Ex.No. 02

# Visualizing using Graphs

**Aim:**

To develop a Python program for analysing and visualising data using graphical representations. The objective is to use various types of plots to understand data distribution, relationships and patterns within a dataset effectively.

**Procedure:**

**Step 01:** Import the necessary libraries for data analysis and visualisation, such as Pandas, Matplotlib, Seaborn and tools to load datasets.

**Step 02:** Load a structured dataset suitable for visualisation and convert it into a tabular format using Pandas for easy handling.

**Step 03:** Explore the dataset by viewing its structure, checking for missing values, and generating basic statistical summaries.

Step 04: Create line plots to observe trends or changes in numerical values across records.

Step 05: Use scatter plots to study relationships between two numerical attributes, possibly colour-coded by category.

Step 06: Draw bar plots to display the frequency distribution of categorical data.

Step 07: Plot histograms to understand the distribution of a numerical feature across intervals.

Step 08: Generate box plots to compare the distribution and spread of values across different categories.

Step 09: Use advanced visualisations such as count plots, violin plots, KDE plots, and pair plots to analyse patterns, relationships and class-wise comparisons in the datasets.

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
mean radius                569 non-null float64
mean texture               569 non-null float64
mean perimeter             569 non-null float64
mean area                  569 non-null float64
mean Smoothness            569 non-null float64
mean Compactness           569 non-null float64
mean Concavity             569 non-null float64
mean Concave points        569 non-null float64
mean Symmetry              569 non-null float64
mean fractal dimension     569 non-null float64
(radius) error             569 non-null float64
texture error              569 non-null float64
perimeter error            569 non-null float64
area error                 569 non-null float64
Smoothness error           569 non-null float64
Compactness error          569 non-null float64
Concavity error            569 non-null float64
Symmetry error             569 non-null float64
fractal dimension error    569 non-null float64
worst radius               569 non-null float64
worst texture              569 non-null float64
worst perimeter            569 non-null float64
worst area                 569 non-null float64
worst Smoothness           569 non-null float64
worst Compactness          569 non-null float64
worst Concavity            569 non-null float64
```

Step 10: Display a heatmap to show the correlation between different features, highlighting the strength and direction of relationships.

Program 2a:

```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.datasets import load_breast_
                                    cancer

cancer = load_breast_cancer()

data = pd.DataFrame(cancer.data, columns =
                                    cancer.feature_names)
data['target'] = cancer.target

print(data.info())
print(data.head(1))
print(data.describe())
print(data.isnull().sum())

plt.figure (figsize =(10,6))
plt.plot (data.index, data ['mean radius'],
          label = 'Mean Radius']
plt.title ('Line Plot of Mean Radius')
plt.xlabel ('Index')
plt.ylabel ('Mean Radius')
```

```
worst concave points      569 non-null float64
worst symmetry            569 non-null float64
worst fractal dimension   569 non-null float64

target                    569 non-null int32
dtypes: float64(30), int32(1)
memory usage: 135.7 KB
None
```

| | mean radius | mean texture | mean Perimeter | mean area | mean Smoothness \ |
|---|---|---|---|---|---|
| 0 | 17.99 | 10.38 | 122.8 | 1001.0 | 0.1184 |

| | mean Compactness | mean Concavity | mean Concave points | mean Symmetry \ |
|---|---|---|---|---|
| 0 | 0.2776 | 0.3001 | 0.1471 | 0.2419 |

| | mean fractal dimension | ... | worst texture | worst Perimeter | worst area \ |
|---|---|---|---|---|---|
| 0 | 0.07871 | ... | 17.33 | 184.6 | 2019.0 |

| | worst Smoothness | worst Compactness | worst Concavity | worst Concave points \ |
|---|---|---|---|---|
| 0 | 0.1622 | 0.6656 | 0.7119 | 0.2654 |

| | worst symmetry | worst fractal dimension | target |
|---|---|---|---|
| 0 | 0.4601 | 0.11890 | 0 |

[1 rows x 31 columns]

```
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize = (10,6))
plt.scatter(data['mean radius'], data
        ['mean texture'], c = data['target'],
                    cmap = 'coolwarm', alpha=0.5)
plt.title('Scatter Plot of Mean Radius vs
                        Mean Texture')
plt.xlabel('Mean Radius')
plt.ylabel('Mean Texture')
plt.grid(True)
plt.show()

plt.figure(figsize=(10,6))
plt.bar(data['target'].value_counts().index,
        data['target'].value_counts().values)
plt.title('Bar Plot of Target Class
                        Distribution')
plt.xlabel('Target Class')
plt.ylabel('Count')
plt.xticks([0,1], ['Malignant', 'Benign'])
plt.grid(True)
plt.show()

plt.figure(figsize = (10, 6))
plt.hist(data['mean area'], bins = 30,
                        alpha =0.7)
```

| | mean radius | mean texture | mean perimeter | mean area \ |
|---|---|---|---|---|
| Count | 569.000000 | 569.000000 | 569.000000 | 569.000000 |
| mean | 14.127292 | 19.289649 | 91.969033 | 645.889104 |
| std | 3.524049 | 4.301036 | 24.298981 | 351.914129 |
| min | 6.981000 | 9.710000 | 43.790000 | 143.500000 |
| 25% | 11.700000 | 16.170000 | 75.170000 | 420.300000 |
| 50% | 13.370000 | 18.840000 | 86.240000 | 551.100000 |
| 75% | 15.780000 | 21.80000 | 104.100000 | 782.700000 |
| max | 28.110000 | 39.280000 | 188.500000 | 2501.000000 |

| | mean Smoothness | mean Compactness | mean Concavity | mean Concave Points \ |
|---|---|---|---|---|
| Count | 569.000000 | 569.000000 | 569.000000 | 569.000000 |
| mean | 0.096360 | 0.104341 | 0.088799 | 0.048919 |
| std | 0.014064 | 0.052813 | 0.079720 | 0.038803 |
| min | 0.052630 | 0.019380 | 0.000000 | 0.000000 |
| 25% | 0.086370 | 0.064920 | 0.029560 | 0.020310 |
| 50% | 0.095870 | 0.092630 | 0.061540 | 0.033500 |
| 75% | 0.105300 | 0.130400 | 0.130700 | 0.074000 |
| max | 0.163400 | 0.345400 | 0.426800 | 0.201200 |

| | mean Symmetry | mean fractal dimension | ... | Worst texture \ |
|---|---|---|---|---|
| Count | 569.000000 | 569.000000 | ... | 569.000000 |
| mean | 0.181162 | 0.062798 | ... | 25.677223 |
| std | 0.027414 | 0.007060 | ... | 6.146258 |
| min | 0.106000 | 0.049960 | ... | 12.020000 |
| 25% | 0.161900 | 0.057700 | ... | 21.080000 |
| 50% | 0.179200 | 0.061540 | ... | 25.410000 |

```
plt.title('Histogram of Mean Area')
plt.xlabel ('Mean Area')
plt.ylabel ('Frequency')
plt.grid (True)
plt.show()

plt.figure (figsize = (10,6))
plt.boxplot ([data[data['target'] == 0]
        ['mean radius'], data[data['target']
        == 1]['mean radius']], labels =
                ['Malignant', 'Benign'])
plt.title('Bar Plot of Mean Radius by
                Target Class')
plt.xlabel ('Target Class')
plt.ylabel ('Mean Radius')
plt.grid (True)
plt.show()
```

| | | | | |
|---|---|---|---|---|
| 75% | 0.195700 | 0.066120 | ... | 29.720000 |
| max | 0.304000 | 0.097440 | ... | 49.540000 |

| | Worst Perimeter | worst area | Worst smoothness | Worst Compactness |
|---|---|---|---|---|
| count | 569.000000 | 569.000000 | 569.000000 | 569.000000 |
| mean | 107.261213 | 880.583128 | 0.132369 | 0.254265 |
| std | 33.602542 | 569.356993 | 0.022832 | 0.157336 |
| min | 50.410000 | 185.200000 | 0.071170 | 0.027290 |
| 25% | 84.110000 | 515.300000 | 0.116600 | 0.147200 |
| 50% | 97.660000 | 686.500000 | 0.131300 | 0.211900 |
| 75% | 125.400000 | 1084.000000 | 0.146000 | 0.339100 |
| max | 251.200000 | 4254.000000 | 0.222600 | 1.058000 |

| | Worst Concavity | Worst Concave points | Worst Symmetry |
|---|---|---|---|
| count | 569.000000 | 569.000000 | 569.000000 |
| mean | 0.272188 | 0.114606 | 0.290076 |
| std | 0.208624 | 0.65732 | 0.061867 |
| min | 0.000000 | 0.000000 | 0.156500 |
| 25% | 0.114500 | 0.064930 | 0.250400 |
| 50% | 0.226700 | 0.099930 | 0.282200 |
| 75% | 0.382900 | 0.161400 | 0.317900 |
| max | 1.252000 | 0.291000 | 0.663800 |

| | Worst fractal dimension | target 5 |
|---|---|---|
| count | 569.000000 | 569.000000 |
| mean | 0.083946 | 0.627417 |
| std | 0.018061 | 0.483918 |
| min | 0.055040 | 0.000000 |
| 25% | 0.071460 | 0.000000 |

Program 2b:

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer

cancer = load_breast_cancer()

data = pd.DataFrame(cancer.data, columns = cancer.
                                    feature_names)
data['target'] = cancer.target

print(data.info())
print(data.head(1))
print(data.describe())
print(data.isnull().sum())

plt.figure(figsize=(6,4))
sns.countplot(x='target', data=data,
                        palette='coolwarm')
plt.title('Count Plot of Target Classes')
plt.xlabel('Target Class')
plt.ylabel('Count')
plt.xticks(ticks = [0,1], labels = ['Malignant',
                                    'Benign'])
plt.show()
```

| | | |
|---|---|---|
| 50% | 0.080040 | 1.000000 |
| 75% | 0.092080 | 1.000000 |
| max | 0.207500 | 1.000000 |

[8 rows x 31 columns]

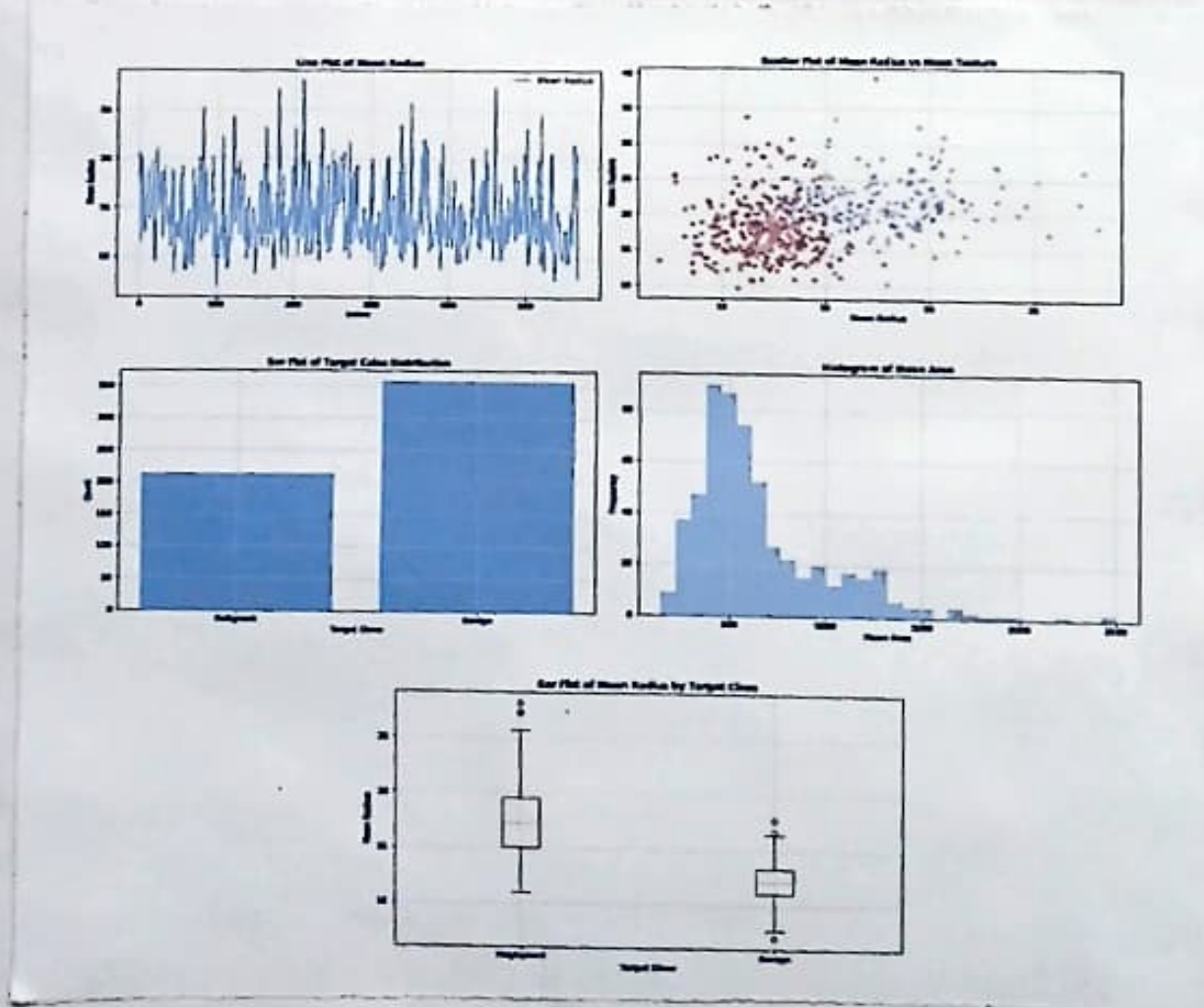
| | |
|---|---|
| mean radius | 0 |
| mean texture | 0 |
| mean perimeter | 0 |
| mean area | 0 |
| mean smoothness | 0 |
| mean compactness | 0 |
| mean concavity | 0 |
| mean concave points | 0 |
| mean symmetry | 0 |
| mean fractal dimension | 0 |
| radius error | 0 |
| texture error | 0 |
| perimeter error | 0 |
| area error | 0 |
| smoothness error | 0 |
| compactness error | 0 |
| concavity error | 0 |
| concave points error | 0 |
| symmetry error | 0 |
| fractal dimension error | 0 |
| worst radius | 0 |
| worst texture | 0 |
| worst perimeter | 0 |

```
plt.figure (figsize = (10, 6))
sns.kdeplot (data = data [data ['target'] == 0]
        ['mean radius'], shade = True,
                label = 'Malignant', color='r')
sns.kdeplot(data= data [data ['target'] == 1]
        ['mean radius'], shade = True,
                label = 'Benign', color='b')
plt.title ('KDE Plot of Mean Radius')
plt.xlabel ('Mean Radius')
plt.ylabel ('Density')
plt.legend ()
plt.show()

plt.figure (figsize = (10,6))
sns.violinplot (x ='target', y ='mean radius',
    data=data, palette = 'coolwarm')
plt.title ('Violin plot of Mean Radius by
                Target Class')
plt.xlabel ('Target Class')
plt.ylabel('Mean Radius')
plt.xticks (ticks = [0,1], labels= ['Malignant',
                'Benign'])
plt.show()

sns.pairplot(data, vars=['mean radius', 'mean
        texture', 'mean perimeter', 'mean area']
                ,hue='target', palette =
                'coolwarm')
plt.title('Pair Plot')
```

| | |
|---|---|
| worst area | 0 |
| worst smoothness | 0 |
| worst compactness | 0 |
| worst concavity | 0 |
| worst concave points | 0 |
| worst symmetry | 0 |
| worst fractal dimension | 0 |
| target | 0 |

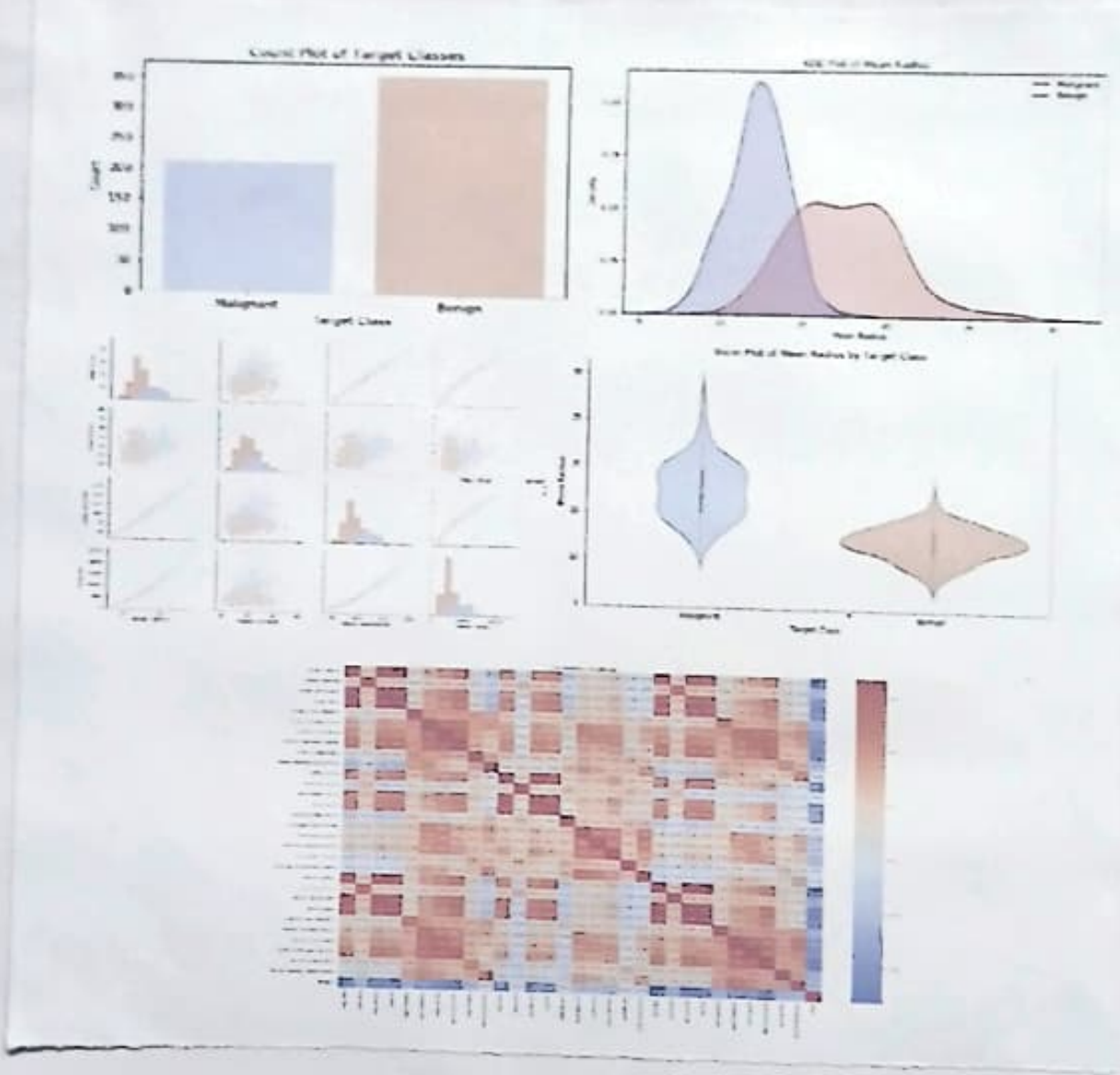
dtype: int 64

Output 2a:



```
plt.show()

plt.figure(figsize=(20,20))
sns.heatmap(data.corr(), annot=True,
            fmt='.2f', cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

Output 8b:



**Result:**

The program was successfully implemented to visualise data using graphs, helps to understand distributions, compare features, and identify patterns for better analysis.