Q1: Scatter plot for Titanic dataset

Code:

```
import plotly.express as px
import seaborn as sns

# Load Titanic dataset
titanic = sns.load_dataset("titanic")

# Scatter plot: age vs fare
fig = px.scatter(titanic, x="age", y="fare",
                 color="pclass", # optional: color by class
                 hover_data=["sex", "embarked"], # show extra info on
hover
                 title="Scatter Plot of Age vs Fare in Titanic
Dataset")
fig.show()
```

✅ This shows the **relationship between age and fare**, optionally colored by passenger class.

Q2: Box plot using tips dataset

Code:

```
# Load Tips dataset
tips = px.data.tips()

# Box plot: total bill by day
fig = px.box(tips, x="day", y="total_bill", color="sex",
             title="Box Plot of Total Bill by Day and Sex")
fig.show()
```

✅ This visualizes the **distribution of total bills** by day and sex.

Q3: Histogram with sex, total_bill, smoker, day

Code:

```
# Histogram with pattern_shape and color
fig = px.histogram(tips, x="sex", y="total_bill",
                   color="day", # color by day
                   pattern_shape="smoker", # different shapes for
smoker status
                   barmode="group", # separate bars for clarity
                   title="Histogram of Total Bill by Sex, Day, and
Smoker")
fig.show()
```

✅ This shows **total bill distributions grouped by sex**, with patterns for smokers and colors for days.

Q4: Scatter matrix (splom) using Iris dataset

Code:

```
# Load Iris dataset
iris = px.data.iris()

# Scatter matrix plot
fig = px.scatter_matrix(iris,
                        dimensions=["sepal_length", "sepal_width",
"petal_length", "petal_width"],
                        color="species",
                        title="Scatter Matrix of Iris Dataset")
fig.show()
```

✅ This shows **pairwise relationships** between sepal and petal measurements, colored by species.

Q5: Distplot using Plotly Express

- **Distplot** is a **distribution plot** used to visualize the distribution of a single variable, usually with a histogram and optionally a kernel density estimate (KDE).

- In **Plotly Express**, `px.histogram` can be used to create distplot-like plots, optionally with `nbins` or `marginal="box"` or `"violin"`.

Code:

```
import numpy as np

# Generate random data
data = np.random.normal(loc=0, scale=1, size=500)

# Distplot using px.histogram
fig = px.histogram(data, x=data, nbins=30, marginal="box",
                   title="Distribution Plot (Distplot) of Random
Data")
fig.show()
```

✅ This creates a **histogram with a boxplot on top**, similar to a traditional distplot.