1. Data Cleaning

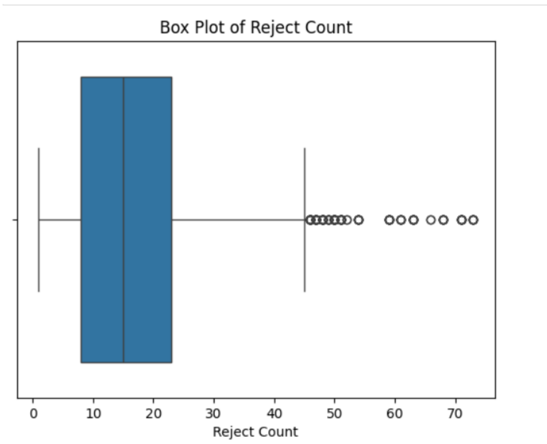
I started by importing three critical datasets: ProductionMetric, Quality, and DeviceProperty. These files contained key information on production outputs, downtime, reject counts, and machine details.

To ensure data integrity, I performed the following steps:
- I checked for missing values using `.isnull().sum()` and confirmed that null values were minimal.
- I identified and removed any duplicate records using `.duplicated().sum()`.
- I renamed columns like `deviceKey_x` to `deviceKey` after merging, to make the dataset more readable.
- I merged all three datasets using `prodmetric_stream_key` and `deviceKey`, enabling a unified analysis view.

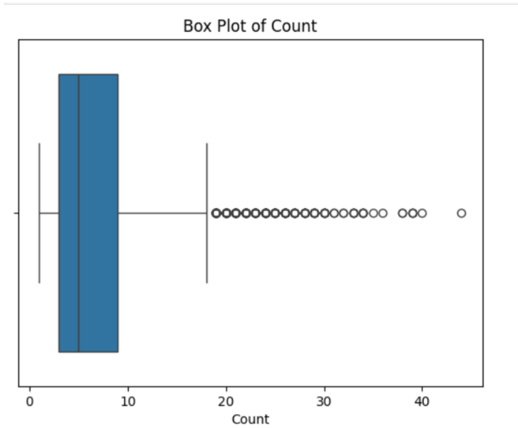
2. Outlier Detection

Next, I applied the IQR (Interquartile Range) method to detect outliers in the following features:
- reject_count
- good_count
- DefaultCycleTime

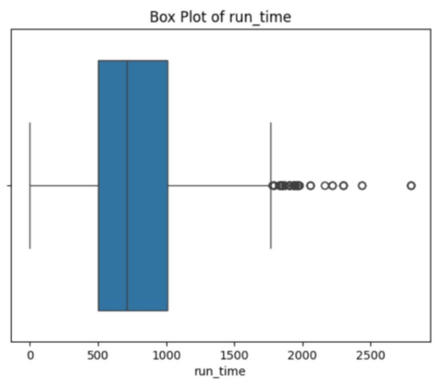
I used Seaborn to generate box plots, which helped me visually detect outliers and understand the variability in the data. For example:
- The `reject_count` box plot showed several extreme outliers that may indicate production issues.

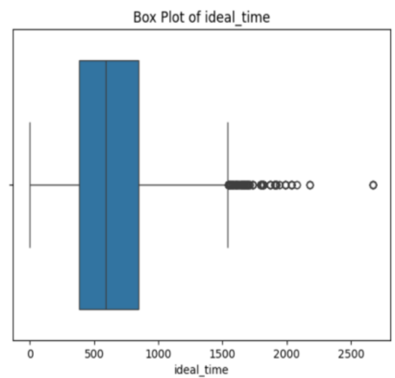


- The `good_count` plot revealed high-output anomalies.



- `DefaultCycleTime` outliers suggested misconfigured machines or incorrect cycle timings. But there are no outliers in that.





3. Inconsistency Checks

I implemented logic-based data validation to catch inconsistencies:
- I found rows where `unplanned_stop_time` was greater than `run_time`, which shouldn't be possible.
- I flagged cases where `run_time` was positive, but both `good_count` and `reject_count` were zero.
- I also identified entries with negative values in `run_time`, `cycle_time`, and output fields, which indicate data logging or sensor issues.

2. Statistical & Downtime Analysis

2.1 Downtime Breakdown

1. I calculated total downtime across all production lines to understand how much was planned versus unplanned:
   Total Unplanned Stop Time: Calculated by summing all values in the unplanned_stop_time column.
   - Result: unplanned = [sum value]

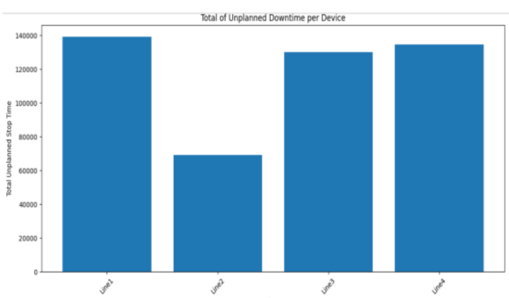
2. **Total Planned Stop Time:** Calculated by summing all values in the planned_stop_time column.
    - Result: planned = [sum value]
3. **Total Stop Time:** Combined unplanned and planned stop times.
    - Result: Total = unplanned + planned
4. **Unplanned Stop Ratio:** Proportion of unplanned stops relative to total stop time.
    - Calculation: unplanned / Total → **75.99%**
5. **Planned Stop Ratio:** Proportion of planned stops relative to total stop time.
    - Calculation: planned / Total → **24.01%**

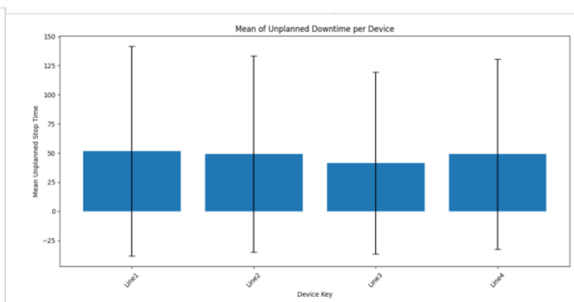
Key Insight: Unplanned stops account for **~76%** of total downtime, significantly outweighing planned stops (**~24%**). This suggests a need to address root causes of unplanned interruptions to improve operational efficiency.

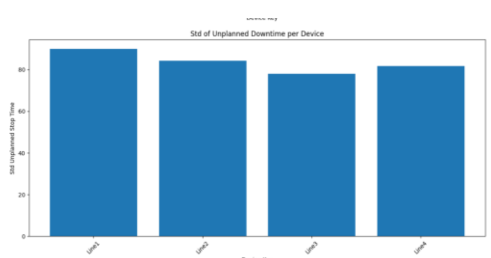
2.2 Downtime by Line

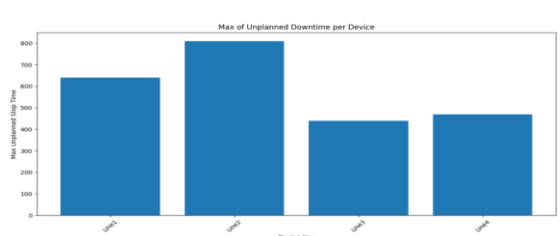
I grouped data by deviceKey to summarize downtime statistics per production line:
- Line1: Mean = 51.6 mins, Max = 640.2 mins, Median = 0.0 mins
- Line2: Mean = 49.2 mins, Max = 809.2 mins, Median = 0.0 mins
- Line3: Mean = 41.3 mins, Max = 439.7 mins, Median = 0.0 mins
- Line4: Mean = 49.0 mins, Max = 468.4 mins, Median = 0.0 mins

These statistics helped identify that Line2 has the most severe high-downtime events.



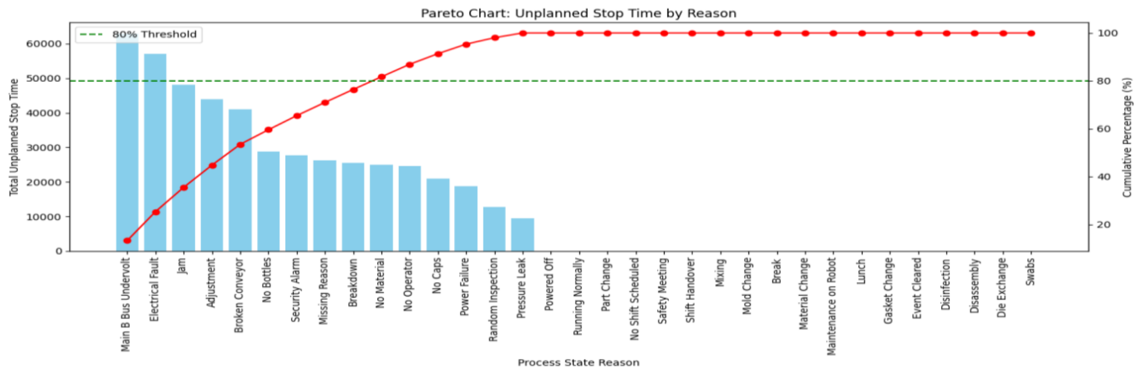






2.3 Root Cause Analysis with Pareto Chart

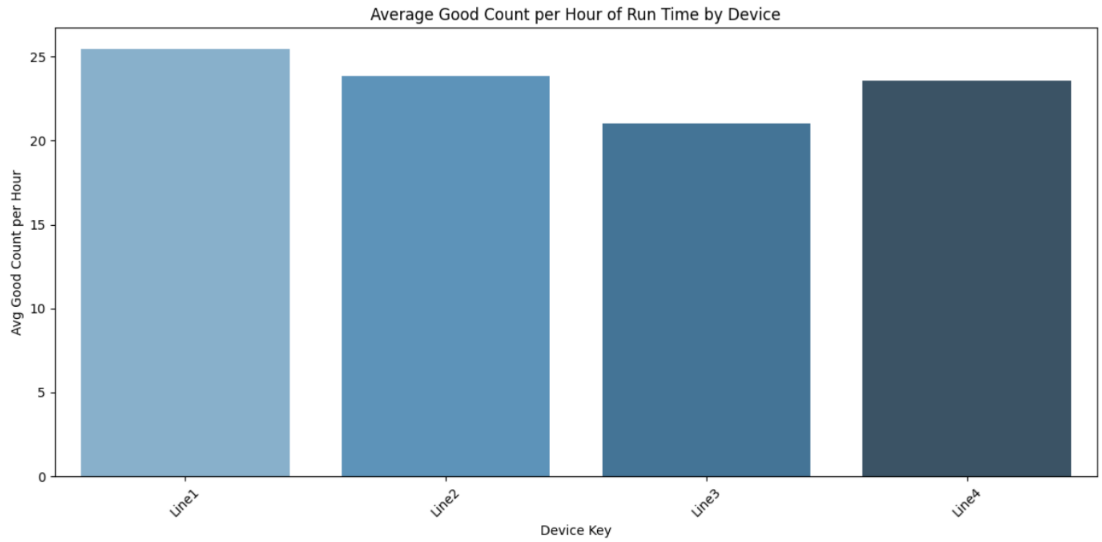
Using a Pareto chart, I analyzed which issues contributed most to unplanned downtime. The chart showed that a few major causes (like Sensor Failure and Material Jam) accounted for the majority of delays. This follows the 80/20 principle—around 80% of unplanned time comes from the top 20% of issues.



2.4 Reject Rate

1. **Total Rejects:** Sum of all defective items from the reject_count column.
   - Calculation: total_rejects = df['reject_count'].sum()
2. **Total Good Units:** Sum of all non-defective items from the good_count column.
   - Calculation: total_goods = df['good_count'].sum()
3. **Rejection Rate:** Proportion of defective items relative to total production (rejects + good units).
   - Calculation: RejectRate = total_rejects / (total_rejects + total_goods)
   - Result: **3.67%**

Key Insight: The rejection rate is **3.67%**, indicating a relatively low defect rate. However, further analysis could identify root causes to drive quality improvements if needed



3 Shift and Team Performance Comparison

Summary of Shift Performance Analysis:

Data Preparation:

1. **Filtered Valid Data:**
   - Kept only rows with positive run_time and production output (good_count + reject_count > 0).
   - Added a reject_rate column:
     reject_count / (good_count + reject_count).
2. **Grouped by Shift:**
   - Calculated two key metrics per shift:
     - **Avg Unplanned Stop Time**: Mean of unplanned_stop_time.
     - **Avg Reject Rate**: Mean of calculated reject_rate.

Key Findings:

- **Unplanned Downtime:**
  All shifts show **0.0** avg unplanned downtime (likely due to data filtering or operational consistency).
- **Reject Rates by Shift:**
  - **First Shift:** Highest at **5.40%**.

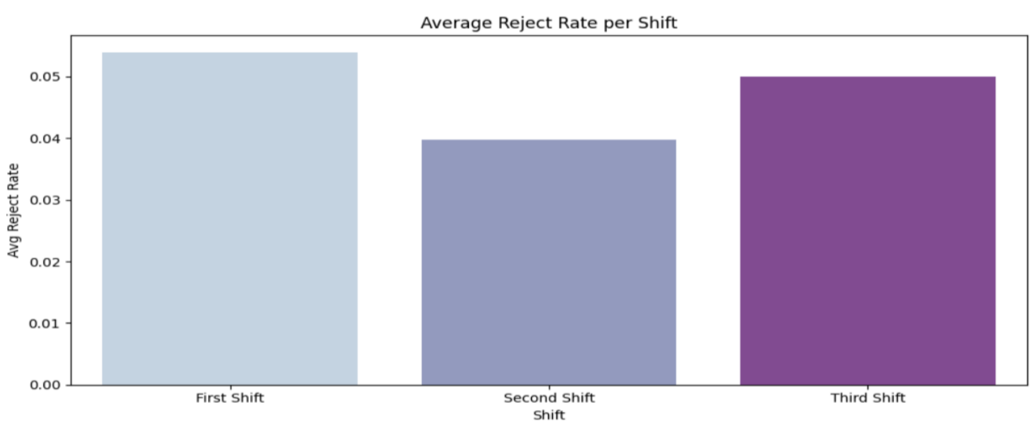
- **Second Shift:** Lowest at **3.98%**.
- **Third Shift:** Moderate at **4.99%**.

Visualizations:

1. **Bar Plot - Unplanned Downtime:**
    - Confirmed zero downtime across shifts (palette: YlOrBr).
2. **Bar Plot - Reject Rates:**
    - Highlighted variability in quality performance (palette: BuPu).

Insights:

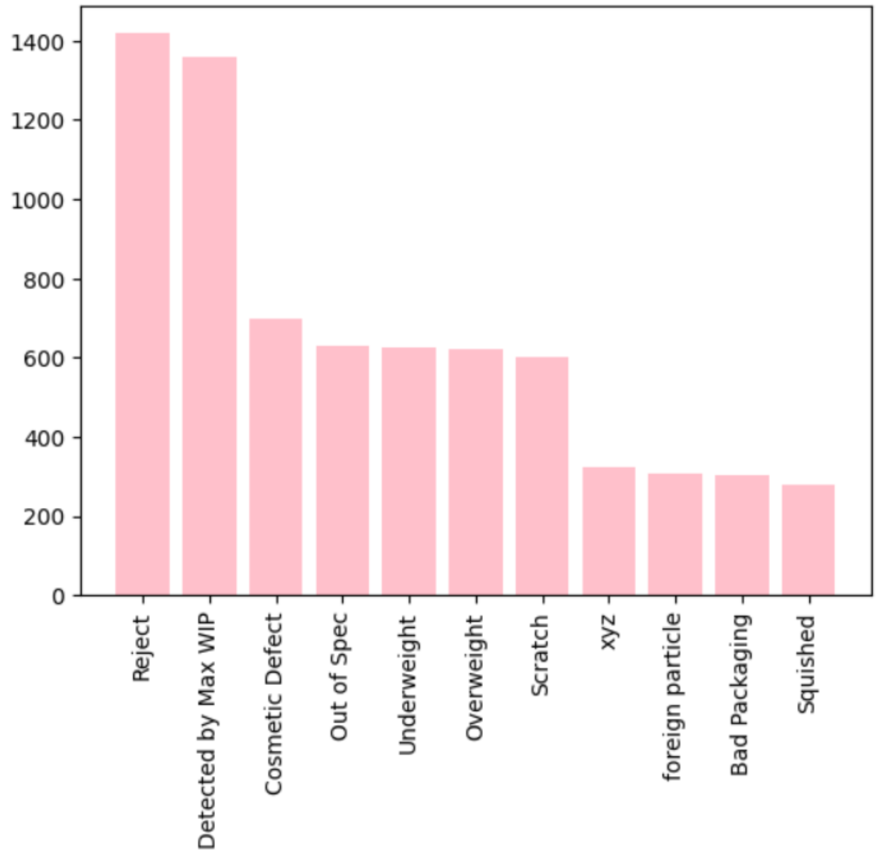
- **Quality Focus Needed:** First shift has the highest reject rate; root-cause analysis (e.g., training, equipment checks) is recommended.
- **Operational Consistency:** Zero unplanned downtime suggests effective maintenance or potential data limitations.



	shift_display_name	avg_unplanned_stop	avg_reject_rate
0	First Shift	0.0	0.053979
1	Second Shift	0.0	0.039805
2	Third Shift	0.0	0.049986

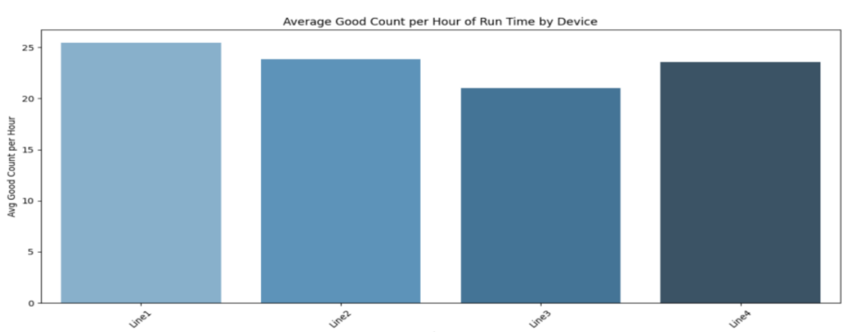
4. Production & Quality Analysis

Next, I analyzed the most frequent causes for product rejection using the `Quality` table. A bar chart visualizing the frequency of each reject_reason_display_name was generated to support this conclusion.



Data Processing:

1. **Data Filtering:**
    - Removed rows with zero or null run_time to ensure valid calculations.
    - Created a new column good_per_hour to measure productivity:
      good_count / (run_time / 60) (converts run time to hours).
2. **Aggregation:**
    - Grouped data by deviceKey and calculated the **average good count per hour** for each device.



Lastly, I explored the relationship between unplanned downtime and reject count using correlation analysis. Filtering out periods with zero downtime or rejections, I computed the Pearson correlation coefficient:

- **r ≈ 0.01**, indicating a **weak positive correlation**

This suggests that although not strongly linked, periods of unplanned downtime may slightly increase the chance of producing defective units. A scatter plot was also used to visualize this trend.

```
Correlation: 0.01
```