VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science Engineering and Information Systems (SCORE)

Machine Learning (SWE4012)

(L51 + L52)

Lab Digital Assignment – 2

BY

MD AZHAR BASHA J

21MIS0380

BOX PLOT (WITH PACKAGE):

CODE:

import matplotlib.pyplot as plt

data = [15, 28, 33, 35, 45, 50, 60, 65, 70, 80, 90, 95, 105, 120, 130, 150]

plt.boxplot(data)

plt.xlabel('Data')

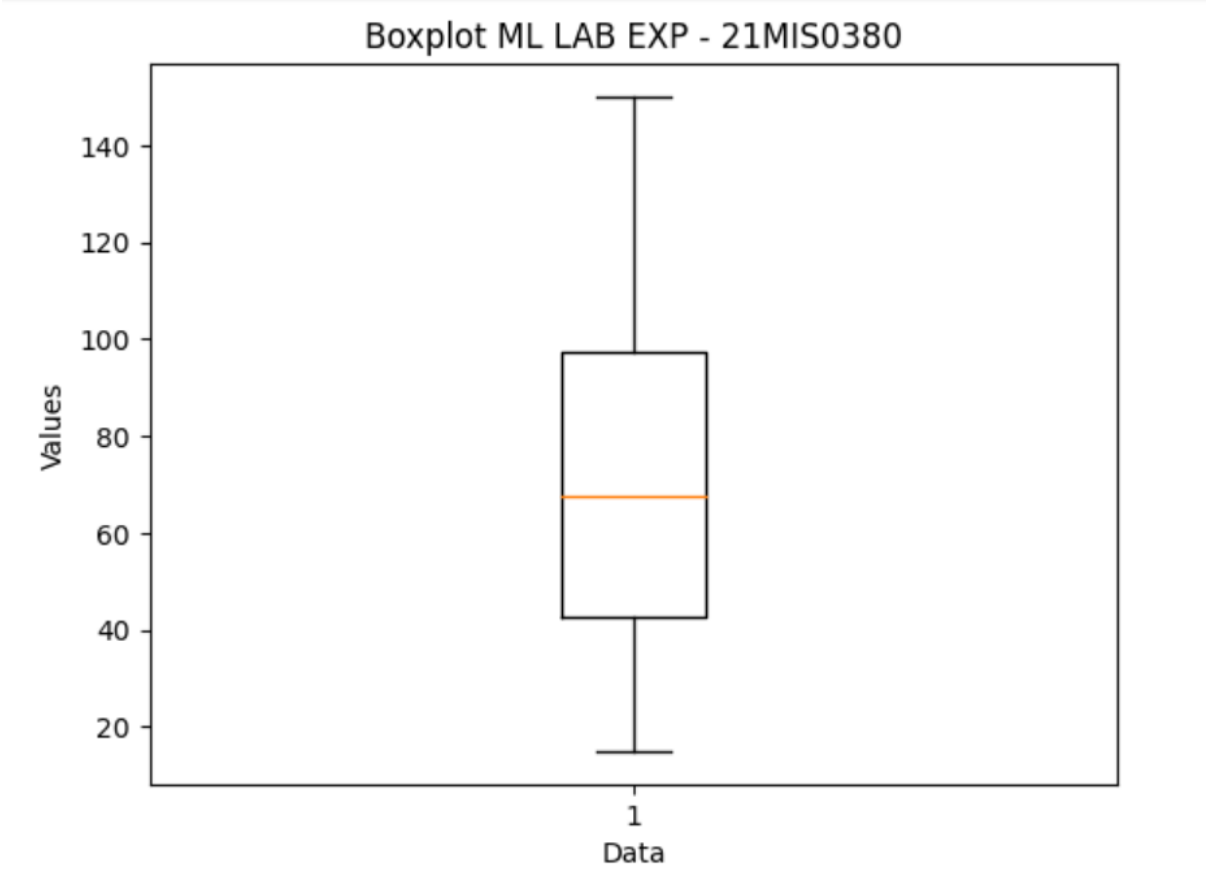
plt.ylabel('Values')

plt.title('Boxplot ML LAB EXP - 21MIS0380')

plt.show()

```
import matplotlib.pyplot as plt

data = [15, 28, 33, 35, 45, 50, 60, 65, 70, 80, 90, 95, 105, 120, 130, 150]

plt.boxplot(data)
plt.xlabel('Data')
plt.ylabel('Values')
plt.title('Boxplot ML LAB EXP - 21MIS0380')
plt.show()
```

OUTPUT:



CONTROL CHART (WITH PACKAGE):

CODE:

```
import matplotlib.pyplot as plt

import numpy as np

data2 = np.array([18, 26, 19, 30, 21, 20, 28, 24, 22, 31, 19, 35, 25, 21, 23, 27, 29])

mean = np.mean(data2)

std_dev = np.std(data2)

lcl = mean - 2 * std_dev

ucl = mean + 2 * std_dev

plt.plot(data2, 'o-', label='Data')
```

plt.axhline(y=mean, color='r', linestyle='--', label='Mean')

plt.axhline(y=lcl, color='g', linestyle='--', label='LCL')

plt.axhline(y=ucl, color='g', linestyle='--', label='UCL')

plt.xlabel('Index')

plt.ylabel('Values')

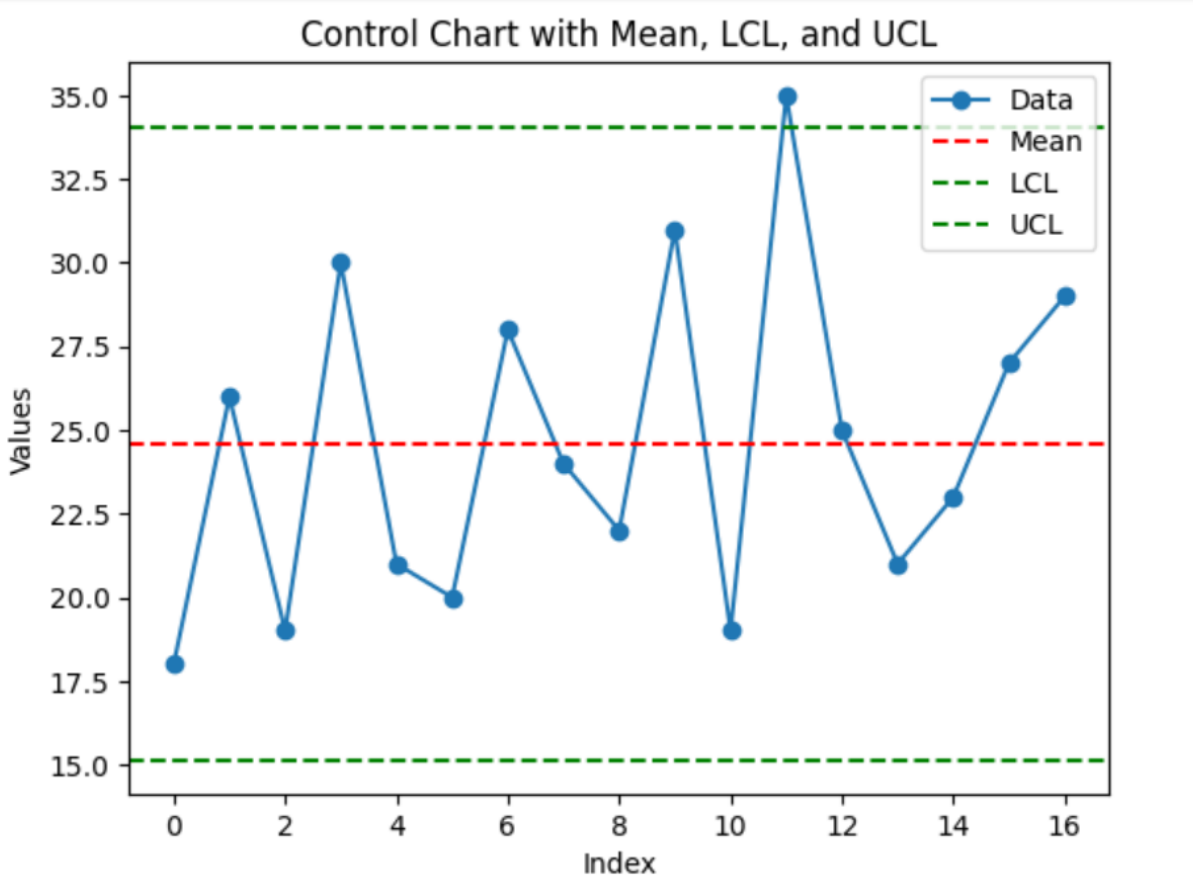
plt.title('Control Chart with Mean, LCL, and UCL')

plt.legend()

plt.show()

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

data2 = np.array([18, 26, 19, 30, 21, 20, 28, 24, 22, 31, 19, 35, 25, 21, 23, 27, 29])

mean = np.mean(data2)
std_dev = np.std(data2)

lcl = mean - 2 * std_dev
ucl = mean + 2 * std_dev

plt.plot(data2, 'o-', label='Data')

plt.axhline(y=mean, color='r', linestyle='--', label='Mean')
plt.axhline(y=lcl, color='g', linestyle='--', label='LCL')
plt.axhline(y=ucl, color='g', linestyle='--', label='UCL')

plt.xlabel('Index')
plt.ylabel('Values')
plt.title('Control Chart with Mean, LCL, and UCL')

plt.legend()

plt.show()
```

OUTPUT:



BOX PLOT (WITHOUT PACKAGES):

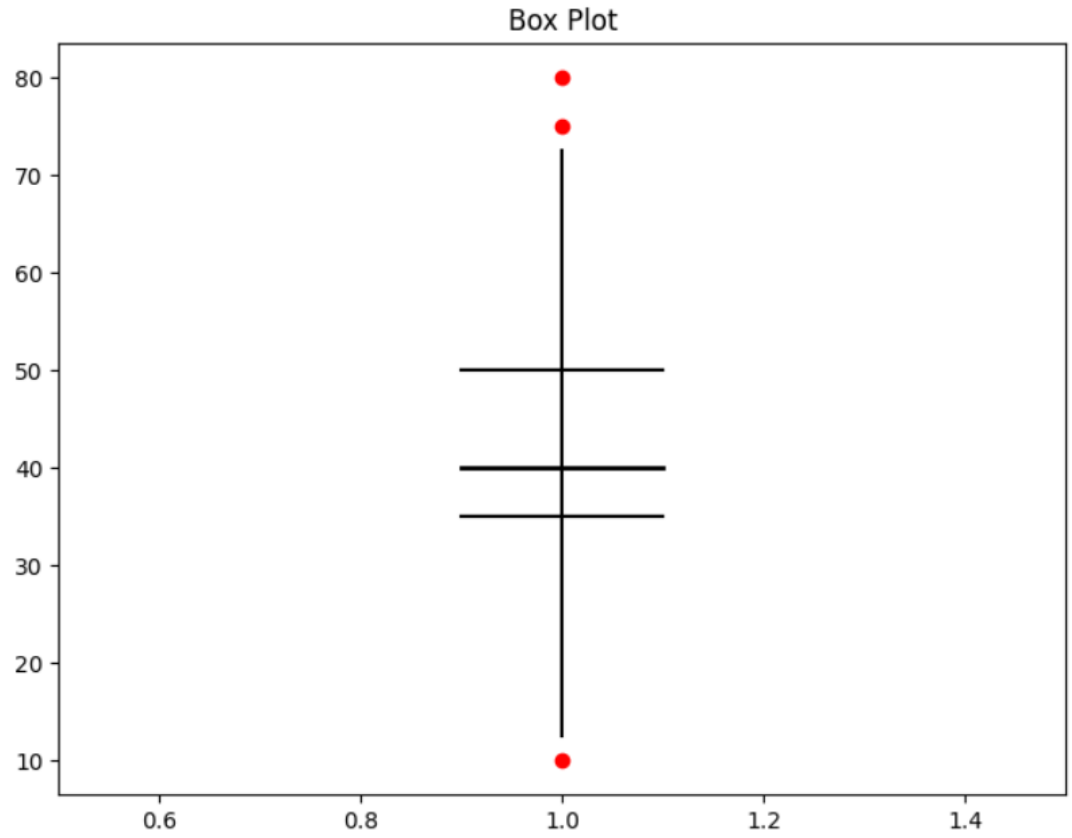
CODE:

```
import matplotlib.pyplot as plt

def calculate_box_plot_stats(data):
    sorted_data = sorted(data)
    n = len(data)
    Q1 = sorted_data[n // 6]
    Q2 = sorted_data[n // 4]
    Q3 = sorted_data[3 * n // 6]
     IQR = Q3 - Q1
    lower_whisker = max(min(data), Q1 - 1.5 * IQR)
```

```
    upper_whisker = min(max(data), Q3 + 1.5 * IQR)
     return Q1, Q2, Q3, lower_whisker, upper_whisker
def draw_box_plot(data):
    Q1, Q2, Q3, lower_whisker, upper_whisker = calculate_box_plot_stats(data)
    plt.figure(figsize=(8, 6))
    plt.plot([1, 1], [Q1, Q3], color='black')
    plt.plot([0.9, 1.1], [Q1, Q1], color='black')
    plt.plot([0.9, 1.1], [Q3, Q3], color='black')
    plt.plot([0.9, 1.1], [Q2, Q2], color='black', linewidth=2)
    plt.plot([1, 1], [lower_whisker, Q1], color='black')
    plt.plot([1, 1], [Q3, upper_whisker], color='black')
    for value in data:
        if value < lower_whisker or value > upper_whisker:
            plt.plot(1, value, 'ro')  # Outliers marked in red
    plt.xlim(0.5, 1.5)
    plt.title('Box Plot')
    plt.show()
data = [10, 25, 35, 40, 45, 48, 50, 52, 55, 60, 75, 80]
draw_box_plot(data)
```

```python
import matplotlib.pyplot as plt

def calculate_box_plot_stats(data):
    sorted_data = sorted(data)
    n = len(data)

    Q1 = sorted_data[n // 6]
    Q2 = sorted_data[n // 4]
    Q3 = sorted_data[3 * n // 6]

    IQR = Q3 - Q1

    lower_whisker = max(min(data), Q1 - 1.5 * IQR)
    upper_whisker = min(max(data), Q3 + 1.5 * IQR)

    return Q1, Q2, Q3, lower_whisker, upper_whisker

def draw_box_plot(data):
    Q1, Q2, Q3, lower_whisker, upper_whisker = calculate_box_plot_stats(data)
    plt.figure(figsize=(8, 6))

    plt.plot([1, 1], [Q1, Q3], color='black')
    plt.plot([0.9, 1.1], [Q1, Q1], color='black')
    plt.plot([0.9, 1.1], [Q3, Q3], color='black')
    plt.plot([0.9, 1.1], [Q2, Q2], color='black', linewidth=2)

    plt.plot([1, 1], [lower_whisker, Q1], color='black')
    plt.plot([1, 1], [Q3, upper_whisker], color='black')

    for value in data:
        if value < lower_whisker or value > upper_whisker:
            plt.plot(1, value, 'ro')  # Outliers marked in red

    plt.xlim(0.5, 1.5)
    plt.title('Box Plot')
    plt.show()

data = [10, 25, 35, 40, 45, 48, 50, 52, 55, 60, 75, 80]

draw_box_plot(data)
```

OUTPUT:



CONTROL CHART (WITHOUT PACKAGES):

CODE:

```
import matplotlib.pyplot as plt

def calculate_control_chart_stats(data):
    mean = sum(data) / len(data)
    std_dev = (sum([(x - mean) ** 2 for x in data]) / len(data)) ** 0.5
    UCL = mean + 3 * std_dev
    LCL = mean - 3 * std_dev
    return mean, UCL, LCL

def draw_control_chart(data):
    mean, UCL, LCL = calculate_control_chart_stats(data)
    plt.figure(figsize=(12, 6))
```

```
    plt.plot(data, marker='o', linestyle='-', color='b')
    plt.axhline(y=mean, color='g', linestyle='-', label='Mean')
    plt.axhline(y=UCL, color='r', linestyle='-', label='UCL')
    plt.axhline(y=LCL, color='r', linestyle='-', label='LCL')
    plt.text(len(data) - 1, mean, 'Mean', color='g', va='center')
    plt.text(len(data) - 1, UCL, 'UCL', color='r', va='center')
    plt.text(len(data) - 1, LCL, 'LCL', color='r', va='center')
    plt.title('Control Chart')
    plt.xlabel('Sample')
    plt.ylabel('Value')
    plt.legend()
    plt.show()
data = [15, 18, 20, 22, 17, 19, 25, 27, 21, 23, 18, 16, 20, 24, 22, 25, 26, 19, 20, 23]
draw_control_chart(data)
```

```python
import matplotlib.pyplot as plt

def calculate_control_chart_stats(data):
    mean = sum(data) / len(data)
    std_dev = (sum([(x - mean) ** 2 for x in data]) / len(data)) ** 0.5
    UCL = mean + 3 * std_dev
    LCL = mean - 3 * std_dev
    return mean, UCL, LCL

def draw_control_chart(data):
    mean, UCL, LCL = calculate_control_chart_stats(data)
    plt.figure(figsize=(12, 6))

    plt.plot(data, marker='o', linestyle='-', color='b')

    plt.axhline(y=mean, color='g', linestyle='-', label='Mean')
    plt.axhline(y=UCL, color='r', linestyle='-', label='UCL')
    plt.axhline(y=LCL, color='r', linestyle='-', label='LCL')

    plt.text(len(data) - 1, mean, 'Mean', color='g', va='center')
    plt.text(len(data) - 1, UCL, 'UCL', color='r', va='center')
    plt.text(len(data) - 1, LCL, 'LCL', color='r', va='center')
```

```
    plt.title('Control Chart')
    plt.xlabel('Sample')
    plt.ylabel('Value')

    plt.legend()

    plt.show()

data = [15, 18, 20, 22, 17, 19, 25, 27, 21, 23, 18, 16, 20, 24, 22, 25, 26, 19, 20, 23]

draw_control_chart(data)
```

OUTPUT: