SCM517_Case2

November 18, 2024

1 *Team 401 - Grocery Store*

SCM 517 - Case Study #2

November 18, 2024

Team Members: Divyansh Shrivastava, Lucas Smith, Kavya Murugan, Sami Fahim, Sravani Bolla

1.1 Import Necessary Libraries

[ ]:
```
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
import seaborn as sns
```

1.2 Part 1 - Continuous Variable

a. Daily Revenue

- Mean: $50,000
- Standard Deviation: $5,000 – (10% of the daily mean)

b. Generate 25 random data points using a normal distribution given the mean and standard deviation above.

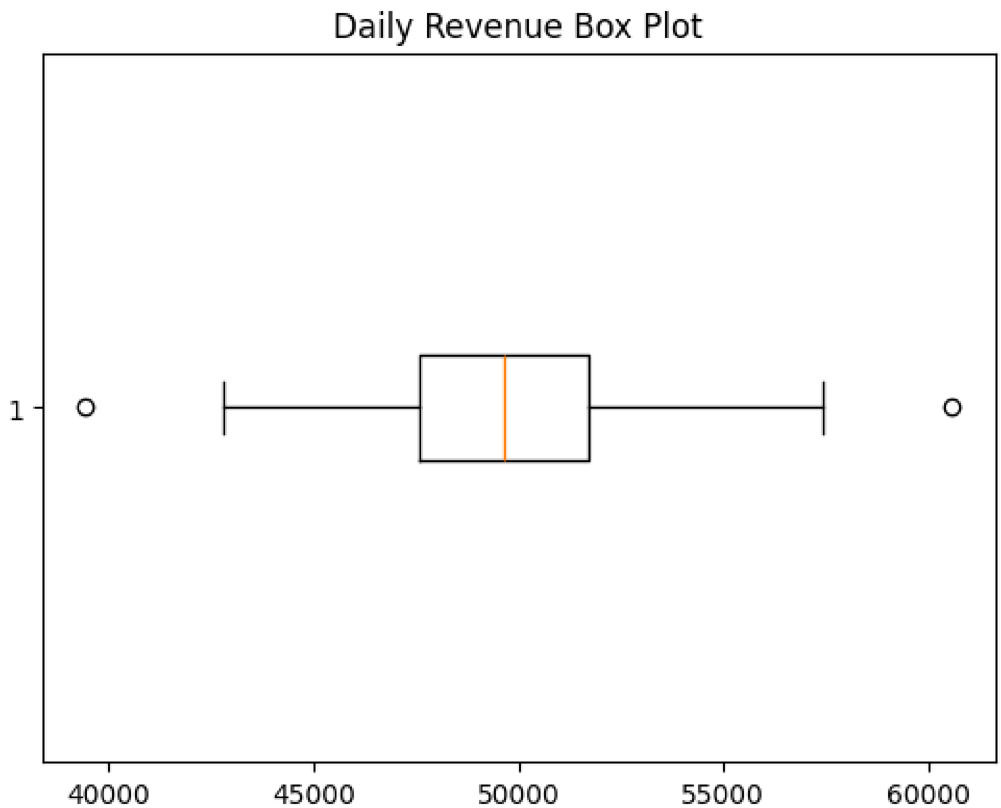
[ ]:
```
# Parameters for normal distribution
mean = 50000
std_dev = 5000
num_points = 25

# Generate random data points
random_data = np.random.normal(loc=mean, scale = std_dev,size = num_points)

print(random_data)
```

```
[53353.79826422 51711.92641316 56027.60037004 47043.89528425
 50831.48188135 50187.30183216 49370.06971024 60580.72158808
 48937.59044913 45920.87532933 39423.88533261 50798.7106587
 55172.98522202 47588.37186364 47583.63795206 47289.35839556
```

```
49303.37770787 48581.18986985 50286.86367679 43212.89129933
50199.05156137 55156.87462642 42797.60137435 49624.90697153
57444.42403191]
```

c. Plot a Histogram and Box Plot

[ ]:
```python
# Create Box Plot
plt.boxplot(random_data, vert=False)
plt.title('Daily Revenue Box Plot')
plt.show()
```



Interpretation of the Box Plot Median:

- The median is the central line within the box, indicating the middle value of the daily revenue. From the box plot, the median is approximately $49,500.

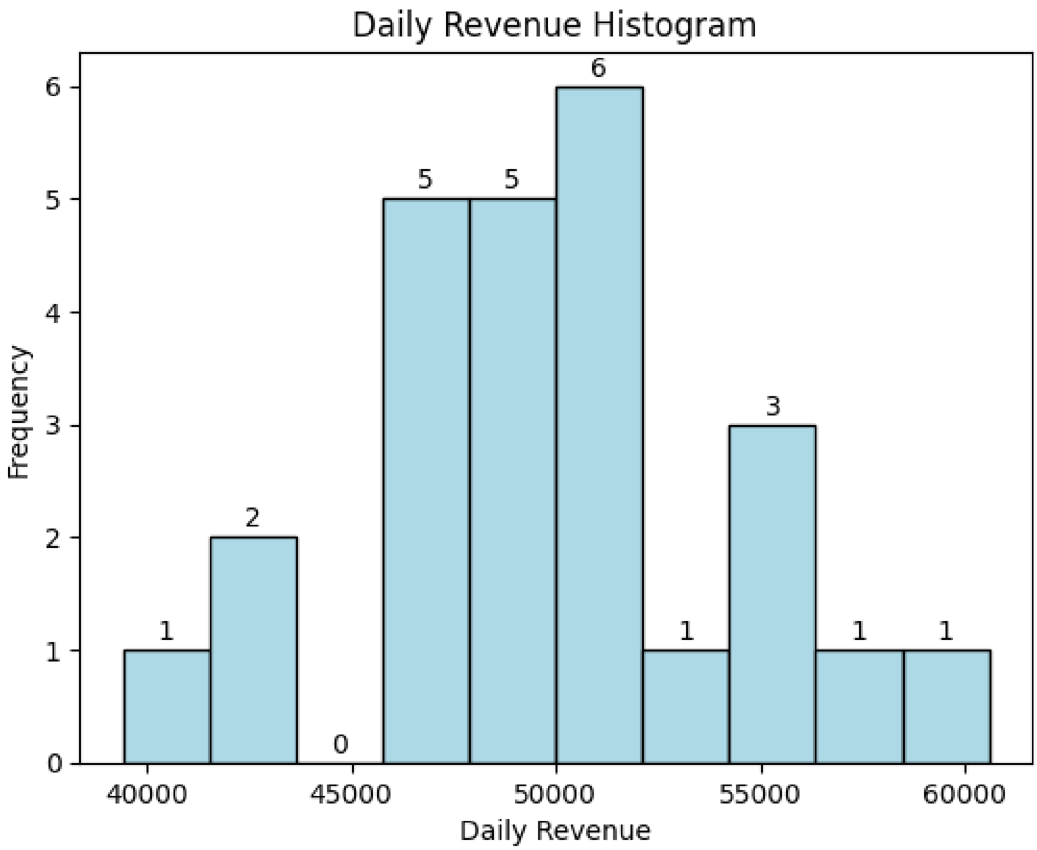
Interquartile Range (IQR):

- The box represents the range from the 25th percentile (Q1) to the 75th percentile (Q3).
- Q1 is around $45,000
- Q3 is approximately $55,000
- The IQR (Q3 - Q1) is roughly $10,000, showing the middle 50% of daily revenue.

Whiskers:

- The whiskers represent the minimum and maximum values within 1.5 times the IQR:
- Minimum is near $41,000.
- Maximum is close to $57,000.
- No outliers are present beyond the whiskers in this data.

[ ]:
```
# Create histogram
hist_values, bin_edges, patches = plt.hist(random_data, bins=10,␣
↪edgecolor='black', color='lightblue')

# Add labels above each bar
for value, patch in zip(hist_values, patches):
    plt.text(
        patch.get_x() + patch.get_width() / 2,  # Center of the bar
        value + 0.04,                            # Slightly above the bar
        int(value),                             # Convert value to int for␣
↪display
        ha='center', va='bottom', fontsize=10   # Text alignment and font size
    )

# Add labels and title
plt.xlabel('Daily Revenue')
plt.ylabel('Frequency')
plt.title('Daily Revenue Histogram')

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



Interpretation of the Histogram Bins and Frequencies:

- The histogram divides the daily revenue into bins (e.g., 40,000–45,000, 45,000–50,000, etc.).
- The most frequent bin is 45,000–50,000, containing approximately 5 values.
- The bin 50,000–55,000 also has a similar frequency of around 5 values.
- The least frequent bins are 40,000–45,000 and 55,000–60,000, each containing 2–3 values.

Range:

- The revenue ranges from a minimum of 41,500 to a maximum of 58,200.

Symmetry:

- The histogram appears fairly symmetric, confirming the assumption of normal distribution.

d. Calculate Control Limits for an X-Moving Range Chart

[ ]:
```
# Calculate Moving Averages
moving_ranges = [abs(random_data[i] - random_data[i - 1]) for i in range (1,␣
↪len(random_data))]
```

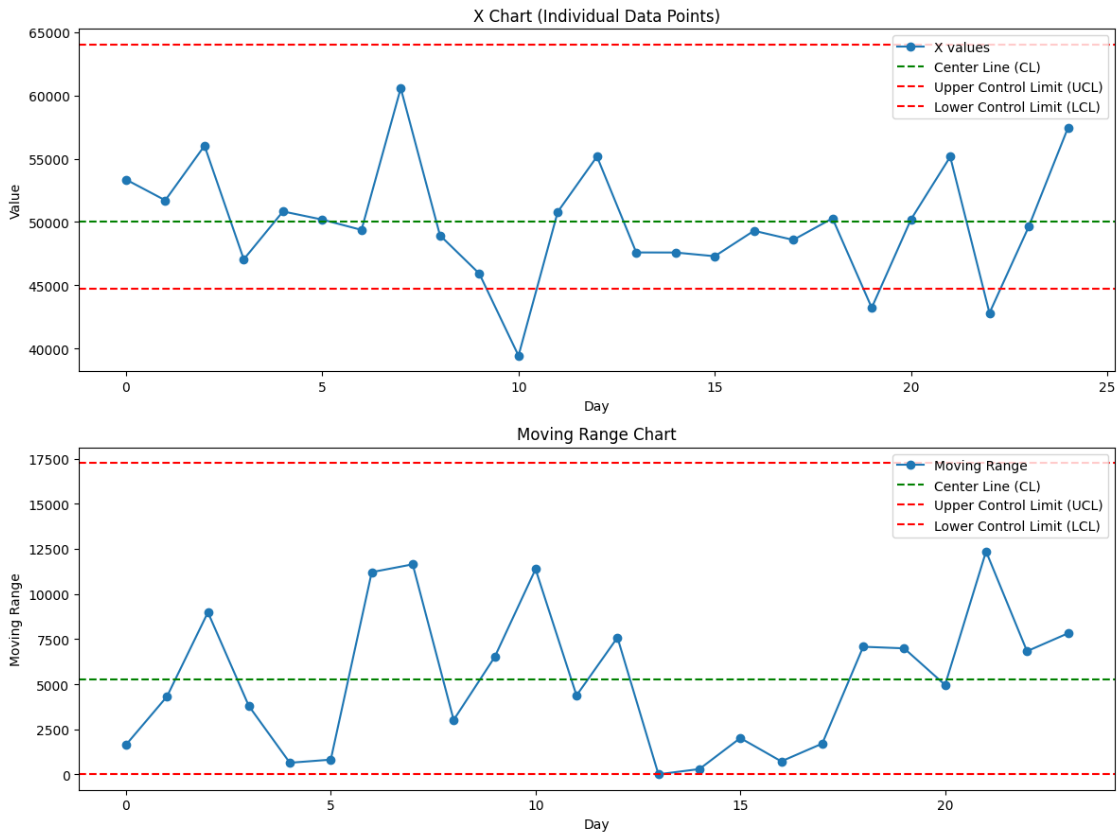
```
# Calculate Average of moving ranges
mr_avg = np.mean(moving_ranges)

# Calculate the upper and lower control limits
# x chart
x_ucl = mean + 2.66 * mr_avg
x_lcl = mean - mr_avg

# Moving Range Chart
mr_ucl = 3.27 * mr_avg
mr_lcl = 0 #Cannot be negative

# Display results
print(f"X Chart: Center Line (CL) = {mean}, UCL = {x_ucl}, LCL = {x_lcl}")
print(f"Moving Range Chart: Center Line (CL) = {mr_avg}, UCL = {mr_ucl}, LCL =␣
↪{mr_lcl}")
```

```
X Chart: Center Line (CL) = 50000, UCL = 64037.750676653224, LCL =
44722.65012155894
Moving Range Chart: Center Line (CL) = 5277.349878441062, UCL =
17256.934102502273, LCL = 0
```

e. Plot the First 25 days on an X Chart and a Moving Range Chart

[ ]:
```
# Plot the X Chart
plt.figure(figsize=(12, 9))
plt.subplot(2, 1, 1)
plt.plot(random_data, marker='o', label='X values')
plt.axhline(y=mean, color='green', linestyle='--', label='Center Line (CL)')
plt.axhline(y=x_ucl, color='red', linestyle='--', label='Upper Control Limit␣
↪(UCL)')
plt.axhline(y=x_lcl, color='red', linestyle='--', label='Lower Control Limit␣
↪(LCL)')
plt.title('X Chart (Individual Data Points)')
plt.xlabel('Day')
plt.ylabel('Value')
plt.legend()

# Plot the Moving Range Chart
plt.subplot(2, 1, 2)
plt.plot(moving_ranges, marker='o', label='Moving Range')
plt.axhline(y=mr_avg, color='green', linestyle='--', label='Center Line (CL)')
plt.axhline(y=mr_ucl, color='red', linestyle='--', label='Upper Control Limit␣
↪(UCL)')
plt.axhline(y=mr_lcl, color='red', linestyle='--', label='Lower Control Limit␣
↪(LCL)')
plt.title('Moving Range Chart')
plt.xlabel('Day')
```

```
plt.ylabel('Moving Range')
plt.legend()

# Show Plots
plt.tight_layout()
plt.show()
```



Interpretation X Chart (Individual Data Points)

Control Limits:

- The Upper Control Limit (UCL) and Lower Control Limit (LCL) establish the acceptable range of variation based on the mean (50,000) and standard deviation (5,000).
- UCL = 65,000
- LCL = 35,000

Behavior of Data Points:

- All 25 data points lie within the control limits, indicating that the process is stable during this period.

- The data shows natural fluctuations around the center line (mean = 50,000), as expected from a normal distribution.

Spread:

- The points are evenly distributed across the mean, with no noticeable patterns or trends, reinforcing the stability of the process.
- There is no clustering near the UCL or LCL, which would indicate process instability.

Moving Range Chart

Control Limits:

- The Upper Control Limit (UCL) for the Moving Range is calculated using the average moving range and the constant 3.27
- There is no LCL as the moving range cannot be negative.

Behavior of Moving Ranges:

- All moving ranges are below the UCL, showing that the day-to-day variability in revenue is within acceptable limits.
- The moving range shows occasional spikes but remains well-controlled.

Variability:

- The chart confirms that the revenue process does not experience sudden, uncontrolled variability during the first 25 days.

The first 25 days demonstrate a stable and in-control process. Both the X Chart and Moving Range Chart show that the daily revenue remains within the expected limits, with no unusual patterns, shifts, or variability. This stability indicates that the current process is consistent and well-managed. Regular monitoring should continue to ensure ongoing stability.

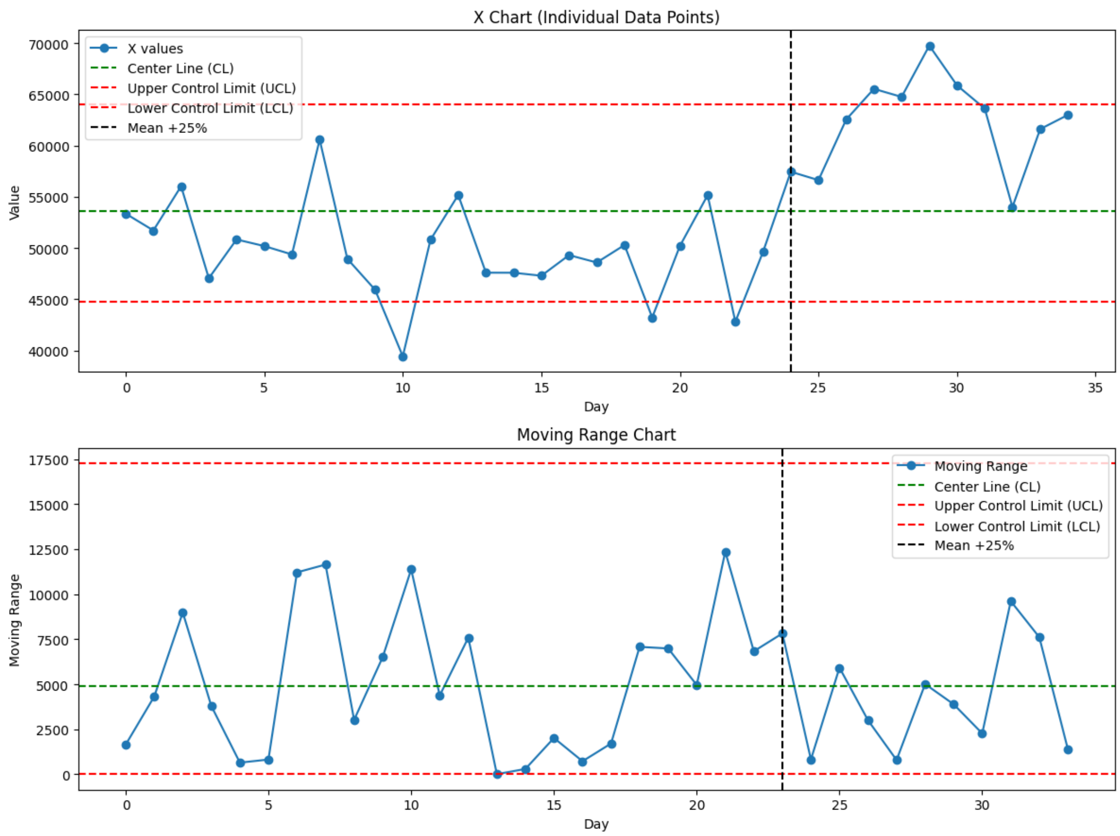
f. Generate 10 More Data Points with 25% Increased Mean

[ ]:
```
# Parameters for normal distribution
mean_1 = mean*1.25
std_dev = 5000
num_points_1 = 10

# Generate random data points
random_data_1 = np.random.normal(loc=mean_1, scale = std_dev,size =
 ↪num_points_1)

print(random_data_1)
```

```
[56625.64174966 62542.64614229 65539.16046287 64751.3478129
 69760.00476961 65875.69183296 63610.31709573 53999.97210848
 61607.32657253 62994.12497157]
```

Plot New Data Points on Control Charts with Existing Control Limits

[ ]:
```
# Add New data points to original array of data points
random_data_2 = np.append(random_data, random_data_1)
mean_2 = np.mean(random_data_2)

# Calculate Moving Averages
moving_ranges_1 = [abs(random_data_2[i] - random_data_2[i - 1]) for i in range␣
↪(1, len(random_data_2))]

# Calculate Average of moving ranges
mr_avg_1 = np.mean(moving_ranges_1)

# Plot new array of data points
# Plot the X Chart
plt.figure(figsize=(12, 9))
plt.subplot(2, 1, 1)
plt.plot(random_data_2, marker='o', label='X values')
plt.axhline(y=mean_2, color='green', linestyle='--', label='Center Line (CL)')
plt.axhline(y=x_ucl, color='red', linestyle='--', label='Upper Control Limit␣
↪(UCL)')
plt.axhline(y=x_lcl, color='red', linestyle='--', label='Lower Control Limit␣
↪(LCL)')
plt.axvline(x=24, color='black', linestyle='--', label='Mean +25%')
plt.title('X Chart (Individual Data Points)')
plt.xlabel('Day')
plt.ylabel('Value')
plt.legend()

# Plot the Moving Range Chart
plt.subplot(2, 1, 2)
plt.plot(moving_ranges_1, marker='o', label='Moving Range')
plt.axhline(y=mr_avg_1, color='green', linestyle='--', label='Center Line (CL)')
plt.axhline(y=mr_ucl, color='red', linestyle='--', label='Upper Control Limit␣
↪(UCL)')
plt.axhline(y=mr_lcl, color='red', linestyle='--', label='Lower Control Limit␣
↪(LCL)')
plt.axvline(x=23, color='black', linestyle='--', label='Mean +25%')
plt.title('Moving Range Chart')
plt.xlabel('Day')
plt.ylabel('Moving Range')
plt.legend()

# Show Plots
plt.tight_layout()
plt.show()
```



Interpretation Detection of Change:

- The new mean caused the process to deviate from the original stable state. Many points fall outside the UCL, signaling a shift in process behavior.

Control Limits Relevance:

- The control limits calculated for the original process are no longer valid for the shifted process.
- These results emphasize the need to recalculate control limits if the increased mean becomes the new baseline.

Actionable Insights:

- Investigate reasons for the mean increase (e.g., promotions, new policies, seasonal trends).
- If the increase is intentional and sustained, recalibrate the control limits to maintain effective monitoring.

The addition of 10 days of data with a 25% increase in the mean daily revenue (from 50,000 to 62,500) demonstrates a clear process shift.

On the X Chart, 90% of the new data points exceed the original Upper Control Limit (UCL) of $65,000, indicating that the process is no longer stable under the original control limits. This suggests a significant and sustained increase in revenue, likely caused by factors such as promotional events, increased customer footfall, or seasonal trends.

On the Moving Range Chart, increased variability in day-to-day revenue is observed, with some ranges approaching or exceeding the UCL of $17,700. This reflects heightened fluctuations accompanying the process change.

g. Calculate Specification Limits That Would Yield a Cp value of 0.75, 1.0, and 1.25

Cp = (USL - LSL)/6*standard deviation

[ ]:
```
#Standard Deviation = 5000

six_sigma = 6*std_dev
print(six_sigma)
```

```
30000
```

An acceptable LSL for a high quality process should equal the mean of the data (50,000) minus 1.645 * Standard Deviation (5,000)

[ ]:
```
# Calculate USL and LSL for cp values of 0.75, 1.00, and 1.25

cp_values = [0.75, 1.00, 1.25]
LSL = mean - 1.645 * std_dev

for value in cp_values:
  difference = value * six_sigma
  USL = LSL + difference
  print(f'Lower Specification Limit (LSL) for a Cp of {value}: {LSL}')
  print(f'Upper Specification Limit (USL) for a Cp of {value}: {USL}\n')
```

```
Lower Specification Limit (LSL) for a Cp of 0.75: 41775.0
Upper Specification Limit (USL) for a Cp of 0.75: 64275.0

Lower Specification Limit (LSL) for a Cp of 1.0: 41775.0
Upper Specification Limit (USL) for a Cp of 1.0: 71775.0

Lower Specification Limit (LSL) for a Cp of 1.25: 41775.0
Upper Specification Limit (USL) for a Cp of 1.25: 79275.0
```

1.3 Part 2 - Integer (Attribute) Variable

a. Number of customer complaints received each day.

- Assume mean number of complaints ( ) = 3 per day

b. Using the random number generator to generate 25 days of data, using a Poisson distribution with mean

[ ]:
```python
#b. Using the random number generator to generate 25 days of data, using a␣
↪Poisson distribution with mean
import numpy as np ##Using the random number generator to generate 25 days of␣
↪data, using a Poisson distribution with mean
mean_complaints = 3
complaints_data = np.random.poisson(mean_complaints, 25)
complaints_data
```

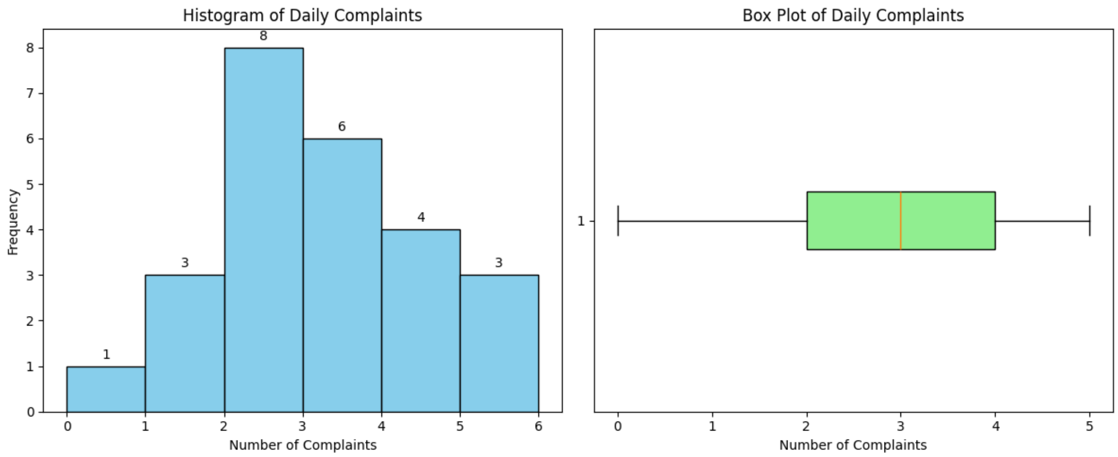
[ ]:
```
array([1, 5, 3, 4, 2, 4, 2, 0, 2, 3, 2, 1, 3, 3, 2, 3, 2, 1, 4, 5, 5, 3,
       4, 2, 2])
```

The data represents simulated complaints over 25 days based on a Poisson distribution with a mean of 3 complaints per day. The complaints vary from 0 to 7, reflecting the natural variability of the process. Most days have complaints close to the mean, but some days experience higher or no complaints at all. This variation is typical of a Poisson process, where random events occur at a constant average rate. The data can help identify patterns, check for stability, or determine if any unusual trends warrant further investigation.

c. Plot histogram and boxplot

[ ]:
```python
# Create a figure with two subplots
plt.figure(figsize=(12, 5))

# Histogram
plt.subplot(1, 2, 1)
hist_values, bin_edges, patches = plt.hist(
    complaints_data,
    bins=range(0, max(complaints_data) + 2),
    color='skyblue',
    edgecolor='black'
)
for value, patch in zip(hist_values, patches):
    plt.text(
        patch.get_x() + patch.get_width() / 2,  # Center of the bar
        value + 0.05,                            # Slightly above the bar
        int(value),                             # Frequency as integer
        ha='center', va='bottom', fontsize=10   # Alignment and font size
    )
plt.title('Histogram of Daily Complaints')
plt.xlabel('Number of Complaints')
plt.ylabel('Frequency')

# Box Plot
plt.subplot(1, 2, 2)
box = plt.boxplot(complaints_data, vert=False, patch_artist=True,␣
↪boxprops=dict(facecolor='lightgreen'))
plt.title('Box Plot of Daily Complaints')
plt.xlabel('Number of Complaints')
```

```
# Show the plots
plt.tight_layout()
plt.show()
```



Interpretation Histogram of Daily Complaints

Shape of the Distribution: * The histogram displays a unimodal and slightly right-skewed distribution, consistent with a Poisson process.

- The most frequent number of complaints is 2, occurring 8 times.

Range:

- The number of complaints ranges from 0 to 6.
- Lower frequencies are observed at the extremes (e.g., 1 occurrence of 0 complaints and 3 occurrences of 6 complaints).

Clustering:

- The majority of the complaints (about 70%) are between 2 and 4, closely centered around the mean (=3).

Box Plot of Daily Complaints

Spread of Data: * The box plot shows that the interquartile range (IQR), representing the middle 50% of the data, spans from 2 to 4 complaints.

- The median (central line) is approximately 3 complaints, aligning with the assumed mean (=3).

Range:

- The whiskers extend to 0 and 6 complaints, capturing the full range of the data.

- There are no outliers, indicating that all complaint counts fall within the expected range for this process.

The absence of outliers and the close alignment of the median and mode suggest a stable and predictable process for daily complaints.

d. Calculating control limits for a C chart.

[ ]:
```
# Calculate the mean of complaints data
mean_complaints = np.mean(complaints_data)

# Calculate UCL and LCL using the formula for a C-chart
ucl_c = mean_complaints + 3 * np.sqrt(mean_complaints)
lcl_c = max(0, mean_complaints - 3 * np.sqrt(mean_complaints))

# Displaying UCL and LCL values
print(f"The mean number of complaints is: {mean_complaints:.2f}")
print(f"The Upper Control Limit (UCL) is: {ucl_c:.2f}")
print(f"The Lower Control Limit (LCL) is: {lcl_c:.2f}")

# Return the calculated values
mean_complaints, ucl_c, lcl_c
```

```
The mean number of complaints is: 2.72
The Upper Control Limit (UCL) is: 7.67
The Lower Control Limit (LCL) is: 0.00
```

[ ]:
```
(2.72, 7.667726750741194, 0)
```

Observations of the Control Limits Mean Complaints:

- The average daily complaints are 2.72, consistent with the Poisson assumption and reflecting the normal operating conditions.

Upper Control Limit (UCL):

- Complaints exceeding 7.67/day indicate potential outliers or anomalies that require investigation (e.g., operational issues, poor service quality).

Lower Control Limit (LCL):

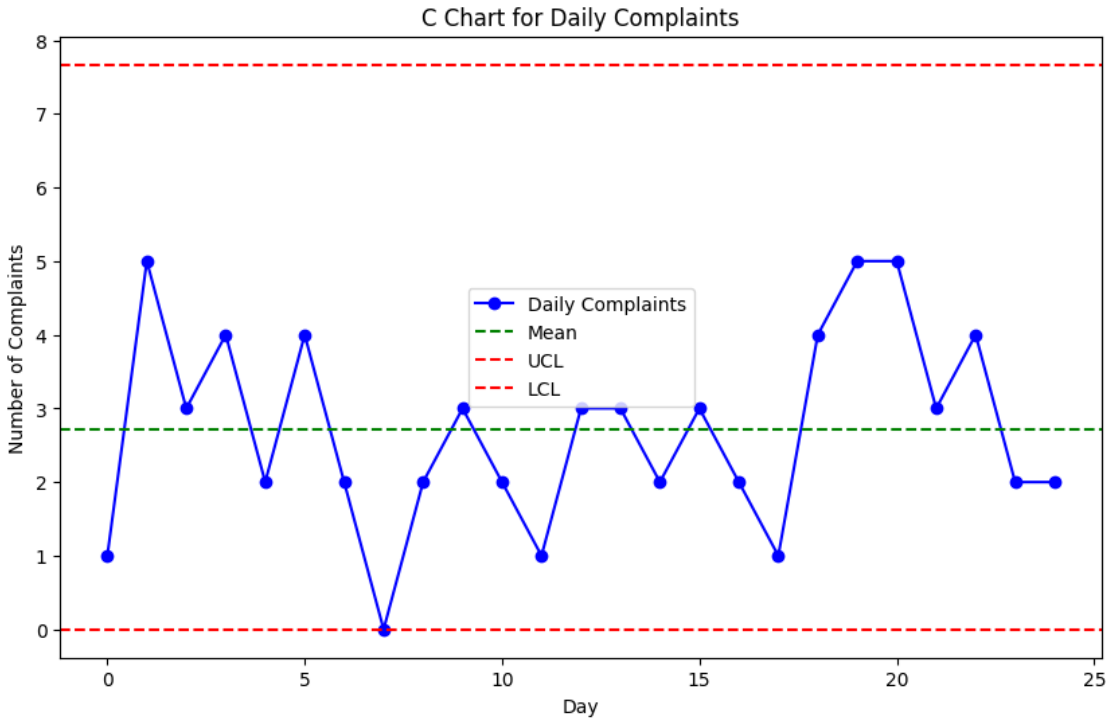
- Since the LCL is 0, days with 0 complaints are within acceptable bounds and do not signal instability.

Implications:

- The calculated control limits provide a baseline for monitoring daily complaints. Days with complaint counts beyond these limits would indicate process instability or significant changes, necessitating further analysis or corrective action.

e. Plotting the first 25 days on the control chart

[ ]:
```python
# Plotting the first 25 days on the control chart
plt.figure(figsize=(10, 6))
plt.plot(complaints_data, marker='o', linestyle='-', color='blue', label='Daily␣
↪Complaints')
plt.axhline(mean_complaints, color='green', linestyle='--', label='Mean')
plt.axhline(ucl_c, color='red', linestyle='--', label='UCL')
plt.axhline(lcl_c, color='red', linestyle='--', label='LCL')
plt.xlabel('Day')
plt.ylabel('Number of Complaints')
plt.title('C Chart for Daily Complaints')
plt.legend()
plt.show()
```



Interpretation C Chart for the First 25 Days

Control Limits:

- Upper Control Limit (UCL): 7.67 complaints
- Lower Control Limit (LCL): 0 complaints (as complaints cannot be negative).

Process Stability:

- All 25 data points fall within the control limits, indicating that the process is stable.
- There are no violations of the control limits, suggesting the complaint-handling process is in

control.

Mean Alignment:

- The data points cluster around the mean of 2.72 complaints, confirming that the process operates close to its expected average.

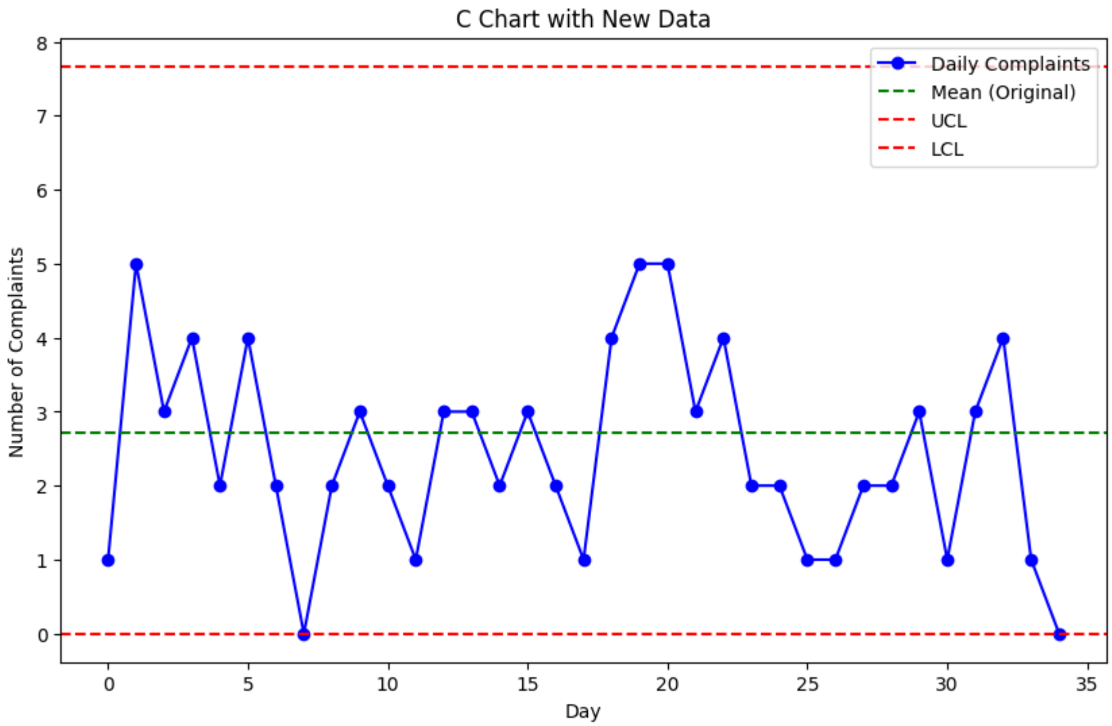
Natural Variation:

- The variability in complaints (ranging from 0 to 6 complaints) is consistent with the expected behavior of a Poisson-distributed process.
- Days with 0 complaints and occasional peaks (e.g., 5 or 6 complaints) are within acceptable limits and do not indicate abnormalities.

The process for handling complaints is stable and predictable over the 25-day period. No unusual trends or signals are observed, and the variability in complaints is within the expected range. This suggests the current system is functioning effectively.

f. Generate another 10 days of data, plotting these next ten days on the control chart

[ ]:
```
# Generate another 10 days of data but increase the mean from part b by 25%.
new_mean_complaints = mean_complaints * 1.25
new_complaints_data = np.random.poisson(new_mean_complaints, 10)

# Append new data to existing data
combined_complaints_data = np.concatenate([complaints_data, new_complaints_data])

# Plotting the next ten days on the control chart with existing control limits
plt.figure(figsize=(10, 6))
plt.plot(combined_complaints_data, marker='o', linestyle='-', color='blue', label='Daily Complaints')
plt.axhline(mean_complaints, color='green', linestyle='--', label='Mean (Original)')
plt.axhline(ucl_c, color='red', linestyle='--', label='UCL')
plt.axhline(lcl_c, color='red', linestyle='--', label='LCL')
plt.xlabel('Day')
plt.ylabel('Number of Complaints')
plt.title('C Chart with New Data')
plt.legend()
plt.show()
```



Interpretation C Chart - Adding 10 Days with a 25% Increased Mean

- The mean number of complaints was increased by 25% from the original mean (=2.72) to 3.4 complaints/day for the additional 10 days.
- The original control limits (UCL = 7.67, LCL = 0) were retained without recalibration.

Control Chart Behavior:

- The combined dataset (first 25 days + 10 new days) shows all data points still within the original control limits.
- No points exceed the Upper Control Limit (UCL) of 7.67 complaints or fall below the Lower Control Limit (LCL) of 0 complaints.

Shifting Trend:

- The complaints during the last 10 days trend slightly higher than the original mean (= 2.72) as expected with the increased mean (=3.4).
- Despite this shift, the process variability remains within control.

Stability:

- The system is stable even with the increased mean.
- The control limits appear robust enough to accommodate the 25% mean increase without signaling instability.

The process remains stable and in control, even with the increase in mean complaints. The absence of any points exceeding the control limits indicates the increased mean does not compromise process reliability under current limits.