Interaction counts & means

Define the qualifying records, count within each interaction, then make the denominator explicit.

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

Mixed outcomes in one log

Case	Outcome
Case A	Complete
Case A	Unspecified
Case A	Internal
Case B	Complete
Case B	Complete
Case C	Exit
Missing	Complete
Case D	Internal

Eight independently invented event records

Complete, Exit and unspecified outcomes qualify when a case key is present. Internal rows and missing case keys are excluded.

SELECTED PYTHON

The working calculation

Pandas is pd. rows is the log; accepted contains Complete and Exit. Unspecified outcomes are intentionally included.

```
events = pd.DataFrame(rows)
eligible = events.loc[
    events['case'].notna()
    & (events['outcome'].isin(accepted)
       | events['outcome'].isna())
]
counts = eligible.groupby('case').size()
mean_count = counts.mean()
```

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

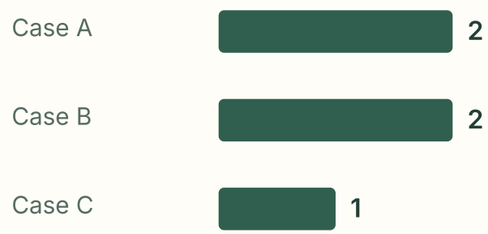
COMPUTED RESULT

A mean with a stated population

1.67

records per qualifying interaction

5 records ÷ 3 interactions



$$5 \div 3 = 1.67$$

3 rows are excluded. Case D has no qualifying records and is not part of the mean's denominator.

What this demonstrates

Filtering changes the population being described. Counting first makes both the numerator and denominator visible before calculating an average.

Check the boundary

This is the mean among interactions with qualifying records, not among every case in the original log. An all-case average would require an explicit zero-count policy.

Thomas Eapen
nowlabs.net/tangerine-woa

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