One grouping, two measures

Use the same grouping structure to compare related counts without mixing their meaning.

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

Repeated reporting groups

Group	Path	Events	Handoffs
A	Direct	18	2
A	Direct	12	1
A	Assisted	10	4
B	Direct	16	1
B	Assisted	14	3
B	Assisted	6	2

Six synthetic group/path records

Repeated rows for a group and path contribute to the same subtotal. Both measures are additive counts.

SELECTED PYTHON

The working calculation

Pandas is pd. rows contains the six input records; counts are already numeric.

```
data = pd.DataFrame(rows)
summary = data.pivot_table(
    index='group', columns='path',
    values=['events', 'handoffs'],
    aggfunc='sum', fill_value=0
)
```

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

COMPUTED RESULT

Two aligned pivot views

Group	Assisted	Direct
Group A	10	30
Group B	20	16

Event counts by path

Group	Assisted	Direct
Group A	4	3
Group B	5	1

Handoff counts by path

76 / 13

events / handoffs

Totals across both groups and paths

The columns have the same meaning in both views, so the reader can compare their distribution directly.

What this demonstrates

A shared grouping structure makes related measures easier to compare. Naming the aggregation prevents a default mean from changing the question.

Check the boundary

Summing counts is appropriate here. Rates and averages require a different aggregation rule, often using their underlying denominators.

Thomas Eapen
nowlabs.net/tangerine-woa

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