THOMAS EAPEN · SELECTED WORK

Search for matches, then count once

Separate finding relevant records from deciding what constitutes one count.

Synthetic example · independently invented data

One-page print edition · A3 landscape

01 Inspect the inputs

02 Apply the method

03 Check the result

INVENTED INPUTS

An event log with a repeat

Case	Stage
Case A	Intake
Case A	Review
Case A	Review
Case B	Intake
Case B	Specialist
Case C	Review
Case C	Specialist
Case D	Intake

Eight invented event rows

Case A has two Review rows. Case C reaches both selected stages. The target stages are Review and Specialist.

SELECTED PYTHON

The working calculation

Pandas is pd. rows is the log at left; targets contains the two selected stage labels.

```
events = pd.DataFrame(rows)
matches = events.loc[events['stage'].isin(targets)]
distinct = matches.drop_duplicates(['case', 'stage'])
counts = distinct.groupby('stage').size().reindex(
    targets, fill_value=0
)
any_match = distinct['case'].nunique()
```

The code shown here is executed to calculate this example's results.

COMPUTED RESULT

Distinct matches

Stage	Cases
Review	2
Specialist	2

Distinct cases reaching each selected stage

3

cases with any target match

5 matching rows → 4 distinct case/stage pairs



Case A counts once for Review. Case C counts once in each stage, but only once in the "any match" total.

What this demonstrates

The counting unit is part of the method. Matching rows, distinct case/stage pairs and distinct cases answer different questions.

Check the boundary

Do not add the two stage counts to obtain the number of cases: a case can appear in both stages. This is a new demonstration of the search/counting idea.

Thomas Eapen
nowlabs.net/tangerine-woa

New illustrative material explaining the approach. Values, configurations and results shown here are not historical workplace records.