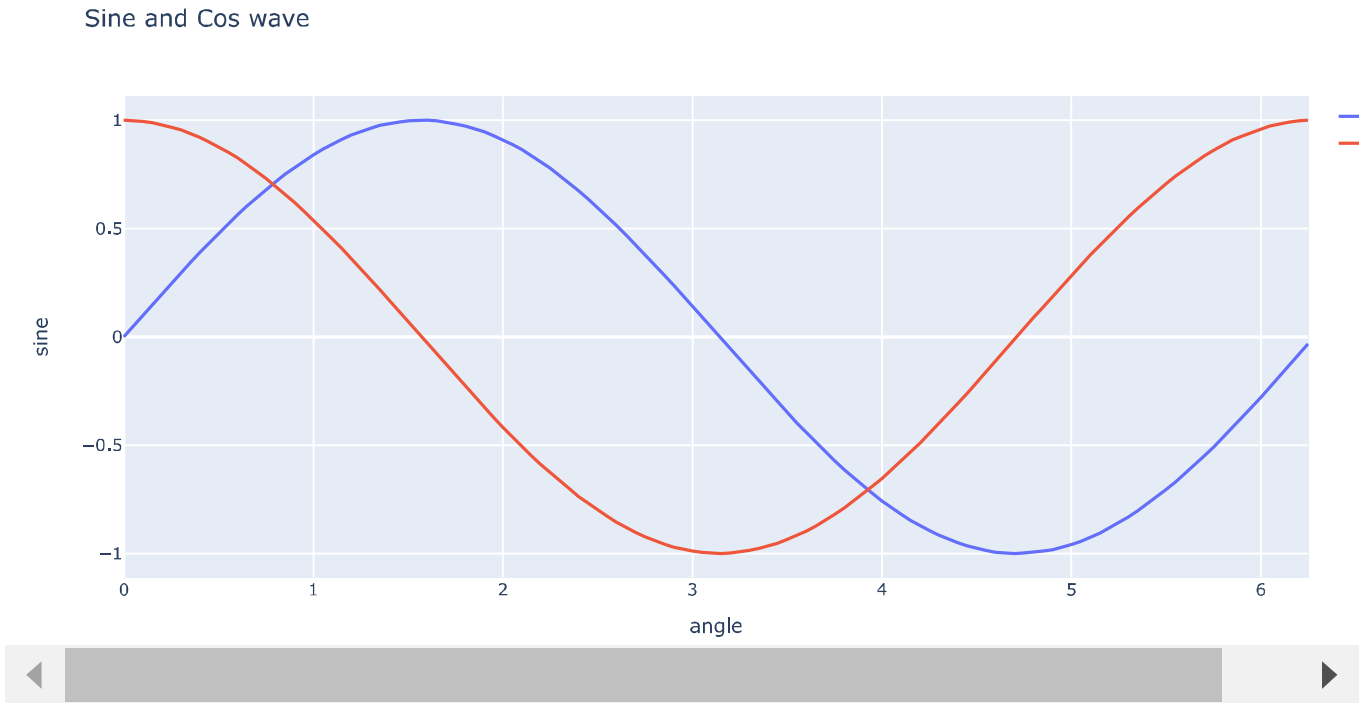
```
fig.show()
```



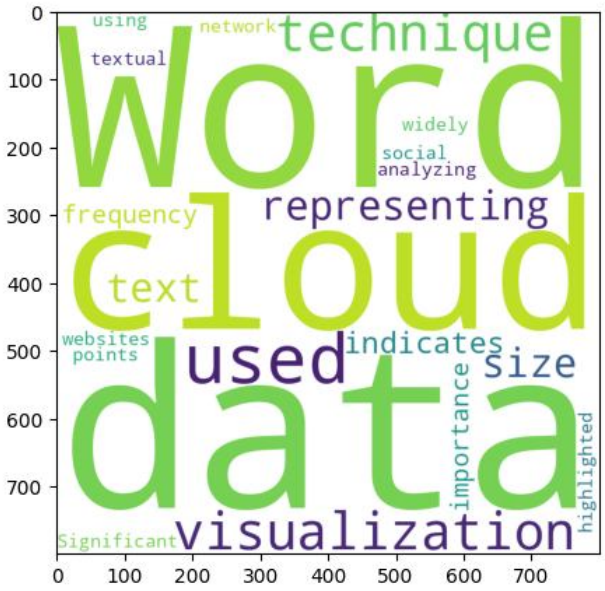

```
import plotly.graph_objs as go
import numpy as np
import math

xpoints=np.arange(0, math.pi*2, 0.05)
ypoints=np.sin(xpoints)

trace0 = go.Scatter(
    x = xpoints, y = ypoints
)
data = [trace0]
layout = go.Layout(title = "Sine wave", xaxis = {'title':'angle'},
                   yaxis = {'title':'sine'})
fig = go.Figure(data, layout)
fig.show()
```



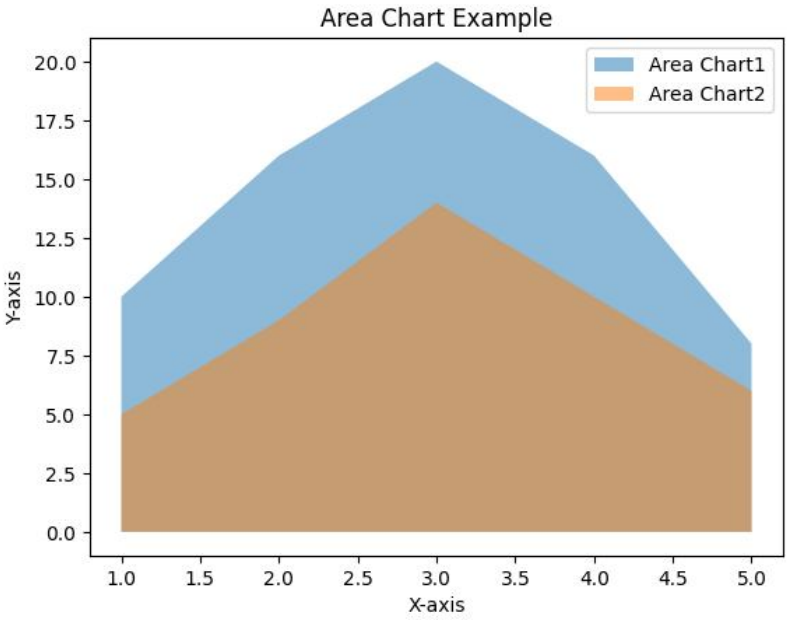

```
import plotly.graph_objs as go
import numpy as np
import math

xpoints=np.arange(0, math.pi*2, 0.05)
ypoints=np.sin(xpoints)
ypoints1=np.cos(xpoints)

trace0 = go.Scatter(
    x = xpoints, y = ypoints, name='Sine'
)
trace1 = go.Scatter(
    x = xpoints, y = ypoints1, name='Cos'
)

data = [trace0, trace1]
layout = go.Layout(title = "Sine and Cos wave", xaxis = {'title':'angle'},
                   yaxis = {'title':'sine'})
fig = go.Figure(data, layout)
fig.show()
```



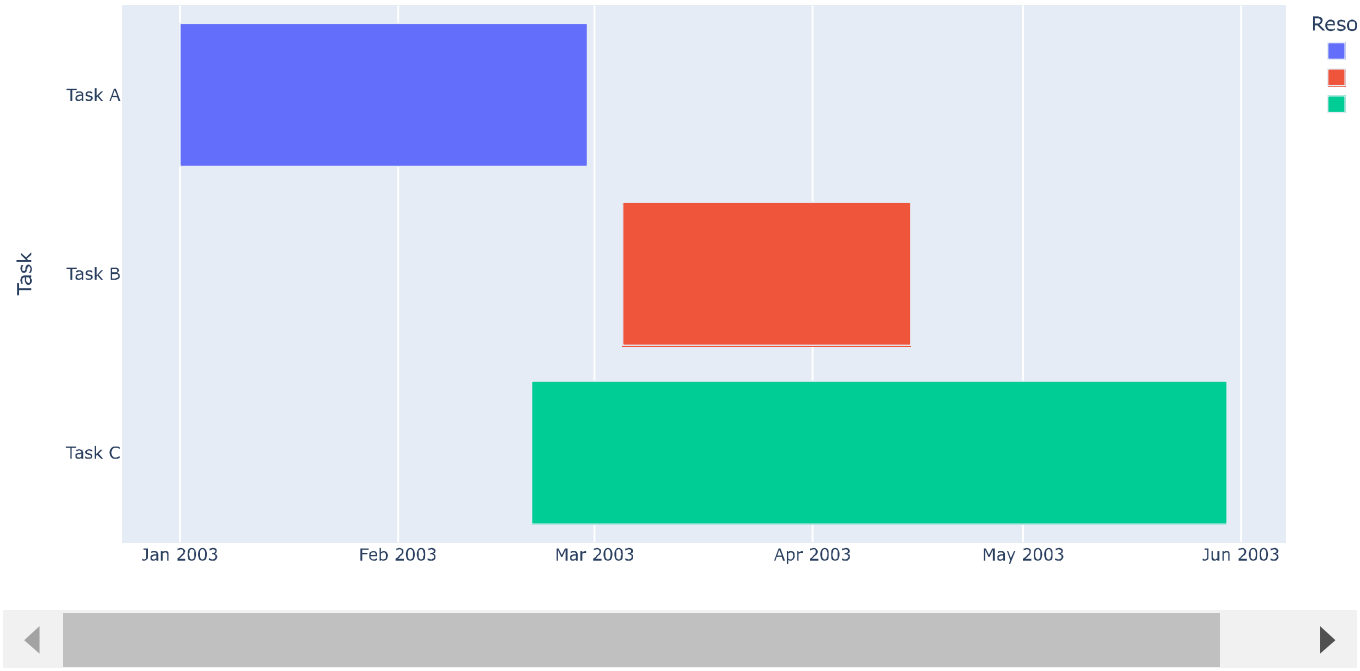
```
#WORD CLOUD
import matplotlib.pyplot as plt
from wordcloud import WordCloud
input_text = '''Word Cloud is a data visualization technique
                used for representing text data in which the
                size of each word indicates its frequency or
                importance. Significant textual data points
                can be highlighted using a word cloud.
                Word clouds are widely used for analyzing
                data from social network websites.'''
wc = WordCloud(width=800, height=800, background_color='white').generate(input_text)
plt.figure(figsize=(5, 10))
plt.imshow(wc)
plt.axis("on")
plt.show()
```



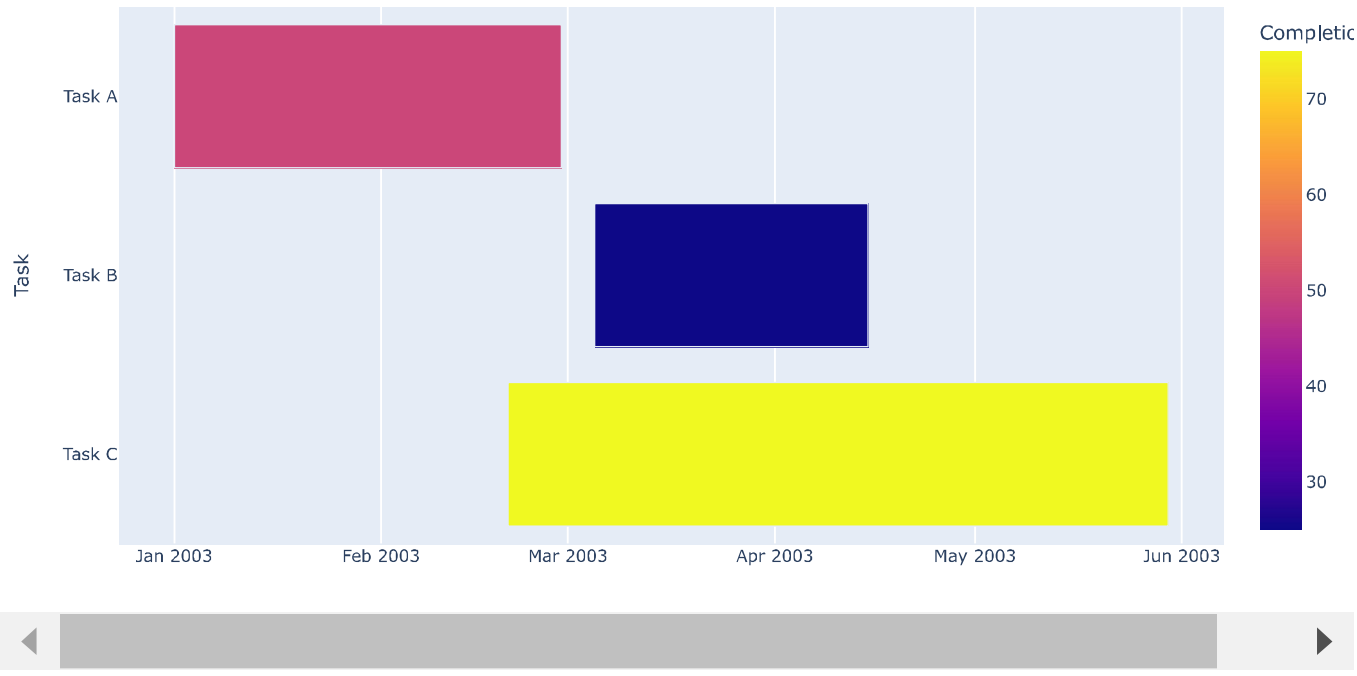

```
#AREA CHART
import matplotlib.pyplot as plt
x1 = [1, 2, 3, 4, 5]
x2 = [10, 16, 20, 16, 8]
x3 = [5, 9, 14, 10, 6]
plt.fill_between(x1, x2, alpha=0.5, label='Area Chart1')
plt.fill_between(x1, x3, alpha=0.5, label='Area Chart2')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Area Chart Example')
plt.legend()
plt.show()
```



```
#GanttChart
import plotly.express as px
import pandas as pd
df = pd.DataFrame([
    dict(Task="Task A", Start = "2003-01-01", Finish="2003-02-28", Resource="Vinod"),
    dict(Task="Task B", Start = "2003-03-05", Finish="2003-04-15", Resource="Ravish"),
    dict(Task="Task C", Start = "2003-02-20", Finish="2003-05-30", Resource="Purush"),
])
fig=px.timeline(df, x_start="Start", x_end="Finish", y="Task", color="Resource")
fig.update_yaxes(autorange="reversed")
fig.show()
```



```
import plotly.express as px
import pandas as pd
df = pd.DataFrame([
    dict(Task="Task A", Start = "2003-01-01", Finish="2003-02-28", Completion_pct=50),
    dict(Task="Task B", Start = "2003-03-05", Finish="2003-04-15", Completion_pct=25),
    dict(Task="Task C", Start = "2003-02-20", Finish="2003-05-30", Completion_pct=75),
])
fig=px.timeline(df, x_start="Start", x_end="Finish", y="Task",color="Completion_pct")
fig.update_yaxes(autorange="reversed")
fig.show()
```




```
#TREE MAP
import pandas as pd
import plotly.express as px
df=pd.read_excel("/content/Superstore.xlsx")
df
```

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code
1	CA-2016-152156	2016-08-11 00:00:00	2016-11-11 00:00:00	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420
2	CA-2016-152156	2016-08-11 00:00:00	2016-11-11 00:00:00	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420
3	CA-2016-138688	2016-12-06 00:00:00	6/16/2016	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036
4	US-2015-108966	2015-11-10 00:00:00	10/18/2015	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311
...	...	...	...	...	...	...	...	...	...	...	...
9990	CA-2014-110422	1/21/2014	1/23/2014	Second Class	TB-21400	Tom Boeckenhauer	Consumer	United States	Miami	Florida	33180
9991	CA-2017-121258	2/26/2017	2017-03-03 00:00:00	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California	92627
9992	CA-2017-121258	2/26/2017	2017-03-03 00:00:00	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California	92627
9993	CA-2017-121258	2/26/2017	2017-03-03 00:00:00	Standard Class	DB-13060	Dave Brooks	Consumer	United States	Costa Mesa	California	92627
9994	CA-2017-119914	2017-04-05 00:00:00	2017-09-05 00:00:00	Second Class	CC-12220	Chris Cortes	Consumer	United States	Westminster	California	92683

9995 rows × 1 columns

```
import plotly.express as px
fig=px.treemap(df,path=['Category','Sub-Category'],values='Sales')
fig.show()
```