# **Evaluate a Manufacturing Process**

This project aims to improve manufacturing quality by applying Statistical Process Control (SPC). Using SQL, we analyze historical production data to calculate the Upper Control Limit (UCL) and Lower Control Limit (LCL) for product height. These limits define the acceptable range for production. Parts falling outside this range signal process issues that need correction. This data-driven approach helps maintain consistent product quality and ensures the manufacturing process runs efficiently with minimal defects.

UCL = avg\_height + 3 × (stddev\_height / √5)

LCL = avg\_height − 3 × (stddev\_height / √5)

The UCL defines the highest acceptable height for the parts, while the LCL defines the lowest acceptable height for the parts. Ideally, parts should fall between the two limits.

### **Data Sources**

The data is available in the **manufacturing\_parts table** which has the following fields:

* **item\_no:** the item number
* **length:** the length of the item made
* **width:** the width of the item made
* **height:** the height of the item made
* **operator:** the operating machine

**Solution – Using Common Table Expression (CTE)**

WITH f AS(

SELECT

operator,

ROW\_NUMBER() OVER (PARTITION BY operator ORDER BY item\_no) AS row\_number,

height,

AVG(height) OVER(PARTITION BY operator) AS avg\_height,

STDDEV(height) OVER(PARTITION BY operator) AS stddev\_height,

AVG(height) OVER (PARTITION BY operator ORDER BY item\_no ROWS BETWEEN 4 PRECEDING AND CURRENT ROW) + 3 \* (STDDEV(height) OVER (PARTITION BY operator ORDER BY item\_no ROWS BETWEEN 4 PRECEDING AND CURRENT ROW) /SQRT(5)) AS ucl,

AVG(height) OVER (PARTITION BY operator ORDER BY item\_no ROWS BETWEEN 4 PRECEDING AND CURRENT ROW) - 3 \* (STDDEV(height) OVER (PARTITION BY operator ORDER BY item\_no ROWS BETWEEN 4 PRECEDING AND CURRENT ROW)/SQRT(5)) AS lcl,

CASE

WHEN height > (AVG(height) OVER (PARTITION BY operator ORDER BY item\_no ROWS BETWEEN 4 PRECEDING AND CURRENT ROW) + 3 \* (STDDEV(height) OVER (PARTITION BY operator ORDER BY item\_no ROWS BETWEEN 4 PRECEDING AND CURRENT ROW)/SQRT(5)))

OR height < (AVG(height) OVER (PARTITION BY operator ORDER BY item\_no ROWS BETWEEN 4 PRECEDING AND CURRENT ROW) - 3 \* (STDDEV(height) OVER (PARTITION BY operator ORDER BY item\_no ROWS BETWEEN 4 PRECEDING AND CURRENT ROW)/SQRT(5)))

THEN TRUE

ELSE FALSE

END AS alert

FROM manufacturing\_parts

)

SELECT

operator,

row\_number,

height,

avg\_height,

stddev\_height,

ucl,

lcl,

alert

FROM f

WHERE row\_number >= 5

ORDER BY row\_number

**Alternative - Cleaner with less code. Using more subqueries.**

-- Flag whether the height of a product is within the control limits

SELECT

b.\*,

CASE

WHEN

b.height NOT BETWEEN b.lcl AND b.ucl (instead of lengthy

THEN TRUE condition. Need to

ELSE FALSE separate with b subquery.)

END as alert

FROM (

SELECT

a.\*,

a.avg\_height + 3\*a.stddev\_height/SQRT(5) AS ucl, (using alias a)

a.avg\_height - 3\*a.stddev\_height/SQRT(5) AS lcl

FROM (

SELECT

operator,

ROW\_NUMBER() OVER w AS row\_number, (using w as moving

height, window CTE. Since row,

AVG(height) OVER w AS avg\_height, height and std all use the

STDDEV(height) OVER w AS stddev\_height same window conditions,

FROM manufacturing\_parts can use shortcut CTE)

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;