

GA4 didn't kill Page Value. It scaled it.

How the move from session-scoped to event-scoped analytics turned a missing metric into a generalizable one, and how to rebuild it in the tools you already have.



Corey Abramson · May 2026 · 7 min read

When Universal Analytics retired, the loudest complaints had nothing to do with metrics. Analysts were being asked to give up something that had worked well enough for a decade and trust a replacement that launched untested and, by Google's own admission, unfinished. The migration itself was the headline.

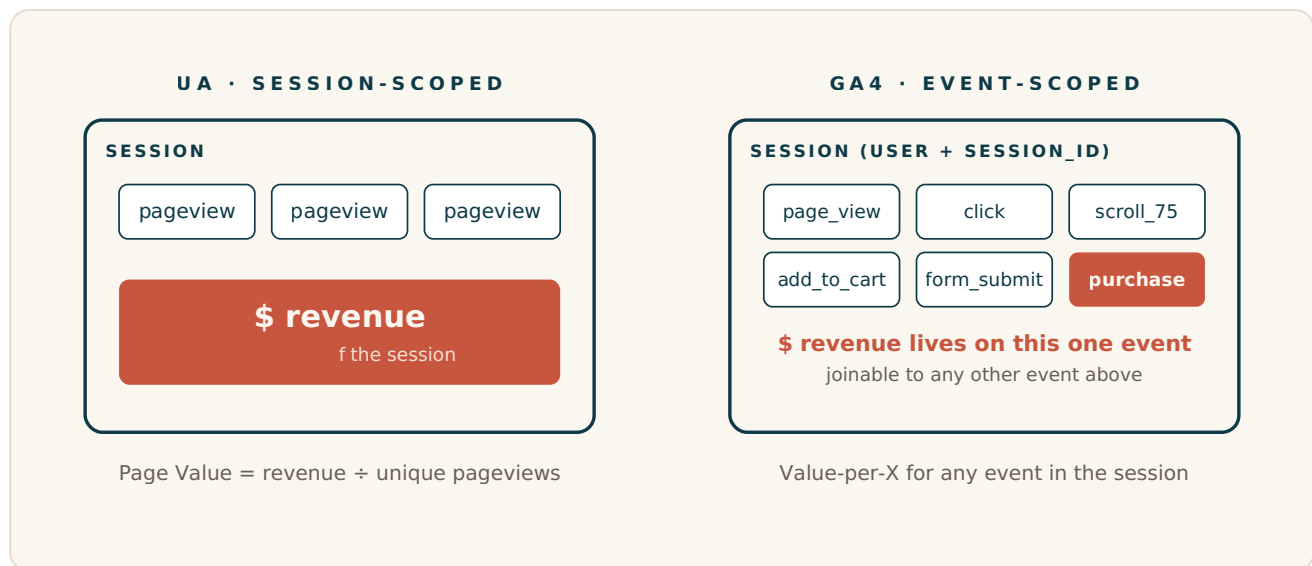
It wasn't until most of us had a year or so under our belts that the deeper story landed. The UX was a re-learn, fine. The bigger shift was structural and educational. Not only did we have to think about our data differently, we had to teach our stakeholders. Talk about building the vehicle as you're driving it.

Session-scoped reporting was gone. Event-scoped reporting was here, and it was different in ways that took real time to understand. Once you started cataloging what didn't make the trip, Page Value sat near the top of the list (stay tuned for my thoughts on Legacy Bounce Rates).

If you weren't living in UA, here's why that one stung. UA gave you a column in the Pages report called Page Value, calculated as revenue divided by unique pageviews of a URL. It was crude. It was session-scoped. And it was extremely useful. If a product page had a Page Value of \$3.28, you knew that page was, on average, present in a session

that generated \$3.28 of revenue. You could rank pages, sort categories, and defend optimizations to a CFO without needing a data engineer in the room.

GA4 took it away. Not because Google decided analysts didn't need it, but because the entire data model changed underneath. UA was session-scoped. A session was the atom, and revenue lived at the session level. GA4 is event-scoped. Every page view, every click, every scroll is its own atom, and revenue lives on the purchase event.



The shift that took most of us a year to internalize. UA bound revenue to the session. GA4 binds it to the purchase event, which means any other event in the session is fair game for attribution.

For a stretch, I treated that as a regression.

The reframe

Then I started writing the queries.

GA4's event-level export to BigQuery doesn't just let you replicate Page Value. It lets you generalize it. The session is no longer the unit of attribution. The user-session-event triple is. Which means I can attribute passthrough revenue and passthrough conversion volume to any event that fired during a session.

Not just page views. Element clicks. Scroll depth flags. Video plays. Form submits. The "size guide" button buried halfway down a PDP. A homepage hero swap. A new piece of social proof.

The math doesn't change. It's the simplest framing in analytics.

Successes over opportunities.

Of all the users who did X, how many converted, and how much did they spend. Page Value was answering one specific version of that question, where X is "viewed this URL." GA4 lets me ask it for any event the site fires.

What that looks like in practice

I started with the obvious move. Rebuild Page Value, but better. Pull every user who hit a given URL in a session, join to the same user's purchase events, divide. Now I had URL-level value, comparable across pages, with none of GA4's UI thresholding or sampling getting in the way.

Once that worked, automating it was a couple of days. Page-level value by URL, refreshed daily, with a 28-day rolling baseline. CRO conversations stopped being "I think this page is important" and started being "this page is worth \$6.10 per visitor today, up from \$4.40 sixty days ago. Here's the change history."

Then the same logic moved sideways into testing. If I can attribute revenue to any event, I can attribute it to any test cohort. Same query, one more filter. Conversion rate by variant. Revenue per user by variant. AOV by variant. Because I'm working from the raw event stream, I can grade a test on the revenue it actually moved, not just the primary metric the testing tool was pointed at.

VARIANT	CR	RPU	AOV
Control	3.4%	\$5.10	\$150
Variant	3.9%	\$5.03	\$129

Illustrative. Variant wins on conversion rate by 14%, but revenue per user holds flat because average order value dropped. A CR-only readout calls this a win. A financial readout calls it a wash.

That last move is the one that quietly changes how a CRO program runs. Most testing tools tell you a variant won on conversion rate. They don't tell you whether the conversions were cheaper, more expensive, or distributed differently across customer

segments. With the BigQuery layer in place, every test post-mortem gets a financial accounting, not just a statistical one.

Why this is the upgrade

GA4's data model is not a worse Universal Analytics. It's a different data model that happens to have a worse UI for the questions UA used to answer in one click. If you stop at the UI, the migration looks like a downgrade. If you push through to the export, the migration is a promotion. Every event becomes a candidate metric. Every interaction becomes attributable.

The analysts I see still frustrated with GA4 a few years in are mostly the ones trying to make it behave like UA. The ones who got over it are running ranged value calculations on micro-interactions UA was never going to expose to them.

If Page Value is what you miss, the honest answer is that you don't have to miss it. You can rebuild it, generalize it, automate it, and pipe it into your testing program. The metric isn't gone. The ceiling is.

GA4 didn't take Page Value away. It made every event in the session worth measuring.

For the analysts in the room

There are two ways to run this calculation, depending on what your org has set up. Both produce the same conversion rate and the same value-per-user. Pick the one that fits your stack.

In GA4 Explorations. Build one segment of users who triggered the event you care about. Build a second segment of users who triggered that event and converted. Pull both totals, drop them in a spreadsheet, divide. Conversion rate and revenue per user fall out of the same two numbers a query would give you. It is manual, it gets slow once you want to run it against fifty pages at once, and you will feel GA4's sampling and thresholding on the edges, but the framing is intact and the answers are usable. This is the path if your org has not connected GA4 to a warehouse, and plenty of strong programs run here permanently.

In BigQuery. Same logic, scaled. The GA4 events export gives you user, session, and event level data directly, so the calculation runs as a single query rather than a manual segment build. Once it is set up you automate it, schedule it, and treat the output as a

metric your team can sort and filter against. Faster than Explorations and uncapped by sampling, but it requires a warehouse connection your org may or may not have.

Whichever path you take, the framing flexes. A page view filtered by URL gets you classic Page Value. Cart adds get you the conversion rate and AOV of cart adders. A scroll depth event gets you the financial signal on scroll engagement. Filter by experiment cohort and you are grading A/B variants on revenue, not just conversion rate.

The data model does most of the work. The framing does the rest. Successes over opportunities, revenue over opportunities, both filterable by anything in the event stream.

Get your metrics back on track.

Lost Page Value. An event taxonomy you can't trust. A testing program that wins on CR but loses on revenue. If any of those sound familiar, you're talking to the right person.

frequentlybayesian.net/#contact-form · corey@frequentlybayesian.net · 215 498 7501

Let's Talk