Statistical Process Control for Manufacturing Parts

Executive Summary

This project implements **Statistical Process Control (SPC)** to monitor and improve a manufacturing process. Using SQL window functions, we calculated moving averages, standard deviations, and dynamic **Upper Control Limits (UCL)** / **Lower Control Limits (LCL)** to determine whether part measurements (height) remain within acceptable quality thresholds.

By identifying **out-of-control points** in real time, the approach provides actionable data to operators, improving both efficiency and product quality.

Project Objectives

- Monitor product **height** measurements for deviations beyond control limits.
- Implement SPC methodology directly in SQL for operational transparency.
- Identify specific operators and items requiring attention.
- Build a scalable method for continuous process monitoring.

Technologies & Tools Used

- **Database:** PostgreSQL
- **Analysis Environment:** Jupyter Notebook (DataCamp DataLab)
- **SQL Techniques:**
  - Window functions (`ROW_NUMBER()`, `AVG()`, `STDDEV()`)
  - Moving window calculations for rolling statistics
  - Nested queries for UCL/LCL computation
  - Conditional flagging using `CASE WHEN`

Dataset Description

Table: `manufacturing_parts`

Column	Description
`item_no`	Unique product identifier
`length`	Length of manufactured item
`width`	Width of manufactured item
`height`	Height of manufactured item (QC parameter)
`operator`	Operator/machine producing the item

SPC Methodology

Control Limit Formulas:

$$UCL = \bar{X} + 3 \times \frac{\sigma}{\sqrt{n}}$$

$$LCL = \bar{X} - 3 \times \frac{\sigma}{\sqrt{n}}$$

Where:

- $ \bar{X} $ is the moving average height
- $ \sigma $ is the rolling standard deviation
- $ n $ = window size (5 consecutive items per operator)

SQL Implementation

We used a **5-item rolling window per operator**:

```
SELECT
 b.operator,
 b.row_number,
 b.height,
```

```
 b.avg_height,
 b.stddev_height,
 b.ucl,
 b.lcl,
 CASE
   WHEN b.height NOT BETWEEN b.lcl AND b.ucl THEN TRUE
   ELSE FALSE
 END AS alert
FROM (
 SELECT
   a.operator,
   a.row_number,
   a.height,
   a.avg_height,
   a.stddev_height,
   a.avg_height + 3 * a.stddev_height / SQRT(5) AS ucl,
   a.avg_height - 3 * a.stddev_height / SQRT(5) AS lcl
 FROM (
   SELECT
     operator,
     ROW_NUMBER() OVER w AS row_number,
     height,
     AVG(height) OVER w AS avg_height,
     STDDEV(height) OVER w AS stddev_height
   FROM manufacturing_parts
   WINDOW w AS (
     PARTITION BY operator
     ORDER BY item_no
     ROWS BETWEEN 4 PRECEDING AND CURRENT ROW
   )
 ) AS a
 WHERE a.row_number >= 5
) AS b
ORDER BY b.row_number;
```

Results & Key Findings

- Total records analyzed: **420 production points**
- UCL & LCL varied per operator based on their historical output.
- **"Alert"** flags signal measurements outside the acceptable range.
- Example flagged cases:
  - **Operator Op-5**, item 5: Height `22.17` — above UCL
  - **Operator Op-14**, item 34: Height `19.99` — below LCL

Sample Output:

Operator	Row No.	Height	Avg Height	UCL	LCL	Alert
Op-5	5	22.17	20.43	21.98	18.88	☑
Op-6	5	20.79	20.42	20.86	19.98	✕
Op-14	34	19.99	20.86	21.68	20.03	☑

Impact

- **Quality Control:** Enables faster detection of machine/operator deviations.
- **Efficiency:** Reduces waste by catching errors before producing large off-spec batches.
- **Scalability:** Method works for any continuous measurement (length, width, weight, etc.).

Challenges & Solutions

Challenge	Solution
Operators had different baseline measurements	Used **per-operator rolling windows** instead of global stats
Dynamic process control calculation in SQL	Applied `ROWS BETWEEN` for moving aggregation
Need to detect deviations in real time	Built query that can be integrated into live dashboards

Conclusion

This SPC-based SQL system effectively monitors manufacturing processes by alerting when part measurements fall outside control limits. The approach minimizes defects, reduces downtime, and boosts product consistency.

It demonstrates both **advanced SQL analytical skills** and **real-world manufacturing data application** — key for any data analyst or process engineer role.

Next Steps

- Automate alerts through dashboard integration (e.g., Power BI, Looker Studio).
- Extend monitoring to **all part dimensions** (length + width).
- Incorporate root-cause analysis for flagged events.
- Apply predictive models to anticipate process drift.