

basic problems based on simple 2D plots using Matplotlib:

1. **Problem: Plotting a Sinusoidal Curve**

Explanation: Plot a sinusoidal curve $y = \sin(x)$ over the range $[0, 2\pi]$.

Code:

```
```python
import matplotlib.pyplot as plt
import numpy as np

Generate data
x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)

Plotting
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.title('Sinusoidal Curve')
plt.grid(True)
plt.show()
```
```

Answer: The code will display a plot of the sinusoidal curve over the range from 0 to 2π .

2. ****Problem: Plotting a Parabola****

****Explanation:**** Plot the parabolic curve $(y = x^2)$ over the range $[-5, 5]$.

****Code:****

```
```python
import matplotlib.pyplot as plt
import numpy as np

Generate data
x = np.linspace(-5, 5, 100)
y = x**2

Plotting
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('x^2')
plt.title('Parabolic Curve')
plt.grid(True)
plt.show()
```
```

****Answer:**** This code will display a plot of the parabolic curve over the range from -5 to 5.

3. ****Problem: Plotting Multiple Curves****

****Explanation:**** Plot the functions $y = x$ and $y = x^2$ on the same graph.

****Code:****

```
` `` `python

import matplotlib.pyplot as plt
import numpy as np

# Generate data
x = np.linspace(-5, 5, 100)
y1 = x
y2 = x**2

# Plotting
plt.plot(x, y1, label='y=x')
plt.plot(x, y2, label='$y=x^2$')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Multiple Curves')
plt.legend()
plt.grid(True)
plt.show()
` `` `
```

****Answer:**** This code will plot both $y = x$ and $y = x^2$ on the same graph.

4. ****Problem: Scatter Plot with Random Data****

****Explanation:**** Generate a scatter plot of 50 random points.

****Code:****

```
` `` python

import matplotlib.pyplot as plt
import numpy as np

# Generate random data
np.random.seed(0)
x = np.random.rand(50)
y = np.random.rand(50)

# Plotting
plt.scatter(x, y)
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Scatter Plot of Random Data')
plt.grid(True)
plt.show()
` ``
```

****Answer:**** This code will display a scatter plot of 50 randomly generated points.

5. ****Problem: Bar Chart of Categorical Data****

****Explanation:**** Create a bar chart showing the sales of different products.

****Code:****

```
` `` python
```

```
import matplotlib.pyplot as plt
```

```
# Sample data
```

```
products = ['Product A', 'Product B', 'Product C', 'Product D']
```

```
sales = [5000, 7000, 4500, 6000]
```

```
# Plotting
```

```
plt.bar(products, sales)
```

```
plt.xlabel('Products')
```

```
plt.ylabel('Sales')
```

```
plt.title('Product Sales')
```

```
plt.xticks(rotation=45)
```

```
plt.grid(axis='y')
```

```
plt.show()
```

```
` ``
```

****Answer:**** This code will generate a bar chart showing the sales of different products. Each bar represents the sales figure for a particular product.

6. ****Problem: Plotting Exponential Decay****

****Explanation:**** Plot the exponential decay function $(y = e^{-x})$ over the range $[[0, 5]]$.

****Code:****

```
```python
import matplotlib.pyplot as plt
import numpy as np

Generate data
x = np.linspace(0, 5, 100)
y = np.exp(-x)

Plotting
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('e^(-x)')
plt.title('Exponential Decay')
plt.grid(True)
plt.show()
```
```

****Answer:**** This code will display a plot of the exponential decay function over the range from 0 to 5.

7. ****Problem: Plotting a Step Function****

****Explanation:**** Plot the step function $(y = \text{sign}(x))$ over the range $[-5, 5]$.

****Code:****

```
```python
import matplotlib.pyplot as plt
import numpy as np

Generate data
x = np.linspace(-5, 5, 100)
y = np.sign(x)

Plotting
plt.step(x, y, where='mid')
plt.xlabel('x')
plt.ylabel('sign(x)')
plt.title('Step Function')
plt.grid(True)
plt.show()
```
```

****Answer:**** This code will display a plot of the step function over the range from -5 to 5.

8. ****Problem: Plotting a Horizontal Line****

****Explanation:**** Plot a horizontal line $(y = 3)$ over the range $[-5, 5]$.

****Code:****

```
` `` ` python

import matplotlib.pyplot as plt
import numpy as np

# Generate data
x = np.linspace(-5, 5, 100)
y = np.full_like(x, 3)

# Plotting
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Horizontal Line')
plt.grid(True)
plt.show()
```



```

**\*\*Answer:\*\*** This code will display a plot of a horizontal line at  $(y = 3)$  over the range from -5 to 5.

#### 9. **\*\*Problem: Plotting a Vertical Line\*\***

**\*\*Explanation:\*\*** Plot a vertical line  $(x = 2)$  over the range  $[-5, 5]$ .

**\*\*Code:\*\***

```python

```
import matplotlib.pyplot as plt
```

```
# Plotting
```

```
plt.axvline(x=2, color='r', linestyle='--')
```

```
plt.xlabel('x')
```

```
plt.ylabel('y')
```

```
plt.title('Vertical Line')
```

```
plt.grid(True)
```

```
plt.show()
```

```

**\*\*Answer:\*\*** This code will display a plot of a vertical line at  $(x = 2)$  over the range from -5 to 5.

10. **\*\*Problem: Plotting a Parametric Curve\*\***

**\*\*Explanation:\*\*** Plot the parametric curve  $(x = \sin(t), y = \cos(t))$  for  $t$  in the range  $[0, 2\pi]$ .

**\*\*Code:\*\***

```
```python
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
# Generate data
```

```
t = np.linspace(0, 2*np.pi, 100)
```

```
x = np.sin(t)
```

```
y = np.cos(t)
```

```
# Plotting
```

```
plt.plot(x, y)
```

```
plt.xlabel('x')
```

```
plt.ylabel('y')
plt.title('Parametric Curve:  $x = \sin(t)$ ,  $y = \cos(t)$ ')
plt.grid(True)
plt.axis('equal')
plt.show()
...

```

****Answer:**** This code will display a plot of the parametric curve defined by $(x = \sin(t), y = \cos(t))$ over the range from 0 to 2π .