ASSIGNMENT

ON

DATA VISUALIZATION TECHNIQUES

Submitted By,
Mathew peter
INTMCA S9
Roll NO: 42

Data visualization Techniques

Data visualization is a crucial aspect of machine learning that enables analysts to understand and make sense of data patterns, relationships, and trends. Through data visualization, insights and patterns in data can be easily interpreted and communicated to a wider audience, making it a critical component of machine learning

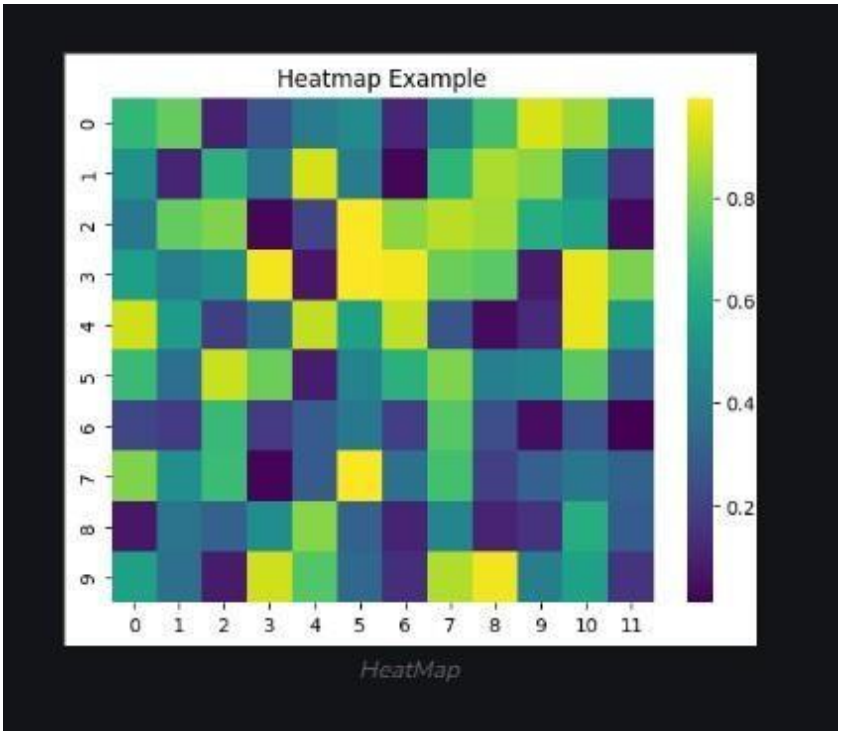
Heatmap

Heatmap use coding of color to represent the values of Matrix. Heatmap helps in finding correlations and patterns in large dataset.

Code

```
import seaborn as sns
import numpy as np
# Example: Heatmap
data = np.random.rand(10, 12)
sns.heatmap(data, cmap='viridis')
plt.title('Heatmap Example')
plt.show()
```

Output



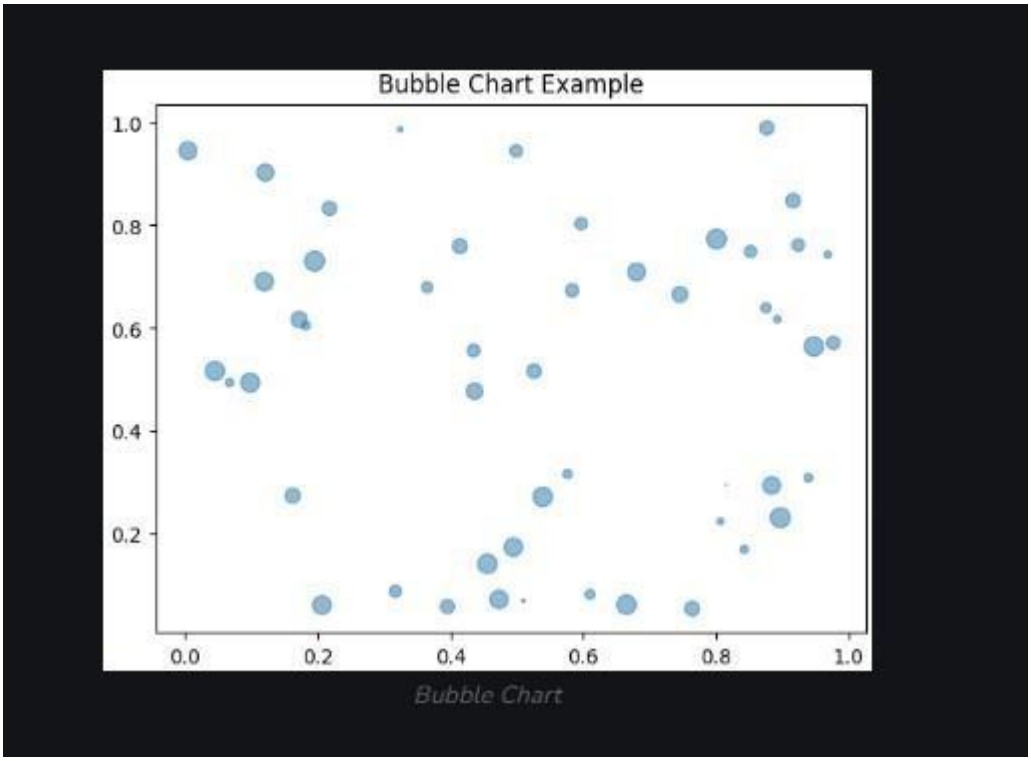
Bubble Chart

By using Bubble Chart_, you can add third dimension in scatter plot. The bubble chart represents the third variable with the size of the Bubble.

Code

```
import matplotlib.pyplot as plt
import numpy as np
# Example: Bubble Chart
x = np.random.rand(50)
y = np.random.rand(50)
sizes = np.random.rand(50) * 100
plt.scatter(x, y, s=sizes, alpha=0.5)
plt.title('Bubble Chart Example')
plt.show()
```

Output



Density Chart

A density chart (also known as a kernel density estimate, or KDE plot) is a way to estimate the probability density function of a continuous random variable. It represents data distribution and is a smoothed version of a histogram.

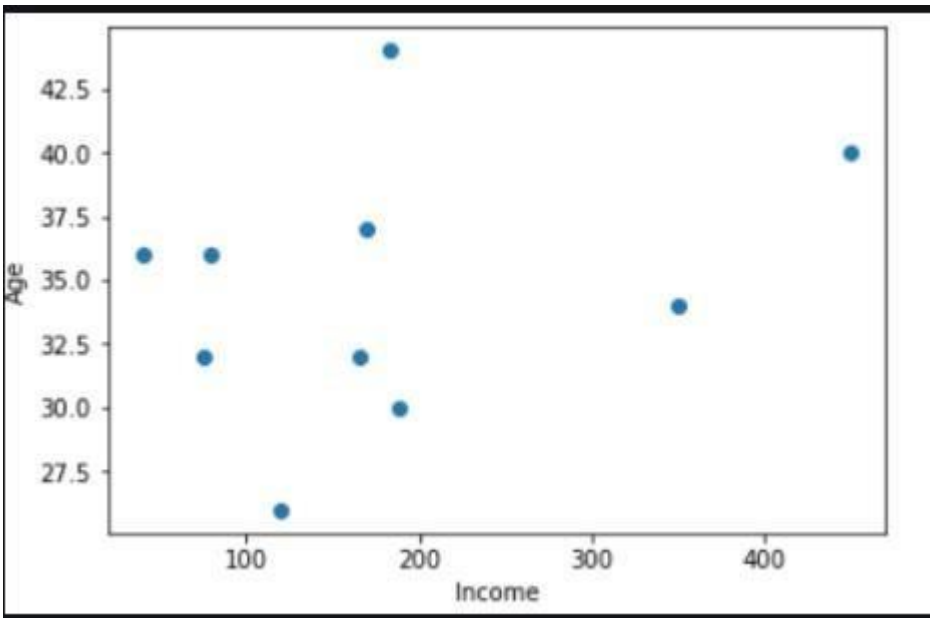
Scatter Chart

A scatter chart shows the relationship between two different variables and it can reveal the distribution trends. It should be used when there are many different data points, and you want to highlight similarities in the data set. This is useful when looking for outliers and for understanding the distribution of your data.

Code

```
# scatter plot between income and age
plt.scatter(df['income'], df['age'])
plt.show()
```

Output



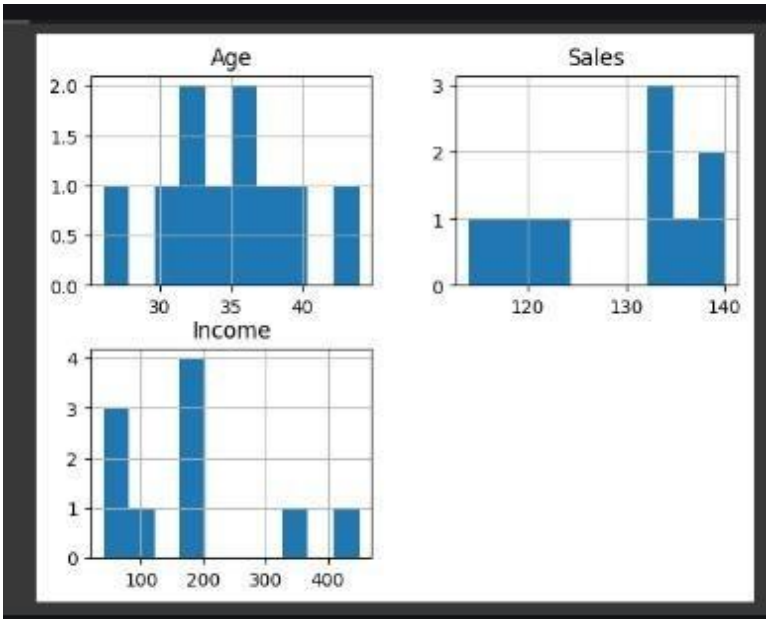
Histogram

The histogram represents the frequency of occurrence of specific phenomena which lie within a specific range of values and are arranged in consecutive and fixed intervals. In the below code histogram is plotted for Age, Income, Sales, So these plots in the output show frequency of each unique value for each attribute.

Code

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

# create 2D array of table given above
data = [['E001', 'M', 34, 123, 'Normal', 350],
        ['E002', 'F', 40, 114, 'Overweight', 450],
        ['E003', 'F', 37, 135, 'Obesity', 169],
        ['E004', 'M', 30, 139, 'Underweight', 189],
        ['E005', 'F', 44, 117, 'Underweight', 183],
        ['E006', 'M', 36, 121, 'Normal', 80],
        ['E007', 'M', 32, 133, 'Obesity', 166],
        ['E008', 'F', 26, 140, 'Normal', 120],
        ['E009', 'M', 32, 133, 'Normal', 75],
        ['E010', 'M', 36, 133, 'Underweight', 40] ]

# dataframe created with
# the above data array
df = pd.DataFrame(data, columns = ['EMPID', 'Gender',
                                   'Age', 'Sales',
                                   'BMI', 'Income'] )

# create histogram for numeric data
df.hist()

# show plot
plt.show()
```

Output



Quartile Plot

A quartile plot, often referred to as a box plot, is a way of summarizing a data set by displaying its quartiles (minimum, first quartile, median, third quartile, and maximum). It provides a visual summary of the data distribution and helps identify outliers.

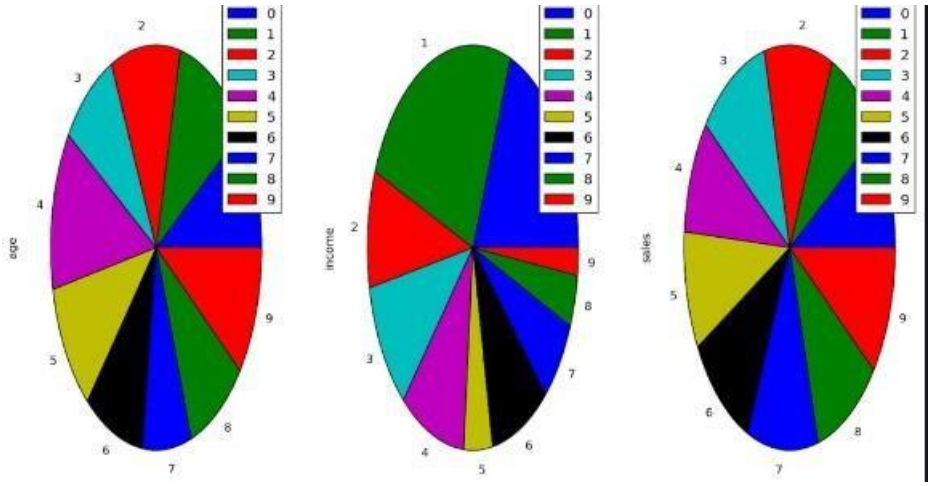
Pie Chart

A pie chart shows a static number and how categories represent part of a whole the composition of something. A pie chart represents numbers in percentages, and the total sum of all segments needs to equal 100%.

Code

```
plt.pie(df['Age'], labels = {"A", "B", "C",
                             "D", "E", "F",
                             "G", "H", "I", "J"},

autopct ='% 1.1f %%', shadow = True)
plt.show()

plt.pie(df['Income'], labels = {"A", "B", "C",
                                "D", "E", "F",
                                "G", "H", "I", "J"},

autopct ='% 1.1f %%', shadow = True)
plt.show()

plt.pie(df['Sales'], labels = {"A", "B", "C",
                               "D", "E", "F",
                               "G", "H", "I", "J"},
autopct ='% 1.1f %%', shadow = True)
plt.show()
```

Output



Bar Chart

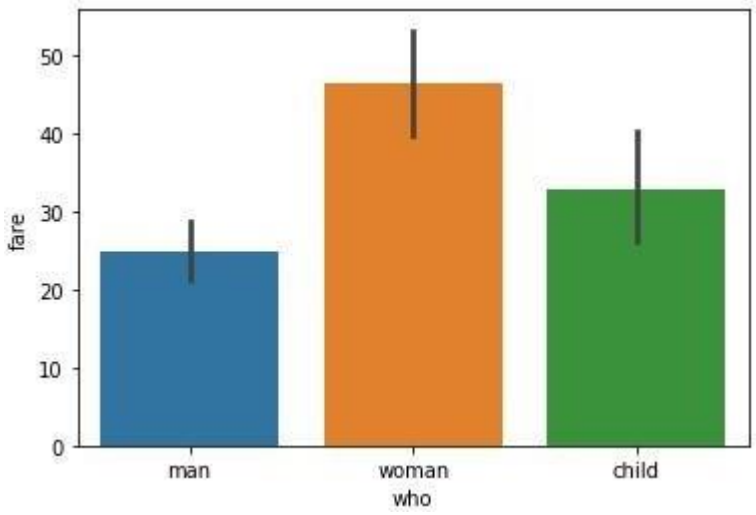
The bar chart is a frequency chart for a qualitative variable. A bar chart can be used to access the most-occurring and least-occurring categories within a dataset.

To draw a bar chart, call ‘barplot()’ of the seaborn library.

Code

```
# importing the required library
import seaborn as sns
import matplotlib.pyplot as plt

# read a titanic.csv file
# from seaborn library
df = sns.load_dataset('titanic')

# who v/s fare barplot
sns.barplot(x = 'who',
            y = 'fare',
            data = df)

# Show the plot
plt.show()
```

Output



Box plot chart

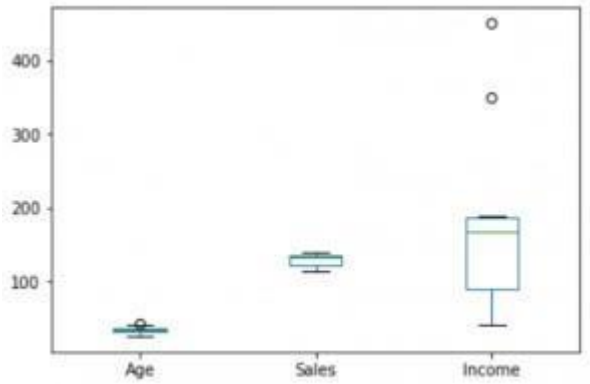
A box plot is a graphical representation of statistical data based on the minimum, first quartile, median, third quartile, and maximum. The term “box plot” comes from the fact that

the graph looks like a rectangle with lines extending from the top and bottom. Because of the extending lines, this type of graph is sometimes called a box-and-whisker plot.

Code

```
# For each numeric attribute of dataframe
df.plot.box()

# individual attribute box plot
plt.boxplot(df['Income'])
plt.show()
```

Output



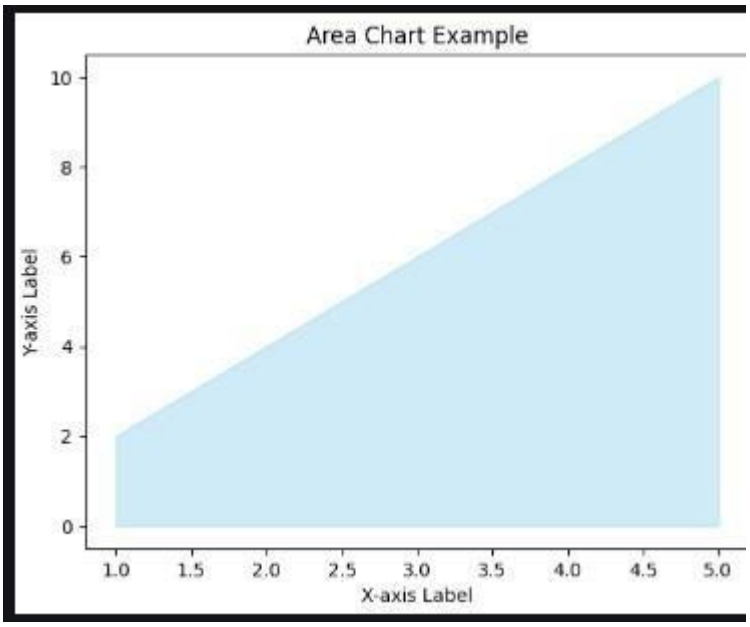
Area Chart

Area Chart are similar to line charts but there is area difference between the line and the x-axis is generally filled. They are helpful generally in showing magnitude over time.

Code

```
import matplotlib.pyplot as plt
# Example: Area Chart
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]
plt.fill_between(x, y, color='skyblue', alpha=0.4)
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.title('Area Chart Example')
plt.show()
```

Output


Area Chart Example
Y-axis Label
X-axis Label
0
2
4
6
8
10
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0