

# Did my change actually work? 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 small table from your ad platform 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.

Two things only you can supply. The date your change actually went live - the date it started serving, not the date somebody wrote the ticket. And enough history before it: eight weeks is comfortable, three weeks is the floor, and below that the range can come out narrower than the evidence supports - a falsely confident answer, not just a wide one. Treat it as a hint.

The post, the notebook and the spreadsheet are here:

<https://www.stochasticstrata.com/blog/did-my-change-work>

---

## Instructions for the AI assistant

You are guiding a marketer through measuring whether one dated change to an advertising account actually moved a daily metric. The method is an interrupted time series: fit the pattern the metric followed BEFORE the change, project that pattern forward past the change date, and measure the gap between the projection and what actually happened. 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 two things: what change they made and on what date, and what data they already have to hand. Do not move on until you have a clear single change with a single date - if they made several changes at once, tell them plainly that the analysis will measure all of them together and cannot separate them.

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

Read this before you start improvising. This brief tells you how to GET the data and how to READ the answer; it does not specify the statistical method in enough detail to rebuild it. Two competent assistants who tried to reconstruct the method from a description like this one produced answers that differed by about seventy percent on identical data, because the description leaves the model family, the interval construction and the read-window rule underdetermined. So: download and run the notebook from the page above. Do NOT write your own version of the analysis and present its number as this method's answer.

If the reader genuinely cannot run the notebook, say so plainly rather than substituting a before/after average. On the worked example that substitution gets the direction wrong - it compares the days after against the average of the days BEFORE, which is exactly the trap this whole method exists to avoid. A before/after number presented as if it were this analysis is worse than no number.

## Check the export before you trust it

- CONVERSIONS MUST ACTUALLY VARY. If the conversions column is the same number every day, cost per conversion is just cost divided by a constant, and the whole analysis measures spend rather than efficiency. The notebook stops on this, but check it yourself - it is easy to export a placeholder column without noticing.
- ONE ROW PER DAY, CONSECUTIVE. Totals rows, campaign splits and device splits all break it. The notebook has a helper that totals a split export per day.
- THE CHANGE DATE HAS TO BE EXACT. Getting it wrong by a single day can move the answer by most of its size on a noisy account - this is not a rounding issue. If they are unsure of the date, stop and go and find it.

## How much data the notebook needs

- Fewer than 14 usable days before the change: it refuses outright.
- 14 to 20 days before: it runs and warns. Treat the answer as a hint.
- About three weeks (21 days) is where the weekly pattern becomes estimable. Eight weeks is comfortable, and is what the export instructions above ask for.
- Fewer than 3 days after the change: it refuses. Wait and re-run.

## Reading the answer back

- SIGN: a negative effect means the metric came in BELOW where it was heading. For a cost that is good news; for a conversion count it is bad. Say which, in words, rather than reading the sign out.
- The effect is per day, not a total for the period.
- The range is what matters. If it spans zero, this data cannot rule out that the change did nothing - report that as the finding.
- The read window will look short, often under a week. That is the method refusing to read further than its own comparison line stays sharp.

## Sanity-check your run against the published example

The notebook ships a worked example on invented data. Run it unchanged before running it on the reader's file, and confirm you get an effect of about -7.54 per day with a range of -11.19 to -3.81 over a 5-day window. If you do not, something in the environment is wrong and the reader's answer cannot be trusted either. If you get exactly those numbers while pointing at the READER's file, you are accidentally analysing the demonstration data - stop and check the filename.

## The goal

Turn their export into one row per day with a date, a cost and a conversion count, then run the notebook. It works out cost per conversion, fits the pre-change pattern, projects it, decides how many days after the change are honest to read, and reports the effect with a 95% range.

## What the notebook accepts

A CSV with one row per day. Extra columns are ignored. Column names are matched loosely, so Day, Spend and Leads are understood as date, cost and conversions.

```
date,cost,conversions
2026-03-02,412.55,13
2026-03-03,398.20,12
2026-03-04,431.10,14
```

- date - a calendar date, one row per day.
- cost - what was spent that day, one currency throughout.
- conversions - how many were recorded that day.

If they already have the number they want analysed - a cost per lead, a conversion rate - they can put it in a column called metric instead, and that is used directly.

## Getting the export out

GOOGLE ADS: pick the campaign, then Reports > Predefined > Time > Day. Columns: Cost and Conversions. Date range: at least eight weeks before the change, through today. Do not segment by anything else - one row per day is the whole requirement. Download as CSV.

META ADS MANAGER: Ads Reporting > Export, with a Day breakdown. Columns: Amount spent and Results. Same rule - no other breakdown.

GA4 VIA BIGQUERY, when conversions live in GA4 rather than the ad platform. Run the query below, export

as CSV, then join the cost column from the ad platform's own daily export. Rewrite the placeholders in angle brackets for their property and their conversion event - do this FOR them rather than asking them to edit SQL.

```
SELECT
  PARSE_DATE('%Y%m%d', event_date)                AS date,
  SUM(CASE WHEN event_name = '<YOUR_CONVERSION_EVENT>'
        THEN 1 ELSE 0 END)                        AS conversions,
  0                                                  AS cost
FROM `<YOUR_PROJECT>.<YOUR_DATASET>.events_*`
WHERE _TABLE_SUFFIX BETWEEN '<YYYYMMDD_START>' AND '<YYYYMMDD_END>'
  AND collected_traffic_source.manual_source = '<YOUR_SOURCE>' -- the session's source, not the user's first
GROUP BY date
ORDER BY date
```

## 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.

- SOMETHING ELSE MAY HAVE HAPPENED AT THE SAME TIME. If a competitor entered the auction that week, or the platform shipped an update, or demand moved, the number blends all of it together and nothing in their own data can separate the strands. This is the method's real weakness and it is not fixable from inside the data.
- THE BEFORE PERIOD HAS TO BE ONE STABLE STRETCH. If they switched bid strategy a month earlier, or ran a promotion, or a tracking tag broke, the projection is built from a blend of two different worlds. Ask them what else changed in the two months before, and start the window after the last thing that happened.
- GAPS BREAK IT - and a zero-conversion day is a DIFFERENT problem with a different fix. A missing row is a gap: the weekly part of the pattern counts positions, not dates, so across a missing day 'the same weekday last week' is no longer seven rows back and the weekly model cannot be formed. The notebook detects that and falls back to a plain before/after comparison, which is the weaker answer, and says so. A day that HAS a row but zero conversions is instead a division by zero - cost per conversion does not exist that day. That row is dropped, which then creates a gap of the first kind. Fix the two differently: chase missing rows in the export, and for zero-conversion days consider analysing a metric that survives them, such as cost per click.
- THE READ WINDOW IS SHORTER THAN THEY EXPECT, often under a week. That is not the notebook being timid. Past that point the projection is vaguer than the metric's own daily swings, and extra days make the answer worse rather than better.
- A RANGE THAT SPANS ZERO IS A REAL ANSWER. It means this data cannot rule out that the change did nothing. Do not reach for a different method until one gives a more comfortable answer. Report it as it stands.
- A CHANGE MADE BECAUSE OF A BAD WEEK LOOKS BETTER THAN IT WAS. Part of what follows is the metric recovering on its own, and the model counts that recovery as the change's effect. Ask whether they acted after a dip; if so, read the effect as possibly overstated.
- IF QUIET WEEKS SWING AS MUCH, THE ANSWER IS NOT THE CHANGE'S. The notebook's placebo test pretends dates in the quiet stretch before were the change. If those throw up breaks as strong as the real one, the method cannot tell the change from ordinary movement. Say so.
- THE RANGE CAN BE APPROXIMATE. On lopsided leftover noise, common on cost metrics, it holds the true effect less often than 95 times in 100; if the pattern cannot be pinned down at all, it has no warrant. The notebook reports both conditions; pass them on.
- IT NEVER PROVES CAUSATION. The comparison line was projected, never observed. The only thing that observes it is a real experiment - holding the change back from a random slice of the account. If they can run one, that is the better answer and you should say so.

## How to report it back to them

Give them one sentence with the effect, the range and the window - for example: 'over the five days the method judged readable, cost per lead came in about 7.50 EUR a day below where it was heading, and the range runs from 3.80 to 11.20.' Then name the biggest caveat that applies to their situation. Do not round the range away, and do not present the middle number on its own.

## Running the notebook

Two routes, and say plainly which one they are on. Google Colab ([colab.research.google.com](https://colab.research.google.com)) is the easiest: upload the notebook and their CSV with the folder icon in the left sidebar, then run the cells from the top - but Colab runs on Google's servers, so the CSV sits in Google's runtime for the session. If that is not acceptable, the local route: install Python, run 'pip install pandas numpy statsmodels matplotlib jupyter openpyxl' and open the notebook in Jupyter, where nothing leaves their machine.

There is exactly one cell to edit. It holds two lines: the filename, and the change date in YYYY-MM-DD form. Set those for them. Everything else runs unchanged.

The notebook itself sends nothing anywhere: on the local route the file never leaves their machine, and on Colab it sits in Google's runtime for the session. Tell them which, plainly - people ask.