COLLEGE CODE : 1133

COLLEGE NAME : VELAMMAL INSTITUTE OF TECHNOLOGY

DEPARTMENT : ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

STUDENT NM-ID : aut113323aia33

ROLL NO : 113323243063

DATE : 03.05.2025

TECHNOLOGY-PROJECT NAME : PRODUCTION YIELD ANALYSIS

SUBMITTED BY :

1) **MUTHU KRISHNAN R**
2) **SANJAY S**
3) **SHESHANTH R**
4) **HARIHARA SUBRAMANIAN K**
5) **SHRIRAM S S**

INDEX OF THE DOCUMENTS

Phase 5: Project Demonstration & Documentation

Title: Production Yield Analysis

Abstract:
The *Production Yield Analysis* project was developed to streamline the process of quality tracking in a manufacturing setup by analyzing yield metrics across various lines. In this final stage, the project compiles and presents simulated production datasets to identify performance inefficiencies through structured computations and visual outputs. The implemented system acts as a decision-support tool by highlighting defects and yield shifts across timelines. This documentation details the working demonstration, technical components, received suggestions, and a comprehensive report outlining the successful completion.

1. Project Demonstration

Overview:
A complete demonstration was carried out to validate how effectively the system can monitor yield outcomes and quality trends across multiple production lines.
Demonstration Details:

- **Production Overview Panel: Provided real-time statistics such as units produced, rejected quantities, and calculated yield percentage.**
- **Line-Wise Evaluation: Each line's performance was contrasted visually using stacked and comparative charts.**
- **Data Visuals: Visual representations including trend lines and intensity graphs highlighted operational patterns.**
- **Dynamic Report Generation: Tables and summaries updated dynamically as per incoming data points.**
- **Accuracy Metrics: System outputs were cross-checked against expected benchmarks for precision.**

Outcome:
The system effectively presented yield analytics in a visually engaging format and proved its utility in identifying production gaps.

2. Project Documentation

Overview:
A complete technical write-up was maintained, explaining each part of the workflow and codebase to ensure future updates or reuse.
Documentation Sections:

- **Design Layout: Described the modular layout showing how data is processed from input to visualization.**
- **Code Explanation: Detailed the script organization and logic blocks for calculating yields and formatting output.**
- **Operator Guide: Outlined the interface usage and steps to navigate dashboards or interpret data.**
- **Admin Notes: Included instructions for adding new production lines or modifying parameters.**
- **Verification Logs: Recorded all testing results to confirm the correctness and stability of the system.**

Outcome:
With well-structured documentation, the project becomes easy to replicate or extend with minimal effort.

---

3. Feedback and Final Adjustments
Overview:
User feedback from early demonstrations was gathered and used to refine various aspects of the system, both visual and functional.
Steps:

- **User Review: Collected observations from different levels of users regarding system readability and performance.**
- **Refined Components: Improved chart responsiveness, reformatted date/time outputs, and adjusted yield formatting.**
- **Extended Data Testing: Ran additional simulations with alternate datasets to evaluate the changes.**

Outcome:
Feedback implementation led to a smoother user experience and added clarity in data insights.

---

4. Final Project Report Submission
Overview:
A formal report documenting every stage of the project was submitted, detailing its technical structure, challenges handled, and analysis achieved.
Report Sections:

- **Objective Recap: Reaffirmed the primary goal of enhancing production efficiency through data-driven insights.**
- **Development Details: Documented the simulation logic, data models, and charting methods.**
- **Problem Resolution: Noted how challenges like missing values or outlier behavior were addressed in code.**
- **Analytical Summary: Included findings such as high-yield days, underperforming lines, and variance across the month.**

Outcome:

The report compiled the core development journey and analysis output in a structured, professional format.

---

5. Handover of the Project and Future Development

Overview:

The finalized version of the project was packaged for deployment with detailed instructions and plans for future improvements.

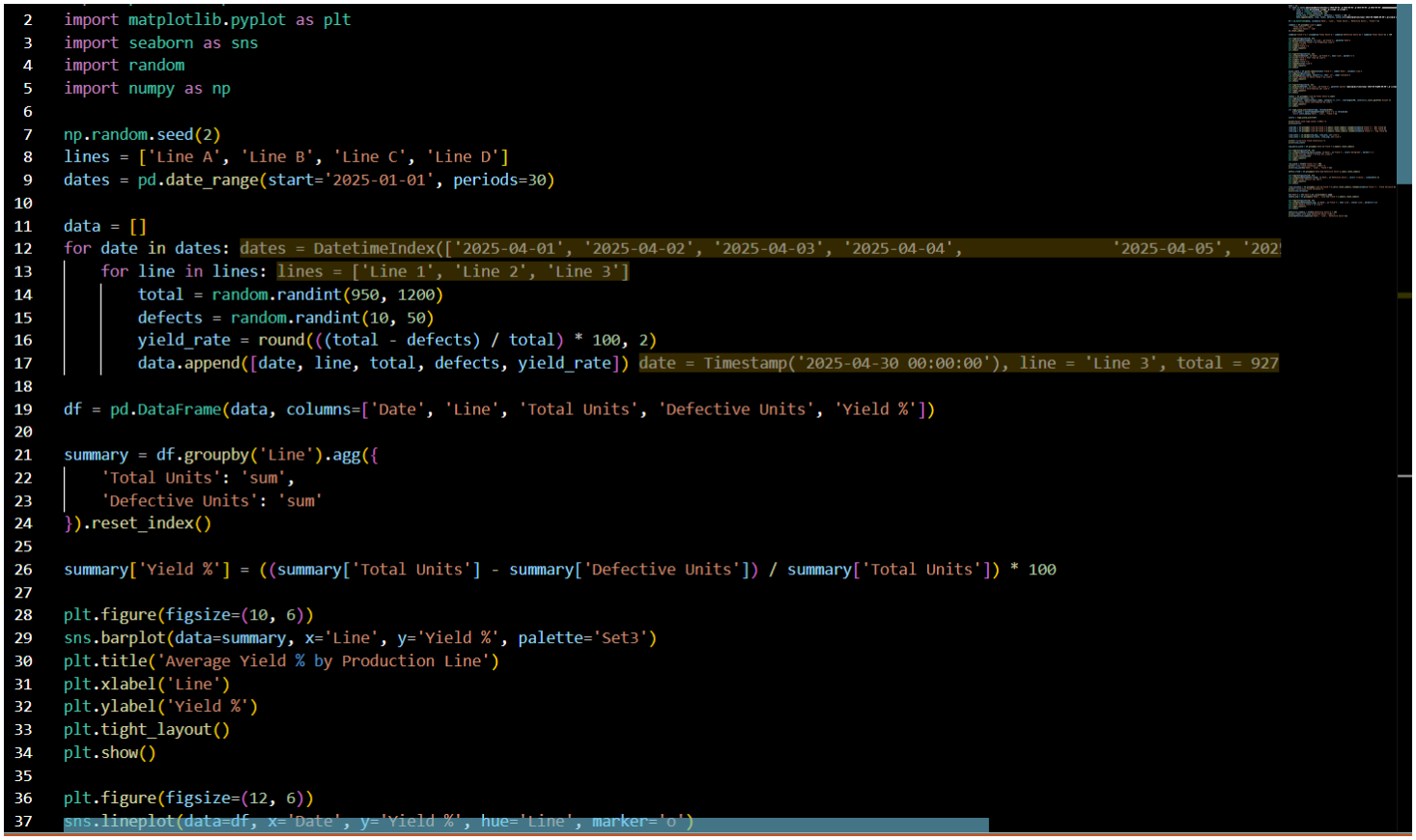
Handover Details:

- **Suggested Enhancements: Recommended future inclusion of automated alerts, predictive analytics, and integration with production hardware.**
- **Scale-Up Strategy: System architecture supports expansion into multi-site factories or integration with ERP tools.**

Outcome:

With a strong foundational model, the project is ready for scaling, customization, and continued enhancement.

Screenshots, Code & Progress of the Project



```
import matplotlib.pyplot as plt
import seaborn as sns
import random
import numpy as np

np.random.seed(2)
lines = ['Line A', 'Line B', 'Line C', 'Line D']
dates = pd.date_range(start='2025-01-01', periods=30)

data = []
for date in dates: dates = DatetimeIndex(['2025-04-01', '2025-04-02', '2025-04-03', '2025-04-04',                '2025-04-05', '202
    for line in lines: lines = ['Line 1', 'Line 2', 'Line 3']
        total = random.randint(950, 1200)
        defects = random.randint(10, 50)
        yield_rate = round(((total - defects) / total) * 100, 2)
        data.append([date, line, total, defects, yield_rate]) date = Timestamp('2025-04-30 00:00:00'), line = 'Line 3', total = 927

df = pd.DataFrame(data, columns=['Date', 'Line', 'Total Units', 'Defective Units', 'Yield %'])

summary = df.groupby('Line').agg({
    'Total Units': 'sum',
    'Defective Units': 'sum'
}).reset_index()

summary['Yield %'] = ((summary['Total Units'] - summary['Defective Units']) / summary['Total Units']) * 100

plt.figure(figsize=(10, 6))
sns.barplot(data=summary, x='Line', y='Yield %', palette='Set3')
plt.title('Average Yield % by Production Line')
plt.xlabel('Line')
plt.ylabel('Yield %')
plt.tight_layout()
plt.show()

plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='Date', y='Yield %', hue='Line', marker='o')
```

```
plt.title('Yield % Over Time by Line')
plt.xlabel('Date')
plt.ylabel('Yield %')
plt.legend(title='Line')
plt.tight_layout()
plt.show()

pivot_table = df.pivot_table(values='Yield %', index='Date', columns='Line')
plt.figure(figsize=(12, 6))
sns.heatmap(pivot_table, annot=True, fmt='.1f', cmap='coolwarm')
plt.title('Heatmap of Daily Yield % by Line')
plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 6))
sns.boxplot(data=df, x='Line', y='Yield %', palette='pastel') data = [[Timestamp('2025-04-01 00:00:00'), 'Line 1', 1075, 98, 90.88]
plt.title('Yield % Distribution per Line')
plt.tight_layout()
plt.show()

totals = df.groupby('Line')['Total Units'].sum()
plt.figure(figsize=(7, 7))
plt.pie(totals, labels=totals.index, autopct='%1.1f%%', startangle=90, colors=sns.color_palette('Accent'))
plt.title('Total Units Contribution by Line')
plt.tight_layout()
plt.show()

def high_yield_alert(dataframe, threshold=96):
    alert_data = dataframe[dataframe['Yield %'] >= threshold]
    return alert_data[['Date', 'Line', 'Yield %']]

alerts = high_yield_alert(df)

print("Dates with high yield (>=96%):")
print(alerts)

line_max = df.groupby('Line')['Yield %'].max().reset_index().rename(columns={'Yield %': 'Max Yield'})
line_min = df.groupby('Line')['Yield %'].min().reset_index().rename(columns={'Yield %': 'Min Yield'})
line_avg = df.groupby('Line')['Yield %'].mean().reset_index().rename(columns={'Yield %': 'Avg Yield'})

line_stats = pd.merge(line_max, line_min, on='Line')
line_stats = pd.merge(line_stats, line_avg, on='Line')

print("\nLine-wise Yield Statistics:")
print(line_stats)

avg_daily_yield = df.groupby('Date')['Yield %'].mean().reset_index()

plt.figure(figsize=(10, 5))
sns.lineplot(data=avg_daily_yield, x='Date', y='Yield %', color='darkgreen', marker='s')
plt.title('Average Yield % Across All Lines')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

low_yield = df[df['Yield %'] < 90]
print("\nLow Yield Instances (<90%):")
print(low_yield[['Date', 'Line', 'Yield %']])

defect_trend = df.groupby(['Date'])['Defective Units'].sum().reset_index()

plt.figure(figsize=(10, 6))
sns.lineplot(data=defect_trend, x='Date', y='Defective Units', color='crimson', linewidth=2.5)
plt.title('Total Defects per Day')
plt.tight_layout()
plt.show()

line_variance = df.groupby('Line')['Yield %'].var().reset_index().rename(columns={'Yield %': 'Yield Variance'})
print("\nLine-wise Yield Variance:")
print(line_variance)

df['Week'] = df['Date'].dt.isocalendar().week
weekly_avg = df.groupby(['Week', 'Line'])['Yield %'].mean().reset_index()
```

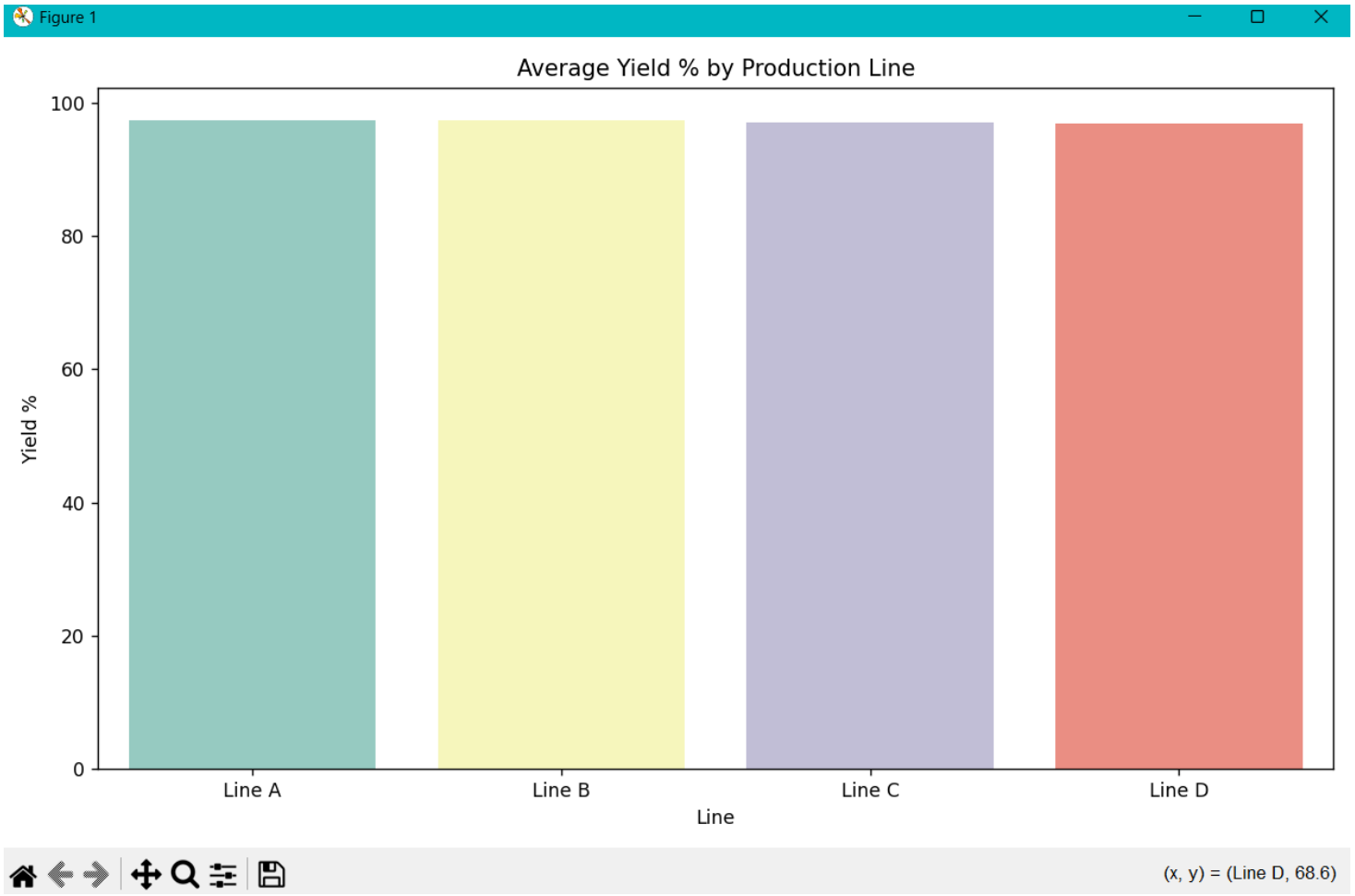
(variable) line_min: DataFrame

```python
print(line_variance)

df['Week'] = df['Date'].dt.isocalendar().week
weekly_avg = df.groupby(['Week', 'Line'])['Yield %'].mean().reset_index()

plt.figure(figsize=(10, 6))
sns.lineplot(data=weekly_avg, x='Week', y='Yield %', hue='Line', style='Line', markers=True)
plt.title('Weekly Yield % by Line')
plt.tight_layout()
plt.show()

defective_summary = df[df['Defective Units'] > 30]
print("\nDays with High Defective Units (>30):")
print(defective_summary[['Date', 'Line', 'Defective Units']])
```



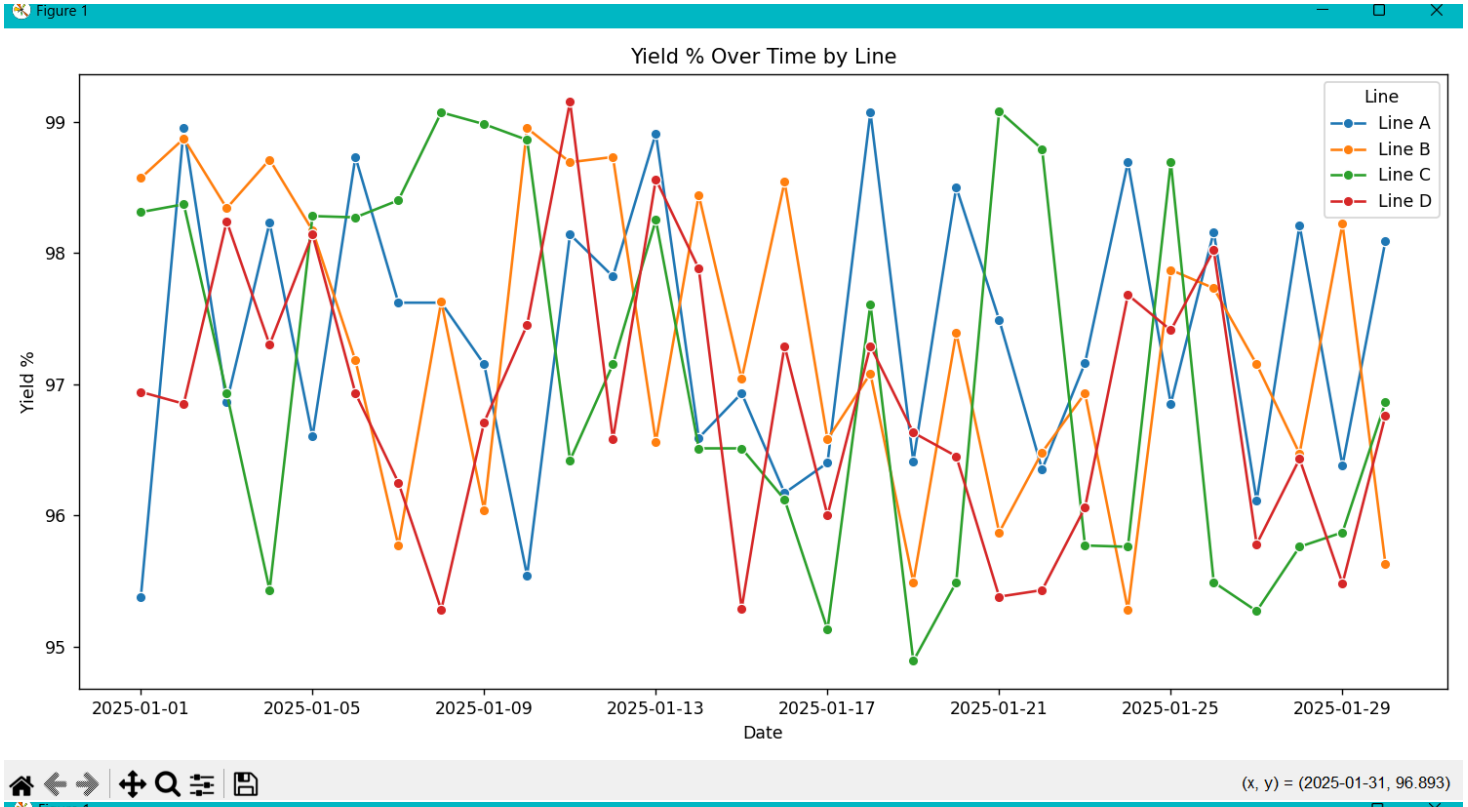


Figure 1
Yield % Over Time by Line
Line
Line A
Line B
Line C
Line D
Yield %
99
98
97
96
95
2025-01-01
2025-01-05
2025-01-09
2025-01-13
2025-01-17
2025-01-21
2025-01-25
2025-01-29
Date
(x, y) = (2025-01-31, 96.893)

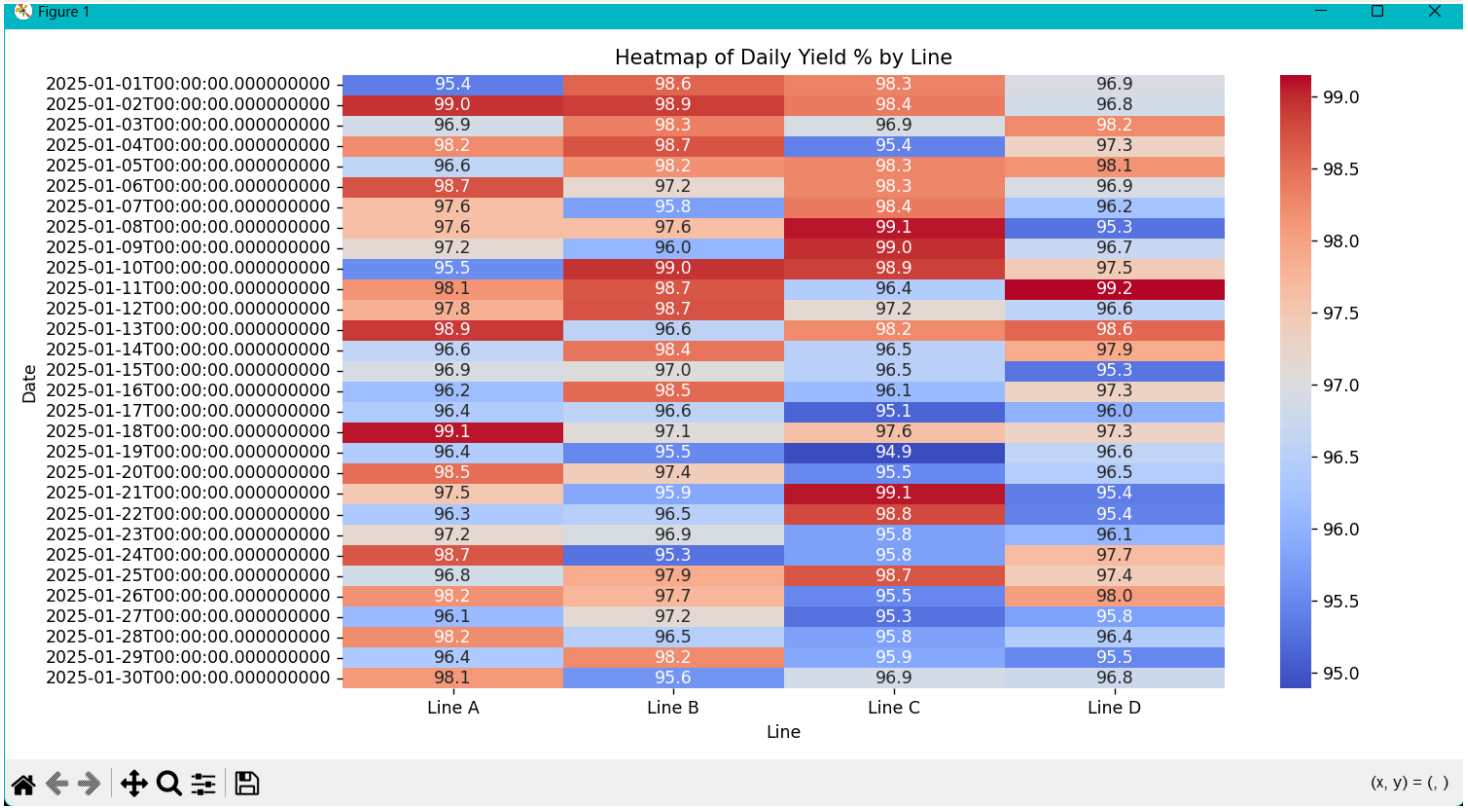


Figure 1
Heatmap of Daily Yield % by Line
Date
2025-01-01T00:00:00.000000000 95.4 98.6 98.3 96.9
2025-01-02T00:00:00.000000000 99.0 98.9 98.4 96.8
2025-01-03T00:00:00.000000000 96.9 98.3 96.9 98.2
2025-01-04T00:00:00.000000000 98.2 98.7 95.4 97.3
2025-01-05T00:00:00.000000000 96.6 98.2 98.3 98.1
2025-01-06T00:00:00.000000000 98.7 97.2 98.3 96.9
2025-01-07T00:00:00.000000000 97.6 95.8 98.4 96.2
2025-01-08T00:00:00.000000000 97.6 97.6 99.1 95.3
2025-01-09T00:00:00.000000000 97.2 96.0 99.0 96.7
2025-01-10T00:00:00.000000000 95.5 99.0 98.9 97.5
2025-01-11T00:00:00.000000000 98.1 98.7 96.4 99.2
2025-01-12T00:00:00.000000000 97.8 98.7 97.2 96.6
2025-01-13T00:00:00.000000000 98.9 96.6 98.2 98.6
2025-01-14T00:00:00.000000000 96.6 98.4 96.5 97.9
2025-01-15T00:00:00.000000000 96.9 97.0 96.5 95.3
2025-01-16T00:00:00.000000000 96.2 98.5 96.1 97.3
2025-01-17T00:00:00.000000000 96.4 96.6 95.1 96.0
2025-01-18T00:00:00.000000000 99.1 97.1 97.6 97.3
2025-01-19T00:00:00.000000000 96.4 95.5 94.9 96.6
2025-01-20T00:00:00.000000000 98.5 97.4 95.5 96.5
2025-01-21T00:00:00.000000000 97.5 95.9 99.1 95.4
2025-01-22T00:00:00.000000000 96.3 96.5 98.8 95.4
2025-01-23T00:00:00.000000000 97.2 96.9 95.8 96.1
2025-01-24T00:00:00.000000000 98.7 95.3 95.8 97.7
2025-01-25T00:00:00.000000000 96.8 97.9 98.7 97.4
2025-01-26T00:00:00.000000000 98.2 97.7 95.5 98.0
2025-01-27T00:00:00.000000000 96.1 97.2 95.3 95.8
2025-01-28T00:00:00.000000000 98.2 96.5 95.8 96.4
2025-01-29T00:00:00.000000000 96.4 98.2 95.9 95.5
2025-01-30T00:00:00.000000000 98.1 95.6 96.9 96.8
Line A
Line B
Line C
Line D
Line
99.0
98.5
98.0
97.5
97.0
96.5
96.0
95.5
95.0
(x, y) = (,)

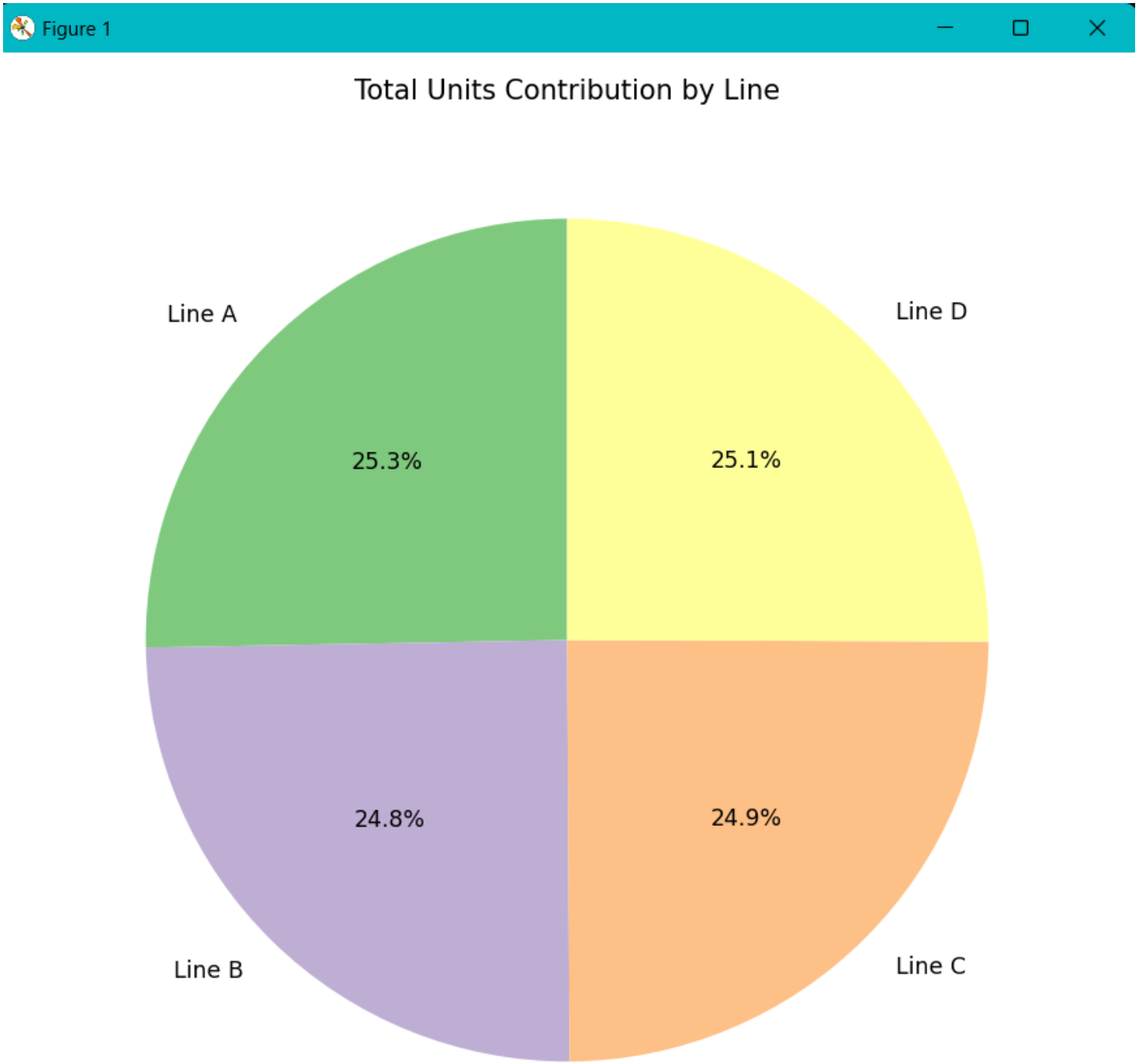


Figure 1
Total Units Contribution by Line
Line A
25.3%
Line D
25.1%
24.8%
Line B
24.9%
Line C