How many real people did I actually reach across platforms - a brief for your AI assistant

Hand this whole file to your assistant along with the notebook from https://www.stochasticstrata.com/blog/how-many-people-did-i-actually-reach. It tells the assistant what to do, what to refuse, and what to say before anyone reports a cross-platform reach number or cuts a platform for 'reaching the same people'.

For the assistant: what you are being asked to do

The person you are helping ran a campaign on two platforms and wants to know how many distinct people it reached. They may not write code. Do the technical parts yourself and walk them through one step at a time. Ask for one thing, wait, then ask for the next.

THE SCOPE, so you do not overreach: this answers one exposure question - how many distinct people the two platforms reached together, and how many the second added. It gives exact counts for what the site can see, bounds on the total reach rather than a point, and a refusal of the total itself. It is not attribution, not incrementality, and not a cost-per-acquisition method: the cost per new customer beside it is a count the site already has and needs no overlap. Do not carry the reach range into a customer figure; that produced a five-fold range when tried, because it rests on a crediting rule no paper supports.

- Ask for the two reach figures, from each platform's own reporting, for the same window. Both platforms label reach an estimate and one counts accounts, not people; take the figures as given and say so. Each must be the platform's own de-duplicated reach for the whole window at campaign or account level - never add up ad-set or campaign rows, which counts the same people twice.
- The landing log needs GA4 linked to BigQuery BEFORE the window began - the export does not fill in the past. Without it there is no visitor-level file; say so plainly rather than building one from grouped counts.
- Find out how their site recognises a returning visitor - a cookie, or a login/email. A cookie splits one person into several visitors across devices and in-app browsers, which makes every shared count a floor. A cookie written by a script in the page is the fragile case - in-app browsers, second devices and cleared browsers all split it - so a first-party cookie set by the site's own server (server-side tagging) or a login is what makes the shared count worth reading. Say this before the numbers, not after.
- Get the visitor file or the landing log (the shapes are below), map their columns onto the ones the notebook expects, set the file path and the reach figures in the marked cell, and run it for them if you can.
- Ask, optionally, for the population both platforms draw from (their audience-size estimates are the nearest thing), the spend on the second platform, and their consent rate if they are in an opt-in region.
- Ask whether both platforms report reach by a segment the site also sees - device class, region, age band. If so, the file carries a 'segment' column and the reach figures are typed in per segment as well; the notebook then computes the stratified floor, the one narrowing the reports allow.
- Ask whether the site knows which visitors became customers in the window. If it does, the file carries a 'converted' column and the notebook reports the customers each platform brought alone - a count, with no overlap in it. It also prints the site's own spend-over-customers arithmetic beside that count; say plainly that this is the site's number, not the method's, and that whether those customers would have bought anyway is a holdout's question.
- Report the exact part first - distinct visitors, the visitors and customers each platform brought alone - then the reach range, then the refusal. Never report a single 'unique reach' number; the reason is measured, and it is in the caveats.

The two files

THE VISITOR FILE: one row per visitor per platform they arrived from - 'visitor_id' (the identity the site recognises a returning visitor by) and 'platform' (meta, tiktok, google, ...). A visitor who came from both platforms has two rows. A 'platforms' column holding 'meta|tiktok' on one row per visitor is accepted too.

Optionally a 'converted' column: 1 if that visitor became a customer in the window, else 0.

THE LANDING LOG: one row per paid arrival - 'timestamp', 'visitor_id' and 'landing_url' with the click tags still in it (fbclid, ttclid, gclid, gbraid, wbraid, msclkid, li_fat_id) or a 'source' column naming the platform, and optionally the same 'converted' flag on any of a visitor's rows. The notebook consolidates it into the visitor file and reports the arrivals it could not place.

Query to adapt

```
-- One row per paid arrival: when, the site's pseudonymous visitor id, the landing URL with its
-- click tags, the collected source, and whether that visitor bought in the window. From a GA4
-- export in BigQuery. The notebook reads the platform off the URL (fbclid, ttclid, gclid, gbraid,
-- wbraid, msclkid, li_fat_id) or off the source, and the customers off the converted flag.
-- user_pseudo_id is a browser or app instance, not a person: read the fragmentation caveat.
WITH buyers AS (
  SELECT DISTINCT user_pseudo_id
  FROM `project.analytics_XXXXXX.events_*`
  WHERE _TABLE_SUFFIX BETWEEN '20260501' AND '20260531'
    AND event_name = 'purchase'                              -- or your own conversion event
)
SELECT
  TIMESTAMP_MICROS(e.event_timestamp)                              AS timestamp,
  e.user_pseudo_id                                                 AS visitor_id,
  (SELECT value.string_value FROM UNNEST(e.event_params)
     WHERE key = 'page_location')                                  AS landing_url,
  e.collected_traffic_source.manual_source                         AS source,
  IF(b.user_pseudo_id IS NULL, 0, 1)                               AS converted
FROM `project.analytics_XXXXXX.events_*` AS e
LEFT JOIN buyers AS b USING (user_pseudo_id)
WHERE e._TABLE_SUFFIX BETWEEN '20260501' AND '20260531'          -- the same window as the reach figures
  AND e.event_name = 'session_start'
  AND (e.collected_traffic_source.gclid IS NOT NULL
       OR REGEXP_CONTAINS(
            (SELECT value.string_value FROM UNNEST(e.event_params) WHERE key = 'page_location'),
            r'(fbclid|ttclid|gclid|gbraid|wbraid|msclkid|li_fat_id)=')
       OR e.collected_traffic_source.manual_medium IN ('cpc', 'paid', 'paid_social', 'ppc'));
```

How to run the notebook

Open notebook.ipynb in Jupyter or upload it to Google Colab. Run every cell once as it is: the synthetic market runs first and shows what a clean result looks like. Then find the cell headed 'your inputs', set YOUR_VISITOR_FILE or YOUR_LANDING_LOG to the file, type the reach figures into YOUR_REACH as {'meta': ..., 'tiktok': ...} using the platform names the notebook prints ('meta', 'tiktok', 'google' - 'facebook' and 'instagram' are read as meta), set the optional inputs, and run the cells below it. Nothing is uploaded anywhere; the notebook runs on the machine it is opened on.

What the method does, in the order the notebook prints it

1. THE EXACT PART. Distinct visitors = visitors via A + visitors via B - visitors via both. Visitors B added that A had not = visitors via B - visitors via both. With the converted flag: the customers who arrived via B and never via A, and those who arrived via both, reported as shared and never split. These are counts of identified visitors and carry no estimate. Report them first. What a new customer cost is the site's own arithmetic over those counts and a holdout's question, not this method's output.

2. THE CLICK-BASED FLOOR. Overlap = (visitors via both) x (reach A) x (reach B) / (visitors via A x visitors via B). Unique reach = reach A + reach B - overlap. This is exact only if the people reached on both platforms click no more than anyone else. It is a FLOOR under a stated condition: that the people on both platforms click at least as readily as each platform's average and their two clicks are linked through who they are - the usual case, but a statement about the market, not a law. Say the condition when you report the floor.

2b. THE STRATIFIED FLOOR, where the segment is available. Compute the overlap of step 2 within each segment from that segment's own reach and visitor counts, cap each at the segment's smaller reach, and add them up. The combined sampling half-width is 1.96 x sqrt(sum over segments of overlap_s^2 x (1/shared_s + 1/visitors A_s + 1/visitors B_s)) / total overlap; refuse the stratified figure when it exceeds twenty per cent and

report the pooled floor. It takes the between-segment part of the keenness out and is still a floor.

3. THE CHANCE CEILING. Overlap = reach A x reach B / population. Unique reach = reach A + reach B - that overlap. Audiences overlap more than chance, so this is a CEILING - and only with a population figure; without one the ceiling is the plain sum.

4. THE RANGE. The truth sits between the floor and the ceiling. Report both ends and say that nothing in the data says where in between. The floor's sampling interval is overlap x exp(+/- 1.96 x sqrt(1/shared + 1/visitors A + 1/visitors B)); it measures noise, not the keenness bias.

5. THE GUARDS - apply both, because the formulas alone do not. If the click-based overlap comes out larger than the smaller platform's reach, that is impossible: cap it at the smaller reach and say the floor has collapsed to the larger reach because the double-reached are clicking far more than others. If the visitors from both platforms are fewer than 97, refuse the estimate: the sampling error alone would exceed twenty per cent. The exact part still stands.

What this brief is worth, measured

A rebuild of the method from the formulas above was diffed against the notebook on its five pinned markets: the formulas alone reproduce the notebook on 4 of 5 pinned markets; on 1 the formula's overlap exceeds a platform's whole reach, which the notebook caps and flags, and on 1 the notebook refuses to estimate at all while the formula answers anyway. The market where the formula overlap is impossible is a very keen pair; the market the notebook refuses is a thin pair. That is why step 5's two guards are in this brief: the arithmetic is easy, and it is the guards a rebuild leaves out.

On the shipped demo files, the notebook refuses the estimate on the thin demo export (35 shared visitors against a floor of 97), refuses one-platform files and a reach smaller than its own visitors in plain words, and fails a wrong-columns file rather than producing a number; a rebuild from the brief's formulas does none of that.

The refusal of a single total is not caution; it is measured. In the article, the twin markets' reports differ by at most 3.0 per cent (sampling noise on the shared count), their click-based readings by 1.7 per cent, and their true unique reaches by 29 per cent - the formulas cannot tell them apart, which is the refusal.

What narrows the range, and what does not

More of the same data does not: the keenness bias is a property of who clicks, not of how many, so a longer window narrows the sampling interval and leaves the bias. Three things do, and all three are in the paper the method rests on. (1) Stratify by a segment both sides report - step 2b; on the article's market it took the floor from 22.3 to 15.6 per cent below the truth. (2) A random second sample: the bias vanishes when the second capture is a random sample of the exposed audience, however selective the clicks; a platform-run survey of a random exposed group is that instrument. (3) Covariates that predict clicking, through a conditional model the notebook does not carry. For the total itself, only a panel.

The caveats to say out loud

- The click-based figure is a floor under a condition, not an estimate. On the article's market it ran 22 per cent below the truth, and across the scored markets it ran 16 per cent low on a typical one. No report shows the keenness, and more data does not estimate it. What the clicks can speak for is the observable audience - the people with a real chance of ever visiting - and even that needs several captures; with two platforms and one window, bound the total and give no point.
- The chance formula is a ceiling and needs a population figure nobody reports. With the figure guessed within a factor of two either way it landed within ten per cent on 59 per cent of scored markets.
- The share of visitors who came from both platforms is NOT the overlap; it is the overlap times two click rates. On the article's market it was 1.0 per cent of visitors against a reach overlap of 20 per cent.
- Cookie fragmentation pushes the click-based figure UP, toward the sum: every arrival the site cannot tie to the same person shrinks the shared count. In-app browsers, second devices and cleared cookies all do it.

A logged-in identity does not.

- Consent scales the shared count in opt-in regions. If the site counts single-platform arrivals on the server side and shared visitors only among consenting people, the shared count is limited by consent. Dividing it by the consent rate assumes people who declined overlap the way consenting ones do, which nothing they hold can check; report the count as consent-limited, and if they divide, say what it assumes.
- The reach figures are the platforms' own estimates of accounts, sampled and modelled. Everything here starts from them.
- Reach is exposure, not effect. The overlap says whether the second platform reached new people, not whether those people did anything; that is a holdout's question.

Numbers to check yourself against

In the article's own synthetic market - 714,202 people reached on one platform and 481,237 on the other, with 22,368, 12,279 and 335 visitors via A, via B and via both, a population of 2,400,000 and a spend of 24,000 euros on B - the exact part is 34,312 distinct visitors and 11,944 visitors B added; the click-based floor is 776,226 unique people (overlap 419,213), the chance ceiling 1,052,230, the sum 1,195,439; 341 customers came via B alone and 9 via both; an extra person reached on B cost between 7.1 and 38.7 cents, an awareness figure. The built-in truth is 998,453 unique people, inside the range. The five pinned markets, their reports and every expected reading are in data/checks.json. If your rebuild reproduces them, you have it right.

Synthetic data throughout the article and notebook. The method transfers; the numbers do not - which is why the notebook reads the counts and the range off the reader's own files rather than shipping them.