

Is this difference real? Run it on your own data

Hand this whole document to ChatGPT, Claude or Gemini, along with the notebook from <https://www.stochasticstrata.com/blog/is-this-difference-real>. It tells the assistant what to do; you only answer questions.

For you (the human): read this first

You have a breakdown table: conversion rate by device, or region, or match type, or creative. One row looks better than the others. This works out whether that is a real difference or the ordinary wobble of small numbers, and above all whether it survives the fact that you looked at the whole table before that row caught your eye.

- You need ONE row per thing you are comparing, with two counts on it: how many clicks (or sessions, or impressions) and how many conversions. Raw counts. Not rates.
- Include EVERY row you looked at, not just the interesting ones. This matters more than it sounds and there is a section on it below.
- Nothing is uploaded anywhere. The notebook runs on your own machine, or in your own Google Colab session, and your file stays there.
- If you use Colab, note that your file is then sitting on Google's servers for the length of that session. If that is a problem for your data, run the notebook locally instead.

What you will get back

Three readings of the same table, and they will disagree. That disagreement is the point of the exercise, so do not treat the largest number as the answer.

- The reflex: anything more than a fifth off the average. On simulated accounts with the truth built in, this flagged 18.4 rows per account, of which 87 percent were wrong.
- One proper test per row: 3 rows, 59 percent wrong.
- Corrected for the whole table: 0.5 rows, 12.7 percent wrong, and it finds only 13.8 percent of the differences that are genuinely there. That last figure is the price, and it is not hidden.

Instructions for the AI assistant

You are helping someone who is comfortable with marketing reports and not with code. Do the technical work yourself. Ask for one thing at a time, wait for it, and never present a number without the range or the caveat that belongs to it.

Why the reader needs the notebook: the honest answer, measured

Not because you cannot do the sums. You can, and I checked rather than asserting it. A rebuild from the formulas printed in this document reproduces the notebook's test result on every row of its own forty-segment example to within three parts in ten million, and its range edges to within five hundred-thousandths of a percentage point. Following only the PROSE description of the correction below, a rebuild reproduces all forty of the notebook's verdicts, with zero disagreements. An earlier draft of this page claimed the correction could not be rebuilt from a description. That was measured and it was false, so it is gone.

The reason to run the notebook is what happens where the data cannot carry an answer. A row with no clicks, and a pair where nothing has converted in either row, both divide by zero. A hand calculation crashes there, and a reader will experience that as a broken row and may simply drop it, which quietly removes it from the family and weakens the correction on every row that remains. The notebook declines with a reason instead. It also warns when rows are too thin to judge, which no formula above does, and it carries the bookkeeping across forty rows without a slip.

So: run the notebook. If the reader genuinely cannot, you may compute everything above by hand (it will be right), but you must then do the refusing yourself. Say plainly which rows you could not evaluate and why, and never drop one silently.

Step 1: get one row per thing being compared

Ask what the reader is comparing and where their data lives. The table needs exactly three columns. Column names do not matter; the notebook auto-detects common aliases and you can rename them anyway.

```
segment, trials, successes
mobile, 4192, 88
desktop, 3117, 59
tablet, 412, 7
mobile_app, 208, 3
```

- segment: the thing being compared: a device, a region, a creative, a match type.
- trials: the denominator. Clicks, sessions, impressions, whatever the rate is out of.
- successes: the numerator. Conversions, sign-ups, clicks.

IF THEIR EXPORT HAS A CONVERSION-RATE COLUMN, DO NOT USE IT. Recover the counts instead. A rate has already thrown away the sample size, and the sample size is the only thing the test needs from them. If the counts genuinely cannot be recovered, stop and say so. Do not multiply a rate back out and pretend.

Step 2: the extraction query

For a GA4 export in BigQuery, adapt this. The placeholders in angle brackets are the parts you rewrite for their account: the breakdown column they care about and whatever marks a converting row in their data.

```
-- GA4 has no table of ad clicks: for clicks, use the Google Ads download instead.
-- This counts SESSIONS and the sessions with a key event, per device, from the GA4 export.
WITH s AS (
  SELECT
    device.category AS segment,
    CONCAT(user_pseudo_id, '.', CAST((SELECT value.int_value FROM UNNEST(event_params)
      WHERE key = 'ga_session_id') AS STRING)) AS session_key,
    LOGICAL_OR(event_name = 'purchase') AS converted -- or your key event
  FROM `<YOUR_PROJECT>.analytics-<PROPERTY_ID>.events_*`
  WHERE _TABLE_SUFFIX BETWEEN FORMAT_DATE('%Y%m%d', DATE_SUB(CURRENT_DATE(), INTERVAL 37 DAY))
    AND FORMAT_DATE('%Y%m%d', DATE_SUB(CURRENT_DATE(), INTERVAL 8 DAY))
  GROUP BY segment, session_key
)
SELECT segment, COUNT(*) AS trials, COUNTIF(converted) AS successes
FROM s GROUP BY segment ORDER BY trials DESC
```

For an ad platform export, most report builders will give this directly: pick the breakdown dimension, then clicks and conversions as the two metrics. Do not let them download the rate column instead.

Step 3: the whole-table rule, which is the one people break

The correction needs to know how many rows the reader LOOKED AT, not how many they found interesting. If they scanned forty rows and hand you the three that caught their eye, the correction is told there were three tests and will happily confirm all three. This is the easiest way there is to get a wrong answer out of a correct method.

So ask, in these words: how many rows were in the table when you spotted this? Then make sure that many rows are in the file. If they cannot reconstruct the full table, say the correction cannot be applied honestly and give them the uncorrected reading with that stated.

A related trap: if they scan one breakdown, correct across it, then go and look at a different breakdown, that is a NEW family and the guarantee does not stretch across both. The honest version is to decide the family before looking.

Step 4: run the notebook

Two routes. Google Colab: open colab.research.google.com, File then Upload notebook, then upload their CSV with the folder icon in the left sidebar. Or locally: `python -m pip install jupyter numpy pandas matplotlib`, then jupyter notebook.

There is one cell marked as the place to point at their file. Set it to their filename and run every cell from the top. Do this for them. Do not talk them through editing Python.

Step 5: read the answer honestly

The notebook returns, for every row: its rate, an honest range around that rate, its own test result, and whether it survives the correction across the whole table. Report the range every time you report a rate.

- A row that survives the correction is a finding. Say what it is worth in money or volume as well as whether it is real. A real difference can still be too small to act on.
- A row that passes its own test but not the correction is the interesting case, and the honest sentence is: on its own this would clear the usual bar, but you found it by scanning a table this size, and a table that size hands out gaps like this for free.
- A row that fails both is not evidence the two are the SAME. It is evidence this data cannot tell them apart. Those are very different statements and the second is almost always the true one at marketing volumes.

The formulas, if you need to show your working

The test on two rows: pool both rows' counts as if they were one row, use that pooled rate to work out how much the gap would move on its own, and divide the observed gap by it. The pooling matters: you are testing whether the two are the same, so the wobble is worked out under the assumption that they are. An unpooled version answers a different question.

```
z = (xa/na - xb/nb) / sqrt( p*(1-p) * (1/na + 1/nb) )
where p = (xa + xb) / (na + nb)
and the two-sided answer is 2 * (1 - NORMSDIST(abs(z)))
```

The range on one rate is the score interval, NOT the rate plus or minus a couple of standard errors. The textbook form is unreliable at exactly the counts a breakdown table is full of: measured on marketing-shaped numbers, a range labelled 95 percent actually held the truth 33.5 percent of the time at 20 clicks and 62.8 percent at 50. The score interval managed 93.7 and 91.7 percent at the same counts.

```
centre = (p + z^2/(2n)) / (1 + z^2/n)
half   = z*sqrt( p*(1-p)/n + z^2/(4n^2) ) / (1 + z^2/n)
range  = centre - half to centre + half      with z = 1.959964
```

The correction: sort every row's result smallest to largest. The row at position k out of m must clear k/m times q , rather than q flat, where q is the share of your flags you are willing to have be wrong. The most surprising row faces the harshest bar and the least surprising the gentlest.

The caveats: tell the reader these BEFORE they act

- The correction assumes the tests do not lean on each other, and theirs do. The 1995 result is proved for independent tests; rows of one account share traffic, share a season, and in this leave-one-out form literally share a denominator with the group they are tested against, which ties the rows together in the direction the 2001 companion result does not cover. Unproven rather than shown wrong. Treat the corrected count as a large improvement over no correction, not as a guarantee with a proof behind it.
- Real rows wobble more than coin flips. The test treats every click as an independent draw at one rate; on real accounts a segment's rate moves week to week beyond that, and on a real store a screen like this flagged a no-difference row 5.5 times as often as the label promises in a typical month. Have them split the counts by week and check the gap holds in most weeks before acting.
- The test's answer is not the probability they are wrong. It is the probability of a gap this large IF there were

no difference. It says nothing about how likely a difference was before they looked, and no arithmetic fixes that.

- Not significant does not mean the same. It means this data cannot tell them apart.
- Significant does not mean worth acting on. Pile up enough clicks and a completely irrelevant difference clears any threshold. Always read the size of the gap and its range beside the test result.
- Checking daily and calling it when it crosses the line is a different procedure from testing once at a planned volume, and it produces false findings far more often than the label says. All of this assumes they looked once.
- The ratio test is for cost-per-something only. On a cost per conversion it cried wolf 4.6 percent of the time, which is what it aims at. On return on ad spend, where the top half is a sum of order values that differ from each other, it cried wolf 16.8 percent of the time. Do not point it at revenue.
- A row with no conversions yet has not been shown to be bad. Out of 40 clicks with nothing converted, the honest range still runs up to 8.76 percent, a perfectly healthy rate. The textbook range says zero to zero, which is certainty from no evidence.
- Their ad platform may show a row as significant that this does not. The platforms test one comparison at a time with no correction; after the correction a row whose own p-value clears 0.05 can stay unflagged on purpose - held to a stricter standard, not hidden. If one row mattered before they looked, test it on its own.
- A window that spans a change blends two rates. A promotion, a tracking break or a new bid strategy part-way through means the pooled rate describes neither side. Cut the window at the change.

The single most useful thing you can tell them

Decide the comparison before looking at it. One planned comparison at decent volume needs no correction and carries none of this arithmetic. It is the scanning that costs, and the correction is what you pay once you have already scanned. On a table where nothing whatsoever is happening, scanning 40 rows hands back 1.87 findings that are not there, and 86.0 percent of accounts produce at least one. At eighty rows it is 97.7 percent.

If the notebook refuses

It refuses on purpose in three cases, and each refusal is the right answer rather than a bug. A row with no trials cannot be tested. A pair where both rows are all-success or all-fail has no variance to divide by. And a table with fewer than two rows is not a family, so there is nothing to correct across.

It also warns rather than refuses when rows are thin. Do not talk the reader past that warning. A thin row's range is honestly wide (except that on one or two conversions at a low rate its bottom end sits a little high), and reporting its rate without the range is the failure this whole post exists to prevent.