

```
import plotly.express as px
import plotly.graph_objects as go
import pandas as pd
import numpy as np
import networkx as nx
```

Question 1

```
data = pd.DataFrame({
    'Company': ['Apple', 'Microsoft', 'Amazon', 'Google', 'Facebook'],
    'Revenue_2022': [394, 198, 513, 280, 117],
    'Revenue_2023': [420, 215, 540, 310, 130]
})
fig1 = px.bar(data, x='Company', y=['Revenue_2022', 'Revenue_2023'],
              barmode='group', title="Company Revenue (2022 vs 2023)")
fig1.show()
```

Question 2

```
data = pd.DataFrame({
    'Advertising Budget': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100],
    'Sales Revenue': [15, 25, 40, 50, 65, 80, 85, 100, 120, 140]
})
fig2 = px.scatter(data, x='Advertising Budget', y='Sales Revenue',
                  color='Sales Revenue', size='Sales Revenue',
                  title="Advertising Budget vs Sales Revenue")
fig2.show()
```

Question 3

```
data = pd.DataFrame({
    'Brand': ['Apple', 'Samsung', 'Xiaomi', 'Oppo', 'Vivo'],
    'Market Share': [30, 28, 17, 12, 13]
})
fig3 = px.pie(data, names='Brand', values='Market Share',
              title="Smartphone Market Share", hover_data=['Market
Share'])
fig3.show()
```

Question 4

```
data = pd.DataFrame({
    'Job Sector': ['IT', 'Finance', 'Healthcare', 'Education',
'Retail'] * 5,
    'Salary': [75000, 85000, 62000, 48000, 40000, 77000, 90000, 65000,
50000, 42000,
              78000, 87000, 67000, 52000, 43000, 80000, 89000, 69000,
54000, 45000,
```

```

82000, 91000, 71000, 56000, 47000]
})
fig4 = px.box(data, x='Job Sector', y='Salary', title="Salary
Distribution by Job Sector")
fig4.show()

```

Question 5

```

data = pd.DataFrame({
    'GDP': [19352, 47585, 36867, 30334, 8644, 3846, 43442, 30454,
35695],
    'Inflation': [0.59, 4.86, 4.24, 1.45, 1.31, 1.86, 2.86, 2.44,
1.81],
    'Unemployment': [10.3, 4.67, 6.50, 7.39, 8.47, 5.39, 9.17, 10.1,
3.55],
    'Interest Rate': [6.46, 2.53, 1.58, 9.53, 9.69, 3.74, 1.87, 7.15,
4.96]
})
fig5 = px.imshow(data.corr(), text_auto=True, title="Correlation
Heatmap")
fig5.show()

```

Question 6

```

data = pd.DataFrame({
    'Month': ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'],
    'Product A Sales': [500, 600, 700, 800, 750, 780],
    'Product B Sales': [400, 450, 470, 490, 520, 550],
    'Profit': [50, 80, 100, 120, 110, 130]
})
fig6 = go.Figure()
fig6.add_trace(go.Bar(x=data['Month'], y=data['Product A Sales'],
name="Product A Sales"))
fig6.add_trace(go.Scatter(x=data['Month'], y=data['Profit'],
mode='lines+markers', name="Profit"))
fig6.update_layout(title="Sales & Profit Analysis")
fig6.show()

```

Question 7

```

data = pd.DataFrame({
    'Country': ['USA', 'China', 'India', 'Germany', 'Brazil'] * 3,
    'Year': [2000]*5 + [2010]*5 + [2020]*5,
    'GDP': [10, 5, 2, 3, 1, 15, 9, 5, 4, 2, 22, 14, 7, 5, 3],
    'Population': [280, 1260, 1000, 83, 175, 310, 1350, 1200, 82, 190,
331, 1440, 1380, 80, 210],
    'Life Expectancy': [77, 71, 65, 80, 68, 79, 74, 69, 82, 72, 81,
76, 72, 83, 75]
})

```

```

})
fig7 = px.scatter(data, x="Population", y="Life Expectancy",
animation_frame="Year",
                    size="GDP", color="Country", title="Economic Growth
Animation")
fig7.show()

```

Question 8

```

edges = pd.DataFrame({
    'User A': ['Alice', 'Alice', 'Bob', 'Charlie', 'David', 'Eve',
'Frank', 'Grace', 'Hannah', 'Ivan'],
    'User B': ['Bob', 'Charlie', 'David', 'Eve', 'Frank', 'Grace',
'Hannah', 'Ivan', 'Alice', 'Bob']
})
G = nx.from_pandas_edgelist(edges, 'User A', 'User B')
pos = nx.spring_layout(G)
fig8 = go.Figure()
for edge in G.edges():
    fig8.add_trace(go.Scatter(x=[pos[edge[0]][0], pos[edge[1]][0]],
                             y=[pos[edge[0]][1], pos[edge[1]][1]],
                             mode='lines'))
fig8.show()

```

Question 9

```

data = {
    "Date": [
        "2024-01-01", "2024-01-02", "2024-01-03", "2024-01-04", "2024-
01-05",
        "2024-01-06", "2024-01-07", "2024-01-08", "2024-01-09", "2024-
01-10",
        "2024-01-11", "2024-01-12", "2024-01-13", "2024-01-14", "2024-
01-15",
        "2024-01-16", "2024-01-17", "2024-01-18", "2024-01-19", "2024-
01-20"
    ],
    "Stock Price": [
        102.48, 100.31, 105.25, 110.64, 102.86,
        103.87, 113.95, 110.90, 105.73, 111.80,
        107.78, 108.78, 113.33, 103.56, 105.51,
        112.34, 111.09, 118.74, 113.64, 112.13
    ]
}
stock_data = pd.DataFrame(data)
stock_data["Date"] = pd.to_datetime(stock_data["Date"])
fig9 = px.scatter(stock_data, x="Date", y="Stock Price",
trendline="ols",
                    title="Stock Price Prediction with Regression")

```

```
fig9.show()
```

Question 10

```
iris = px.data.iris()
fig10 = px.scatter(iris, x="sepal_width", y="sepal_length",
color="species",
                    title="Iris Dataset - Sepal Width vs Sepal Length")
fig10.show()
```