Q1. Histogram

- **Code**

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("./data sets/tips.csv")

plt.hist(df['day'])
plt.title("Most Smoking Day")
plt.show()
```

Q2. Bar-chart

- **Code**

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("./data sets/tips.csv")

plt.bar(df['total_bill'],df['tip'])
plt.xlabel("Total Bill")
plt.ylabel("Tip")
plt.show()
```

Q3. Scatter-plot

- **Code**

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("./data sets/tips.csv")

plt.scatter(df['total_bill'], df['tip'])
plt.xlabel("Total Bill")
plt.ylabel("Tip")
plt.show()
```

Q4. Line-chart

- **Code**

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

x = np.array(np.random.randint(100, size=(20)))
y = x*3
```

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

Q5. Pie-chart

- **Code**

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("./data sets/tips.csv")

gender = df['sex'].value_counts()

plt.pie(gender, labels=['male', 'female'])
plt.show()
```

Q6. Donut Chart

- **Code**

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("./data sets/tips.csv")
gender = df['sex'].value_counts()

plt.pie(gender, labels=['male', 'female'], autopct='%1.1f%%')
center_circle = plt.Circle((0,0), 0.70, fc = 'white')
fig = plt.gcf()
fig.gca().add_artist(center_circle)
plt.show()
```

Q7. Box-plot

- **Code**

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("./data sets/tips.csv")

plt.boxplot(df['tip'])
plt.show()
```

Q8. Bobble-plot

- **Code**

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("./data sets/tips.csv")
```

```
plt.scatter(df['total_bill'], df['tip'], s=df['tip']*10, alpha=0.5)
plt.xlabel("Total Bill")
plt.ylabel("Tip")
plt.show()
```

Q9. Violin-plot

- **Code**

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("./data sets/tips.csv")

gender = df['sex'].value_counts()

plt.violinplot(gender, positions=None, vert=True, widths=0.5,
showmeans=False,showextrema=True,showmedians=False,quantiles=None,points=100)
plt.show()
```

Q10. Heat-map

- **Code**

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

data = np.random.rand(10, 10)
plt.colorbar(plt.imshow(data, cmap='viridis'))
plt.title('Heatmap Example')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

Q11. Venn-Diagram

- **Code**

```
from matplotlib_venn import venn2
import matplotlib.pyplot as plt

set1 = {'A', 'B', 'C', 'D'}
set2 = {'B', 'C', 'D', 'E'}
venn2([set1, set2], ('Set1', 'Set2'))

plt.title('Venn Diagram')
plt.show()
```

Q12. Tree-Map chart

- **Code**

```
import squarify
import matplotlib.pyplot as plt

sizes = [50, 30, 15, 5]
labels = ['A', 'B', 'C', 'D']

squarify.plot(sizes=sizes, label=labels, alpha=0.7)
plt.title("Treemap Chart")
plt.axis('off')
plt.show()
```

Q13. Recurrence-plot

- **Code**

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

np.random.seed(42)
data = np.random.rand(100)

threshold = 0.1
N = len(data)
recurrence_matrix = np.zeros((N, N))

for i in range(N):
    for j in range(N):
        if abs(data[i] - data[j]) < threshold:
            recurrence_matrix[i, j] = 1

plt.figure(figsize=(6, 6))
plt.imshow(recurrence_matrix, cmap='binary', origin='lower')
plt.title('Recurrence Plot')
plt.xlabel('Time')
plt.ylabel('Time')
plt.show()
```

Q14. 3D Scatter Plot

- **Code**

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

np.random.seed(42)
x = np.random.rand(100)
y = np.random.rand(100)
z = np.random.rand(100)
```

```
fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x, y, z, c='blue', marker='o')

ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
ax.set_title('3D Scatter Plot')
plt.show()
```