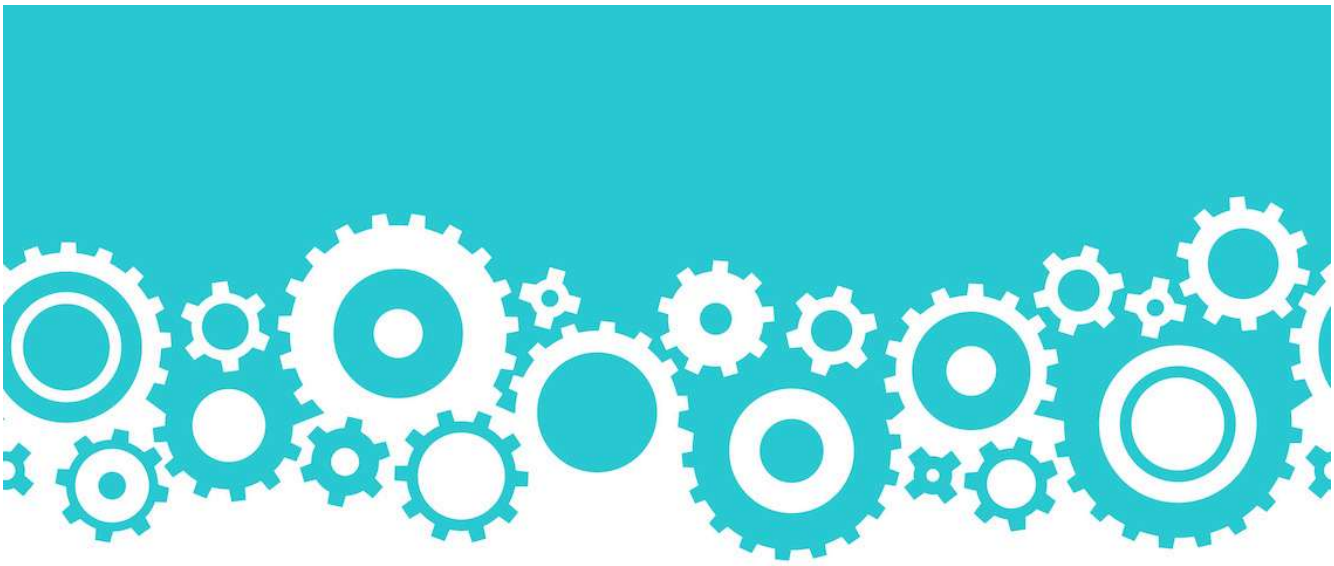

Manufacturing processes for any product is like putting together a puzzle. Products are pieced together step by step, and keeping a close eye on the process is important.

For this project, you're supporting a team that wants to improve how they monitor and control a manufacturing process. The goal is to implement a more methodical approach known as statistical process control (SPC). SPC is an established strategy that uses data to determine whether the process works well. Processes are only adjusted if measurements fall outside of an acceptable range.

This acceptable range is defined by an upper control limit (UCL) and a lower control limit (LCL), the formulas for which are:

$$ucl = avg\_height + 3 * \frac{stddev\_height}{\sqrt{5}}$$

$$lcl = avg\_height - 3 * \frac{stddev\_height}{\sqrt{5}}$$

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.

Using SQL window functions and nested queries, you'll analyze historical manufacturing data to define this acceptable range and identify any points in the process that fall outside of the range and therefore require adjustments. This will ensure a smooth running manufacturing process consistently making high-quality products.

The data

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

Projects Data DataFrame as `alerts`

```sql
-- Final Solution
--- Step 1: Create sub query for creating row number column that contains iteration per calculation
WITH ranked AS
(
SELECT
    operator,
    item_no,
    height,
    ROW_NUMBER() OVER(PARTITION BY operator ORDER BY item_no) AS row_number
FROM public.manufacturing_parts
),
--- Step 2: Create another sub query for creating summary statistics such as average, standard deviation and also counter to limit window function to 5 rows.
--- Actually, COUNT here is optional as it can be also replace with 5 in the calculation of UCL and LCL.
stats AS
(
    SELECT *,
    AVG(height) OVER ( PARTITION BY operator ORDER BY item_no ROWS BETWEEN 4 PRECEDING AND  CURRENT ROW) AS avg_height,
    STDDEV(height) OVER  (PARTITION BY operator ORDER BY item_no ROWS BETWEEN 4 PRECEDING AND  CURRENT ROW) AS stddev_height,
    COUNT(height) OVER (PARTITION BY operator ORDER BY item_no ROWS BETWEEN 4 PRECEDING AND  CURRENT ROW) AS n_in_window_5
    FROM ranked
)

SELECT
    operator,
    row_number,
    height,
    avg_height,
    stddev_height,
    avg_height + (3*(stddev_height)/SQRT(5)) AS ucl,
    avg_height - (3*(stddev_height)/SQRT(5)) AS lcl,
    CASE
    WHEN height > avg_height + (3*(stddev_height)/SQRT(n_in_window_5))  THEN TRUE
    WHEN height < avg_height - (3*(stddev_height)/SQRT(n_in_window_5))  THEN TRUE
    ELSE FALSE END AS alert

FROM stats
WHERE stats.n_in_window_5 = 5
```

i...	ope...	row_nu...	h...	avg_hei...	stddev_height	ucl	lcl	al...
0	Op-1	5	19.46	19.778	1.062812307	21.2039123395	18.3520876605	False
1	Op-1	6	20.36	19.912	1.0908116244	21.3754773657	18.4485226343	False
2	Op-1	7	20.22	20.03	1.084573649	21.4851082434	18.5748917566	False
3	Op-1	8	21.03	19.934	0.9312249997	21.183369441	18.684630559	False
4	Op-1	9	19.78	20.17	0.5988321969	20.9734176996	19.3665823004	False
5	Op-1	10	20.71	20.42	0.4768123321	21.0597108722	19.7802891278	False
6	Op-1	11	20.62	20.472	0.4827732387	21.1197082677	19.8242917323	False
7	Op-1	12	19.51	20.33	0.6506535176	21.2029432971	19.4570567029	False
8	Op-1	13	20.06	20.136	0.5215649528	20.8357528135	19.4362471865	False
9	Op-1	14	20.3	20.24	0.4832701108	20.8883748916	19.5916251084	False
10	Op-1	15	20.25	20.148	0.4095973633	20.6975325286	19.5984674714	False
11	Op-1	16	20.52	20.128	0.3823218539	20.6409385928	19.6150614072	False
12	Op-1	17	19.33	20.092	0.4563660811	20.704279348	19.479720652	True
13	Op-1	18	19.12	19.904	0.6324792487	20.7525599566	19.0554400434	False
14	Op-1	19	19.37	19.718	0.6235944195	20.5546397074	18.8813602926	False
15	Op-1	20	18.8	19.428	0.6508993778	20.3012731531	18.5547268469	False

Rows: 420 Expand

4 hidden cells