plotly

Reference

https://plot.ly/python/

Import Packages

```
import plotly.express as px
import plotly.graph_objects as go
```

Layout

Setting Graph Size

```
fig.update_layout(autosize=False,
    width=500,height=500)
```

Titles

```
fig.update_layout(
  title={'text':Test Chart},
  xaxis_title-'Test Title',
  yaxis_title='Test Title',
  font=dict( family='Helvetica',
   size=14, color="#333333"))
```

Legend

```
fig.update_layout(showlegend=True)
```

Subplots

```
fig = make_subplots(rows=1, cols=2)
```

Custom Controls

RangeSlider

```
fig.update_layout(xaxis_rangeslider
    _visible=True)
```

Dropdown Menu

```
fig.update_layout(updatemenus=
dict(buttons= dict(args=["type","Num"],
label="Types",method="restyle")),
direction="down", pad={"r":10, "t":10},
showactive=True, x=0.1,
xanchor="left",
y=1.1, yanchor="top")
```

Basic Charts

Bar Chart

```
fig = go.Figure(data=[
    month = ["Jan","Feb","Mar"]
    go.Bar(x=month, y=[1, 3, 2]),
    go.Bar(x=month, y=[2, 6, 4]),
    name="Bar Chart"])

fig.update_traces(
    marker_color='blue',
    marker_line_color='red',
    marker_line_width=1,
    opacity=0.4)
fig.update_layout(barmode='group')
```

Line Chart

```
fig = go.Figure(data=[
  go.line(x=[1, 2, 3], y=[1, 3, 2],
  line_color='purple')])
```

Scatter Plot

```
Using the Plotly Express Library
df = px.data.tips()
fig = px.scatter(df, x="total_bill",
y="tip")

Breaking Data into Categories
fig = px.scatter(df, x="total_bill",
y="tip", facet_col="sex",
facet_row="time", color="sex")
```

Statistical Charts

Box Plot

```
fig = px.box(df, x='day', y="Bills",
color="smoker", notched=True)
```

Histogram

```
fig = px.histogram(df,
x='sepal_length', y="petal_length",
color="species"
)
```

Maps

Bubble Maps

```
px.add_trace(go.Scattergeo(
        lon = [50,50]
        lat = [20,10])
```

Choropleth Maps

```
px.choropleth(
    locations=["CA","TX","NY"],
    locationmode="USA-states",
    color="Bergeron",
    colorscale="Red"
    scope="usa")
```

Scatter Mapbox

```
px.scatter_mapbox(
    lat=[20,50],
    lon=[100,400],
    color="Red",
    mode="markers", size=8)
```

Financial Charts

Time Series

```
px.add_trace(go.Scatter(
    x=Date,
    y=df["AAPL.High"],
    name="AAPL High",
    line_color='deepskyblue'))
```

Funnel Charts

```
px.funnel(dict(
    number=[20,30,40,50]
    stage=["visit","Down",
    "Invoice"]))
```

Waterfall Charts

```
add_trace(go.Waterfall(
    x = [["2018", "2019", "2020"],
    measure = ["sales", "consulting",
    "profit", "tax"],
    base = 1000))
```

3D Charts

3d Scatter Plot

```
fig = go.Figure(data=[
go.Scatter3d(
x=x, y=y, z=z, mode='markers',
marker=dict(
    size=12,
    color=z,
    colorscale='Viridis',
    showscale= True,
    colorbar = dict(thickness=15,
    len=0.5,x=0.8, y=0.6),
    opacity=0.8
    )
```

Surface Plot with Contour

```
fig = go.Figure(go.Surface(
contours = {
    "x": {"show": True, "start": 1.5,
    "end": 2, "size": 0.04,
    "color":"white"},
    "z": {"show": True, "start": 0.5,
    "end": 0.8, "size": 0.05}
    },
    x = x, y = y, z = z]))
```

3d Camera Controls

```
fig = go.Figure(data=go.Surface(z=z,
showscale=False))
camera = dict(
    up=dict(x=0, y=0, z=1),
    center=dict(x=0, y=0, z=0),
    eye=dict(x=1.25, y=1.25, z=1.25))
fig.update_layout(scene_camera=camera)
```

Volume Plot

```
fig = go.Figure(data=go.Volume(
    x=X.flatten(),
    y=Y.flatten(),
    z=Z.flatten(),
    value=values,
    isomin=-0.1,
    isomax=0.8,
    opacity=0.1, # needs to be small to
  # see through all surfaces
    surface_count=21, # needs to be a
  # large number for good volume render
    ))
```