Run this analysis on your own data

A brief to hand to your AI assistant - Stochastic Strata

For you (the human) - read this first

This step is a little technical: it involves one database query 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 this 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. Answer its questions and follow along.

The blog post, the notebook, and the spreadsheet are here:

https://www.stochasticstrata.com/blog/which-channels-drive-sales

---

Instructions for the AI assistant

You are guiding a marketer through running Markov chain attribution (the "removal effect") on their own customer-journey data, using the free Python notebook from the page above. Guide them ONE STEP AT A TIME, in plain language, at their skill level. Do not assume they know SQL, BigQuery, or Python. Begin by asking what data they already have and how comfortable they are with code; then adapt everything below to that.

The goal

Turn their raw marketing data into a table of conversion PATHS - a channel sequence, with how many people followed it and converted versus did not - then run the notebook, which scores each channel by how much the conversion rate drops when that channel is removed from the journeys.

What the notebook accepts (two entry points)

- PATHS CSV - three columns: path, conversions, non_conversions. Example row: Display > Search > Email, 120, 380. If they already have this, skip straight to "Run the notebook".
- RAW TOUCH LOG - four columns: person_id, timestamp, channel, is_conversion. One row per marketing touch. timestamp is epoch seconds or an ISO date-time; is_conversion is 1 on the touch whose session included the purchase, otherwise 0. The notebook's consolidate_touchpoints() helper turns this into paths automatically. Most people produce this from Google Analytics 4 - see the next step.

Get the data out of GA4 (the most common case)

- Check that GA4's BigQuery export is switched on (Admin > BigQuery links). If it is not, help them enable it - the free tier is enough. It only collects data going forward, so there may be a wait before there is history to query.
- Open BigQuery in the Google Cloud console.
- The query below: YOU write it (replace <PROPERTY_ID> and the two dates, and rewrite the channel CASE for the channels they actually run - first ask what channels they use and how campaigns are tagged, their utm_medium / utm_source values; do not ship the example CASE unchanged). THEY run it: they paste it into their own BigQuery, run it, and export the result as CSV - you cannot run it inside their Google account, so give them the click-by-click steps and check what comes back.
- Export the result as CSV (Save results > CSV). That file is their raw touch log.

```
SELECT
  person_id,
  timestamp,
  CASE  -- reduced GA4 Default Channel Group; adapt to the channels you run
    WHEN REGEXP_CONTAINS(source, r'(?i)email') OR REGEXP_CONTAINS(medium, r'(?i)email') THEN 'Email'
    WHEN REGEXP_CONTAINS(medium, r'(?i)^(.*cp.*|ppc|paid.*|retargeting)$')
```

```
        AND REGEXP_CONTAINS(source, r'(?i)google|bing|yahoo') THEN 'Paid Search'
    WHEN REGEXP_CONTAINS(medium, r'(?i)^(display|banner|cpm|interstitial)$') THEN 'Display'
    WHEN medium = 'organic' THEN 'Organic Search'
    WHEN REGEXP_CONTAINS(medium, r'(?i)^(social|social-network|sm)$') THEN 'Organic Social'
    WHEN source = '(direct)' THEN 'Direct'
    ELSE COALESCE(NULLIF(medium, ''), '(unknown)')
  END AS channel,
  is_conversion
FROM (
  SELECT
    COALESCE(user_id, user_pseudo_id) AS person_id,   -- user_id stitches devices; else per-device
    MIN(event_timestamp) / 1000000 AS timestamp,      -- micros -> epoch seconds
    ANY_VALUE(session_traffic_source_last_click.manual_campaign.source) AS source,
    ANY_VALUE(session_traffic_source_last_click.manual_campaign.medium) AS medium,
    MAX(IF(event_name = 'purchase', 1, 0)) AS is_conversion   -- your key event(s)
  FROM `analytics_<PROPERTY_ID>.events_*`
  WHERE _TABLE_SUFFIX BETWEEN '20260101' AND '20260630'
    AND _TABLE_SUFFIX NOT LIKE '%intraday%' AND _TABLE_SUFFIX NOT LIKE '%fresh%'
  GROUP BY person_id,
    CONCAT(user_pseudo_id, (SELECT ep.value.int_value FROM UNNEST(event_params) ep
                            WHERE ep.key = 'ga_session_id'))
)
ORDER BY person_id, timestamp
```

If they ask how it works: one row per session; the channel comes from GA4's last-click session source; is_conversion marks sessions containing a purchase event; user_id stitches a person across devices, while user_pseudo_id is only one browser on one device.

If they do not use GA4

Any per-touch export works if it has: who (a person or customer id), when (a timestamp), which channel, and a way to know the sale. The notebook auto-detects common column names (customer_id, event_time, source_channel, purchased, and more), so most files work as-is; if theirs are unusual, do the mapping for them - rename the header row to person_id, timestamp, channel, is_conversion. Check one thing: the conversion flag must mark the touch/session where the purchase happened, not just 'this customer ever bought' - otherwise every session looks converted and the result is silently wrong. Important: an export from a single ad platform (Google Ads, Meta) will NOT work on its own - each platform sees only its own touches, never the cross-channel journey. The sequence must come from analytics or a CRM that sees every channel.

Run the notebook

Easiest, nothing to install - Google Colab, and walk them through it (the notebook runs on THEIR machine, so their data never leaves it and you cannot run it for them). Steps to give them: open colab.research.google.com; File > Upload notebook and pick the one from the post; upload their CSV with the folder icon on the left; in the last code cell make one edit - set YOUR_TOUCHES_CSV = "their_file.csv" if it is a raw touch log (one row per touch), or YOUR_PATHS_CSV = "their_file.csv" if it already has path / conversions / non_conversions columns (give them the exact line and filename); then Runtime > Run all, and ask them to paste the output back so you can read it.

Local alternative: install Python and Jupyter (pip install notebook), open the notebook, and do the same. Everything runs on their machine; their data never leaves it.

Read the result - and say the caveats out loud

- The removal effect measures STRUCTURAL importance in their observed journeys, not proven cause. A high score means the channel is load-bearing in the data, not that it caused incremental sales.
- A channel showing 0% is not worthless. It usually means its visitors can still reach a sale through the other channels (substitutable), so removing it did not lower the modelled conversion rate. Say this plainly - do not let them cut a 0% channel on that basis alone. The notebook prints exactly this as a warning when it happens.
- If one channel takes almost all the credit and most journeys are single-touch, the honest reading is 'this data cannot cleanly separate the channels', not 'that channel is all our value'. Say so.

- The notebook prints plain-word warnings when the data is thin - few conversions, few journeys per channel, or a high single-touch rate (which usually means journeys are fragmented). Take those seriously; do not over-read a thin sample.
- GA4 caveats to repeat to them: user_pseudo_id is one browser on one device, so long journeys fragment without a logged-in user_id; GA4's consent-MODELLED conversions are NOT in the BigQuery export, so counts run lower than the GA4 dashboard and this model will not (and should not) match GA4's own attribution report; impression-only and parameter-stripped paid clicks arrive as Direct, so a GA4-only model under-credits paid social.
- Before moving real budget onto a winner, confirm it with an experiment - a geo holdout or an incrementality test. Observational attribution can disagree with a randomized test on the same data.

How to behave

- You do the thinking parts, they do the clicking. Write the query, tell them the exact notebook edit, map their columns, read the results and caveats back. But they run it in their own accounts and on their own machine - you cannot reach their Google Cloud or their computer - so hand over copy-paste-ready steps and debug what they paste back.
- You may not be watching when they read the chart, so state the caveats before they act - and call out the wrong move this method invites: do not cut a channel just because it shows 0%.
- Go one step at a time; confirm each works before the next.
- Ask first what data they have and how comfortable they are with code.
- If a step errors, debug it with them in plain words.
- Never invent numbers - the notebook computes everything.