

Why do today's numbers look bad? Run it on your own data

A brief to hand to your AI assistant - Stochastic Strata

For you (the human) - read this first

This is a little technical: it means exporting a table from your ad platform or analytics, and running a free notebook. You do NOT have to do it alone. Give this whole document to a capable AI assistant - ChatGPT, Claude, Gemini or similar - by uploading the PDF or pasting its text, then say: "Help me do this, step by step." Everything the assistant needs is below, even if you have never written a line of code.

What you get at the end: two numbers. One says how much of a recent day's conversions have already arrived, so you know what a young number is really worth. The other says how many days you have to wait before a number is finished, which is how long your reporting window honestly has to be.

One thing only you can supply: a file that includes the clicks that did NOT convert. Most exports people reach for are conversion lists, and a conversion list cannot answer this question at all. The clicks still waiting are the entire signal.

The post, the notebook, the spreadsheet and the interactive version are here:

<https://www.stochasticstrata.com/blog/why-todays-numbers-look-bad>

Instructions for the AI assistant

You are guiding a marketer through measuring how long their conversions take to arrive after the click, and turning that into two usable numbers. The method is a survival fit on the click cohort: every click that has not converted YET is kept in the data as an unfinished observation rather than a failure, and the population is split into a share that will eventually convert and a share that never will. That fit is the teaching route; the platform itself reads both answers straight off the Kaplan-Meier curve, only as far as the history still holds at least 30 clicks and one in twenty of the cohort waiting, and never extends a fitted tail. The notebook prints both. Guide them ONE STEP AT A TIME, in plain language, at their level. Do not assume they know SQL, BigQuery or Python.

Start by asking what they can export. The best case is one row per click with a conversion date where one exists. The common case is that they have a conversion report and nothing else - if so, say plainly that it cannot answer this, and help them get the clicks as well.

You need the notebook. This document is not enough on its own.

Read this before you improvise. This brief tells you how to GET the data and how to READ the answer. It does not specify the method in enough detail to rebuild it, and the parts it leaves out are the parts that decide the answer: how the unconverted clicks enter the likelihood, how the never-convert share is separated from the slow converters, and how the interval carries both the uncertainty in the curve and the randomness in which conversions have landed so far. A version written from this description will look reasonable and be wrong. Download and run the notebook from the page above. Do NOT write your own version of the analysis and present its numbers as the method's.

Step 1 - get one row per click

The notebook needs a CSV or .xlsx with three columns, one row per click (a campaign column is optional and lets them fit one campaign at a time):

```
click_date,converted,conversion_date
2026-08-14,1,2026-08-17
2026-08-14,0,
2026-08-15,0,
2026-08-15,1,2026-08-15
```

click_date is when the click happened. converted is 1 or 0. conversion_date is filled in only when converted is

1. Clicks that have not converted stay in the file with converted = 0 and an empty conversion_date - those rows are not padding, they are what the method reads.

Two routes to that file, in order of preference.

- GA4 THROUGH BIGQUERY. If their GA4 property exports to BigQuery, the query below produces the file directly. Adapt the placeholders for them - project, dataset, date range, conversion event name - and explain what each one is before asking. It keeps each user's first paid click, read off the session's own source, and their first conversion after it. The dates come out as UTC timestamps; the notebook reads them to the minute. If their campaigns differ a lot, have them set CAMPAIGN in the notebook and fit one at a time.

```
WITH paid_clicks AS (
  -- The session's own source, read off the event that carried the ad click. Not
  -- traffic_source: that holds the user's FIRST-ever source, whatever brought them now.
  SELECT
    user_pseudo_id                                AS user_id,
    TIMESTAMP_MICROS(event_timestamp)              AS click_time,
    collected_traffic_source.manual_campaign_name    AS campaign,
    collected_traffic_source.gclid                  AS click_id
  FROM `<YOUR_PROJECT>.<YOUR_DATASET>.events_*`
  WHERE _TABLE_SUFFIX BETWEEN '<YYYYMMDD_START>' AND '<YYYYMMDD_END>'
    AND (collected_traffic_source.gclid IS NOT NULL
        OR collected_traffic_source.manual_medium IN ('cpc', 'ppc', 'paid'))
),
first_click AS (
  SELECT user_id,
    ARRAY_AGG(STRUCT(click_time, campaign, click_id)
      ORDER BY click_time LIMIT 1)[OFFSET(0)] AS c
  FROM paid_clicks
  GROUP BY user_id
),
conversions AS (
  SELECT
    user_pseudo_id                                AS user_id,
    TIMESTAMP_MICROS(event_timestamp)              AS conversion_time
  FROM `<YOUR_PROJECT>.<YOUR_DATASET>.events_*`
  WHERE _TABLE_SUFFIX BETWEEN '<YYYYMMDD_START>' AND '<YYYYMMDD_END>'
    AND event_name = '<YOUR_CONVERSION_EVENT>'
)
SELECT
  f.c.click_time                                AS click_date,
  f.c.campaign                                  AS campaign,
  f.c.click_id                                  AS click_id,
  IF(MIN(v.conversion_time) IS NULL, 0, 1)      AS converted,
  MIN(v.conversion_time)                       AS conversion_date
FROM first_click f
LEFT JOIN conversions v
  ON v.user_id = f.user_id
  AND v.conversion_time >= f.c.click_time
GROUP BY f.user_id, click_date, campaign, click_id
ORDER BY click_date
```

- AD PLATFORM REPORTS. In Google Ads the timing half is a standard segment: Campaigns > Segment > Conversions > Days to conversion. That gives the shape of the delay but not the unconverted clicks, so it suits the SPREADSHEET route rather than the notebook. If that is all they can get, point them at the workbook on the post page instead, and tell them it needs a date range that ENDED at least a month ago, because a recent window is still filling up and the sheet cannot allow for that.
- A RAW EVENT LOG. If they have per-touch rows with a user id, an event name and a timestamp, the notebook carries a consolidator that turns it into the format above. Point them at the function called consolidate_event_log and set its conversion event name for them.

Step 2 - how long a history to pull

Ask for at least two to three times their expected conversion window, and ideally more. The reason is specific: the method finds the share who never convert by looking for where the curve flattens out, and a history shorter than the conversion cycle never flattens. If it cannot see that flat stretch, the notebook says so and tells them

the window it reports is an over-estimate. The platform's own reading errs the other way on a short history: its window reads short and its multiplier too small, and it says so when the curve is still rising at the edge of the data.

Eight weeks is enough for an account whose conversions are mostly in within about three weeks, like the post's example; a slower account needs more. Thirty converted clicks is the floor below which the notebook refuses to fit at all.

Tell them this BEFORE they act on the number

These are not small print. Say them out loud before the reader takes a decision to anyone else.

- A YOUNG DAY IS NOT A BAD DAY. This is the whole point and it is worth saying plainly, because the reflex it corrects is strong. On the post's example account a day-old number is showing about a quarter of what it will finish with. Nothing is wrong with that campaign.
- THERE IS A POINT WHERE THE MULTIPLIER STOPS BEING HONEST, and the notebook refuses past it. Below about 30% arrived, the multiplier is above three, and it magnifies every wobble in the estimate by the same three. The answer for a day that young is 'not yet', not a cleverer estimate. Do not talk them past that refusal.
- READ THE CURVE NO FURTHER AHEAD THAN THEIR OWN HISTORY REACHES. Inside the days their clicks have actually been watched over, the curve is measured; beyond it, a multiplier is the shape of the fitted tail, not conversions anyone has seen. Call any horizon beyond their own history a projection, at any length, and trust it less the further out it sits. The platform does not project at all: past the last age the history can read it gives no multiplier and says why.
- A PLATFORM EXPORT ONLY SHOWS CONVERSIONS INSIDE ITS OWN WINDOW. From Google Ads, each conversion counts only inside the conversion window they set (1 to 90 days), so a short window hides the slow tail: more clicks look like they never convert, and the fitted window comes out short. Fit on a window at least as long as the one they mean to read, or on first-party timestamps.
- ONE CURVE PER CONVERSION TYPE, AND THE REPORTING DELAY ON TOP. A form fill, a call and a purchase arrive on different clocks: fit each conversion action on its own. The platform's own delay in showing a recorded conversion adds to the click-to-conversion delay: fit on rows old enough to be fully reported.
- THE CURVE IS THEIR ACCOUNT'S, AND ONLY FOR NOW. Conversion timing moves with the season, with a shift toward colder audiences, and with consent or tracking changes. Tell them to re-fit monthly, and after anything that changes who is clicking.
- IT ASSUMES THE CLICKS STILL WAITING ARE LIKE THE ONES THAT ALREADY CONVERTED. The notebook cannot test this for them. If they changed who they were reaching part-way through the fitting window - a new audience, a new market, a shift in targeting - the recent clicks are a different population from the older ones, the fit is biased, and nothing in the output will say so. Fit on a stretch where the targeting held still, and re-fit from after a change rather than across it.
- ONE CURVE PER ACCOUNT IS ALREADY AN AVERAGE. Brand search and cold prospecting convert on completely different timescales. If their campaigns differ that much, the honest move is to fit them separately rather than to apply one multiplier to all of them.
- IT SAYS NOTHING ABOUT WHETHER THE ADS CAUSED THE CONVERSIONS. This is a timing correction. A lag-adjusted number is still a reported number, with all the attribution problems it had before. If they want causation, they need a holdout test, and you should say so rather than let the correction stand in for one.
- IT CORRECTS IN ONE DIRECTION ONLY. It fixes numbers that are too low because conversions have not arrived. It does nothing about numbers that are too high because two platforms both claimed the same sale.

How to report it back to them

Give them three sentences. What share of a recent day has arrived, and therefore what that day is really worth. How long their window has to be before a number is finished. And which days in their current report are too young to read at all. For example: 'a day-old number is showing about a quarter of its final total, your numbers are not complete until about day twenty-four, and the last two days of this report should not be read yet.' Do not present a multiplier without saying what share it came from.

Running the notebook

Two routes. Easiest, nothing to install: Google Colab, where the notebook runs on Google's servers - which means the file they upload goes to Google, and they should be told that before they upload anything sensitive. The private route is Jupyter on their own machine, where the file never leaves it. Say which route they are on, plainly. People ask, and the honest answer is not always the convenient one.

There is exactly one cell to edit, marked YOUR FILE HERE. It holds the filename (or, on the raw-log route, the log's filename and the name of its conversion event), and optionally one campaign to fit, the date to treat as 'today' if their export ends before today, and DAY_FIRST if the notebook says it cannot tell 06/07 from 07/06. Set those for them. Everything else runs unchanged; with a file set, the demonstrations below it are skipped.

If their dates are whole days, the notebook prints two readings that differ for the first few days. Tell them to act on the platform's reading inside their history for the first week; the notebook says so too. Yesterday, read in the morning, is day 0 in that table.

The notebook will print refusals rather than numbers when the data cannot support an answer: too few conversions, a history too short to see the curve flatten, or a horizon where too little has arrived. Read those out to the reader as findings. They are the method working, not the method failing.