How do product analytics work?

You're being watched, sort of

Justin
Apr 5, 2022

62 11

Product analytics – what the hell is it? Why is Amplitude a public company? What do product managers even do?

This Technically post will dive into how companies analyze what their users are doing in their product. We'll cover what kinds of questions teams want answers to, how the data gets generated and moved, and what tools are out there to simplify things.

> *This post was graciously sponsored by PostHog, the open source product analytics platform you can self-host. PostHog helps you across the stack from instrumentation to exploration, and is like, totally free.*
>
> PostHog

## It all starts with decision making

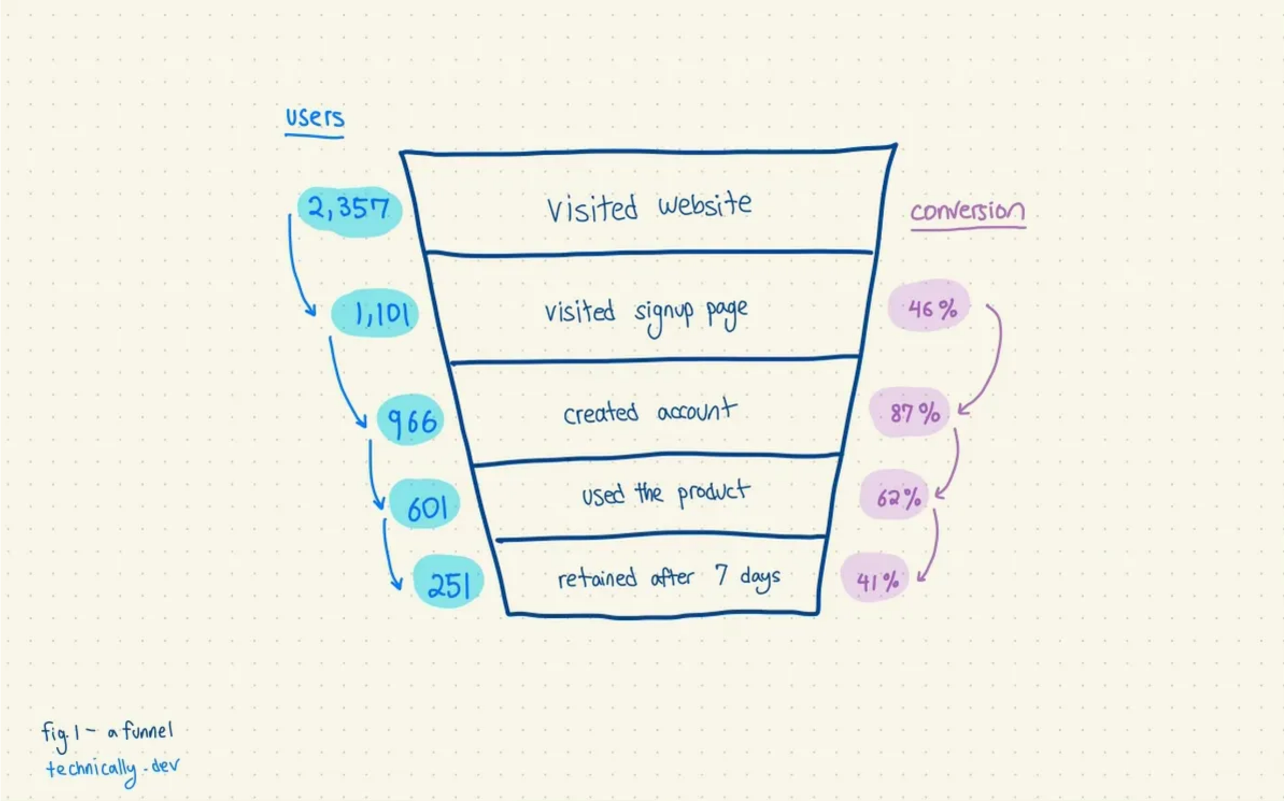
The wonderful world of product analytics starts with knowledge, or at least the eternal yet fleeting desire for it. Startups and large companies alike want **visibility** into how their users use their digital products. They're looking to understand things like:

- What are our most popular features?
- Which pages on our website convert the best?
- What areas of our product experience are confusing?
- What are common paths that users take to accomplish something?

Armed with this information, product teams can make educated decisions on what they should be **improving and building**. These kinds of analytics can also help catch bugs: if the number of users visiting a page drops precipitously one day, chances are something is wrong with that page, and needs fixing (or your analytics are broken 😏).

### Product funnels

Product and growth teams like to use the concept of a **funnel** to visualize the paths that a user takes through the product. The funnel is wider on top – representing actions that *a lot of people* take, like visiting a website, and narrower on the bottom, representing actions that *fewer people* take, like signing up for the product and using it.



Each stage of the funnel (except the top one) has a **conversion rate** representing the percentage of people in the previous bucket that made it to the current step. Teams try to benchmark their conversion rates against other products to get a sense of whether they are too low (or hopefully, too high).

> 🚨 **Confusion Alert** 🚨
>
> Even though it's referred to as *product* analytics, this area has blurry overlap with *marketing* data. Growth funnels can include data on website visits, and

> which sources (referrals, organic search, etc.) are driving quality leads. Who is responsible for using this data varies; just don't be surprised if you see traffic-related information – what you'd normally consider marketing – in this post.
>
> 🚨 **Confusion Alert** 🚨

### User research and session recording

Numbers can only go so far in giving product teams a good sense of what their users are actually doing in the product. Sometimes, you just need to **see** how they navigate around and do their tasks. Traditionally, **user research** teams would find prospective or existing users, and actually observe them accomplishing common tasks in your app. For example, the user research team at Microsoft might pay users to join a study and ask them to build a basic financial model in Excel; they'll pay close attention to exactly what the users do and what problems they run into.

More recently, tools like FullStory have popped up, which allow teams to literally **record** what users are doing in the product during their day to day, making all of this significantly easier. More on these tools later.

## How it all works under the hood

As with all things data, it's a pain in the ass to arm your company with the data to *answer* all of the questions product teams have. You need to instrument your product to "record" what users are doing, get that data into your warehouse, transform it into a format that's useful for analysis, and then build visualizations that give you the information you want.

This section will walk through how that's done, and the next will cover newer tools that are making things easier than they were.

Note: most tech companies today are using event-driven architectures like the one we're about to describe. But there are other, worse ways of doing this.

### Instrumenting the product

Step one is to actually create the data. And doing this is fairly simple, albeit custom: you just send what's called an **event** any time a user does something in the product. Suppose we'd like to record every time a user clicks on the "save changes" button when editing their profile. We'd fire an event like this:

```
analytics.track(
  {
    event_name: “save_changes_button_clicked”,
    timestamp: “2022-03-09 23:45:33”,
    context_page: “/profile/edit”,
    user_id: “239hg72eyrhdaadgdg99277hd7ve88d”
  }
)
```

The event gets a name, a timestamp, and some metadata: in this case, the context in which it took place (on the edit profile page), and the ID of the user who took the action.

Fun fact, you don't need to take my word for this. If you're logged into a SaaS product on your computer right now, open up your developer tools (⌘ + Option + C for chrome) and click on the "network" tab. Click on any button or link in the app, and you should see some events showing up at the bottom.

> 🔍 **Deeper Look** 🔍
>
> The logistics of sending these events gets interesting. Most companies send them client side, e.g. right from your browser. But if you're using an ad blocker, that blocks any client side JavaScript from hitting external servers, which renders these events useless. For that reason, some companies will set up a proxy server and send these events from their backend; more work, but it avoids the ad blocker problem.
>
> 🔍 **Deeper Look** 🔍

These events get created and fired for pretty much anything you do in the product, and later get assembled into useful insights. But first, the data needs to be stored somewhere.

### Getting the data into the warehouse

This is where things start to get complicated. Once the event gets "fired" where does it go?

These days, the state of the art is to put it straight into your data warehouse. There are a few ways to do that. You can build it yourself – you'd need to build a system that parses the event data and inserts it into Snowflake, BigQuery, etc. via SQL – or, more commonly, use a third party tool like Segment or PostHog to do it for you.

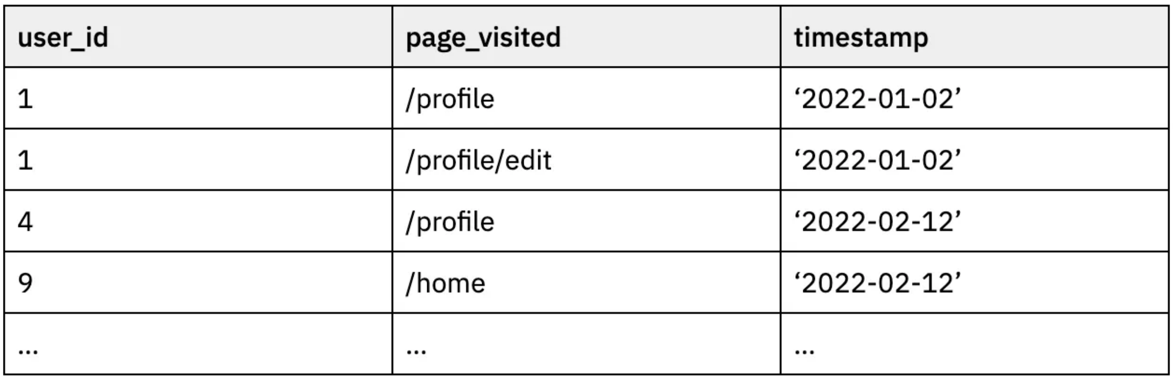
The end "product" is usually a bunch of tables in your warehouse, one for each type of event. You'll have a table called `clicked_save_changes`, one called `edited_profile`, and so on and so forth. Sometimes these events will get aggregated into a big events table called, well, `events`. Each one of these tables will have columns for the important data: the time the event happened, which user did it, what page it happened on, etc.

Now we're getting somewhere!

### Transforming the data into something useful

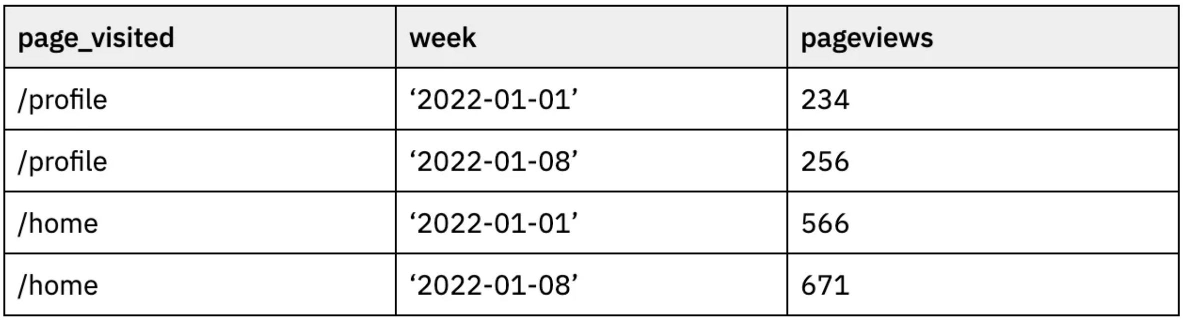
A bunch of raw events aren't exactly useful for a product manager, even if they *can* write SQL. How am I supposed to know what my funnel looks like when I'm working with a bunch of random events? The next step in the process is to **transform** the event data into aggregated, more useful information.

Transforming data today just means writing some SQL that takes these events and filters and aggregates them. For a concrete example, imagine we're trying to find the most popular "page" in our product, and look at how many users visit each page on a weekly basis. The raw event data looks like this:



user_id	page_visited	timestamp
1	/profile	‘2022-01-02’
1	/profile/edit	‘2022-01-02’
4	/profile	‘2022-02-12’
9	/home	‘2022-02-12’
…	…	…

But we want the data to look like this:



page_visited	week	pageviews
/profile	‘2022-01-01’	234
/profile	‘2022-01-08’	256
/home	‘2022-01-01’	566
/home	‘2022-01-08’	671

To do that, we write a bunch of SQL (I will spare you the logistics). State of the art today is to use something like dbt to do this. This is usually the responsibility of the data team, and they will work with business stakeholders like the product team to figure out what these transformations should actually look like.

SQL isn't necessarily the ideal tool for analyzing event data like this. While we do eventually want to aggregate a lot of this data into rollup tables like the above, sometimes explorations into why users are doing a particular action requires **diving into the granular events themselves**. Writing custom SQL queries for these over and over again, and analyzing the results tabularly, is a nightmare. In other words, there are still many open questions as to how this type of data should be analyzed most effectively.

### Visualization, BI, and exploration

Once the data is in the format we want it in, we'll want to get it into a chart of some sort. There are *tons* of ways to do this, from Metabase to Retool to Hex to homegrown solutions. Even Looker, before it was all but destroyed by acquirer Google, was a tool that teams used for this, among other things. You'll usually end up with a series of graphs that update regularly and show you the metrics you care about.

The final product isn't *always* a static graph, though. More and more, data teams are working on ways to **empower** non-technical stakeholders to explore data themselves and answer their own questions. I wrote a bit about this distinction here.

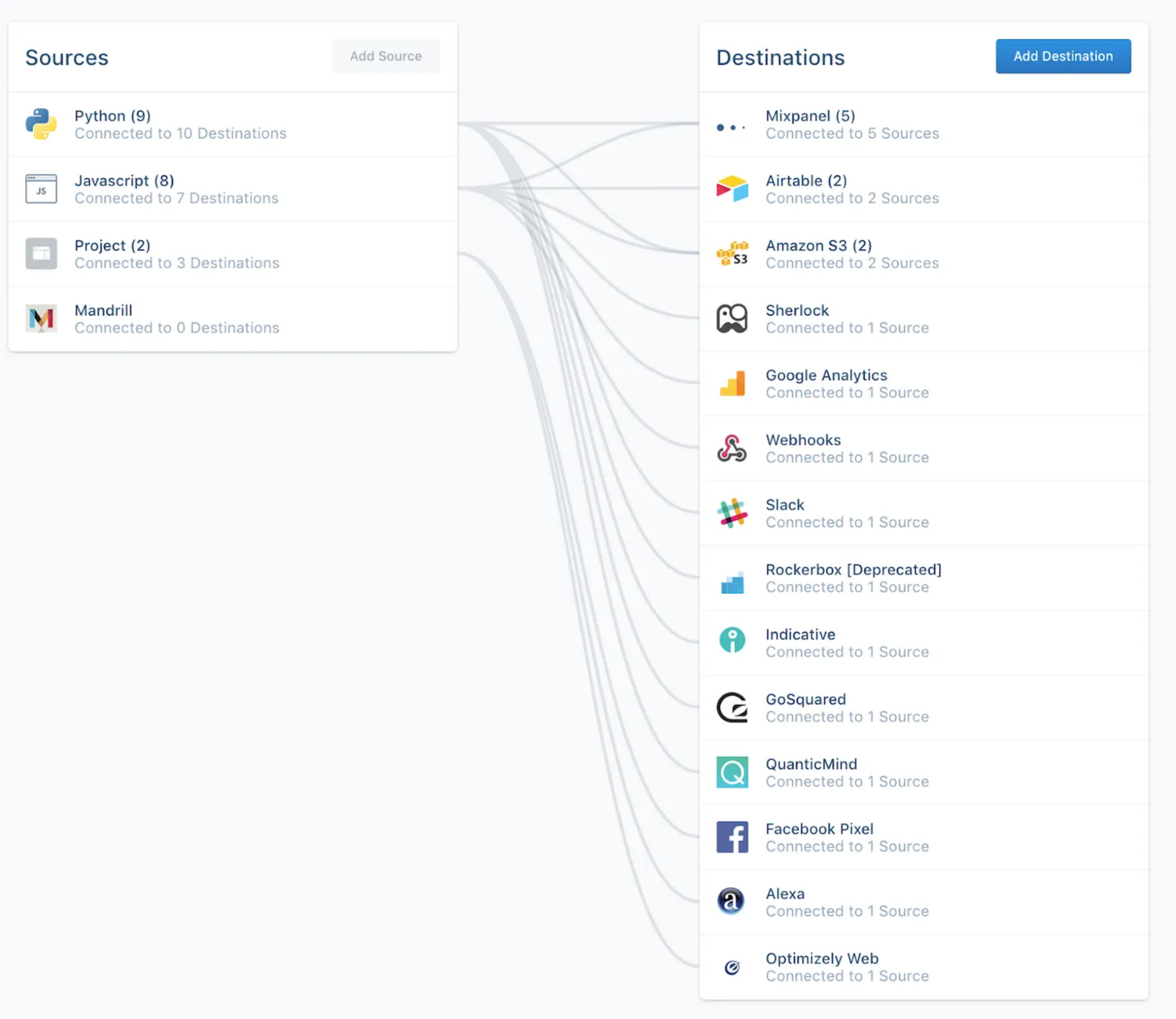
## Tools for product analytics

As you can (hopefully) tell, there's a lot that goes into that graph showing you how many users your product has, from raw events to transformation and visualization. Though most modern companies will do *something* like what's outlined above, there are better and better tools on the market that automate pieces of this process and make them more accessible for non-technical teams.

There is no substitute for a good data team, and there are *always* going to be questions that are too deep for SaaS tools to help you answer. That's why even if you use some of the tools below, chances are you will still have a data team working on the steps outlined above. There are, of course, many many more tools I haven't mentioned here – leave a comment with any of your favorites.

### Segment – instrumentation and routing

Segment helps with multiple parts of the process, from a library for tracking events to routing those events to your warehouse (and other destinations). I wrote more in depth about Segment here.



Sources

Add Source

Python (9)
Connected to 10 Destinations

Javascript (8)
Connected to 7 Destinations

Project (2)
Connected to 3 Destinations

Mandrill
Connected to 0 Destinations

Destinations

Add Destination

Mixpanel (5)
Connected to 5 Sources

Airtable (2)
Connected to 2 Sources

Amazon S3 (2)
Connected to 2 Sources

Sherlock
Connected to 1 Source

Google Analytics
Connected to 1 Source

Webhooks
Connected to 1 Source

Slack
Connected to 1 Source

Rockerbox [Deprecated]
Connected to 1 Source

Indicative
Connected to 1 Source

GoSquared
Connected to 1 Source

QuanticMind
Connected to 1 Source

Facebook Pixel
Connected to 1 Source

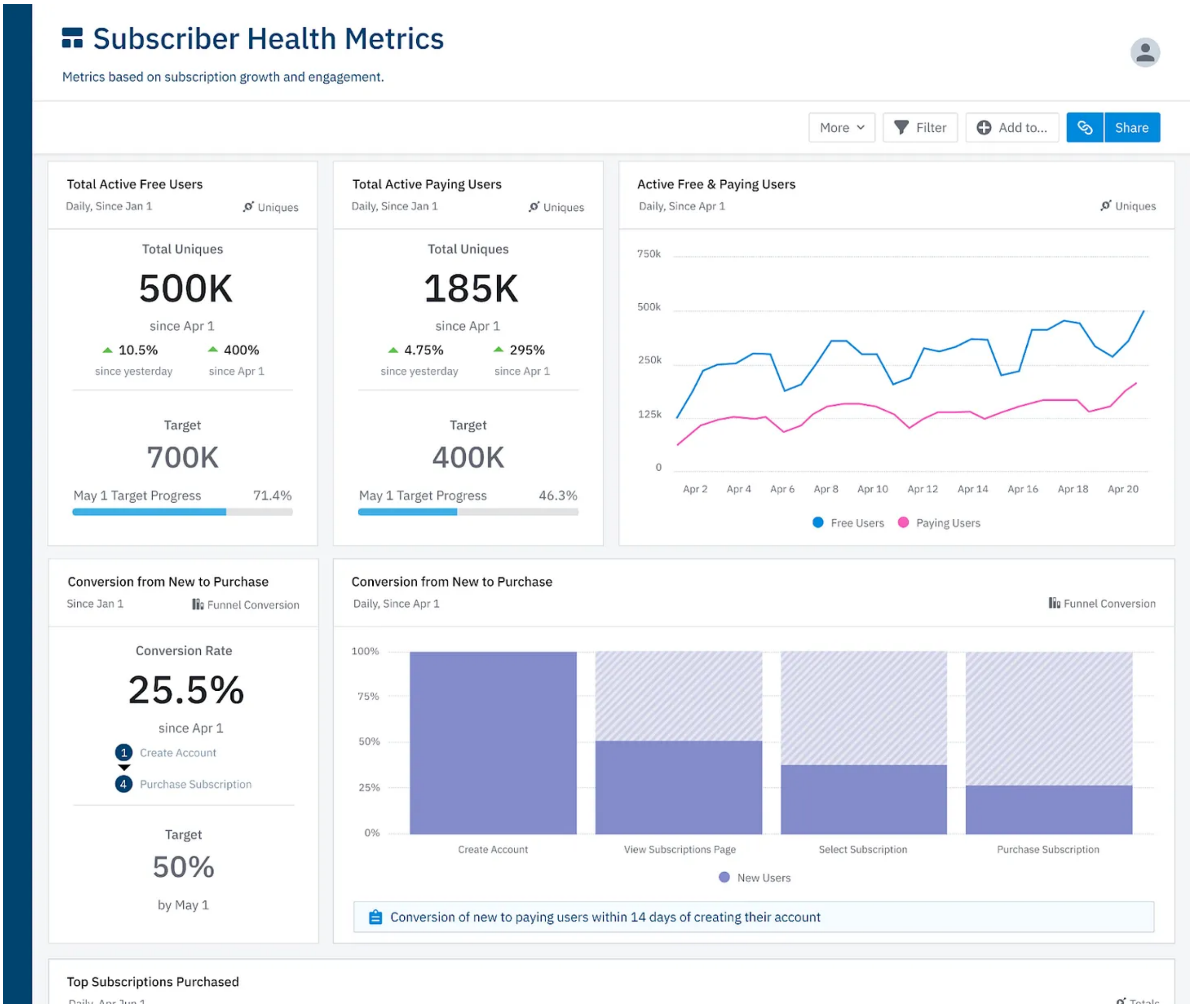
Alexa
Connected to 1 Source

Optimizely Web
Connected to 1 Source

Segment was acquired by Twilio in 2020, so things are a bit up in the air there.

### Amplitude – visualization and exploration

Amplitude is a data visualization platform made specifically for events. You can ingest your product events and build charts, dashboards, etc. It's made mostly for non-technical teams, so you don't need to write SQL to analyze your data; but as mentioned above, using SQL to analyze event streams isn't exactly a walk in the park. Personally, I like using Amplitude for questions on the simpler side of things.



Subscriber Health Metrics

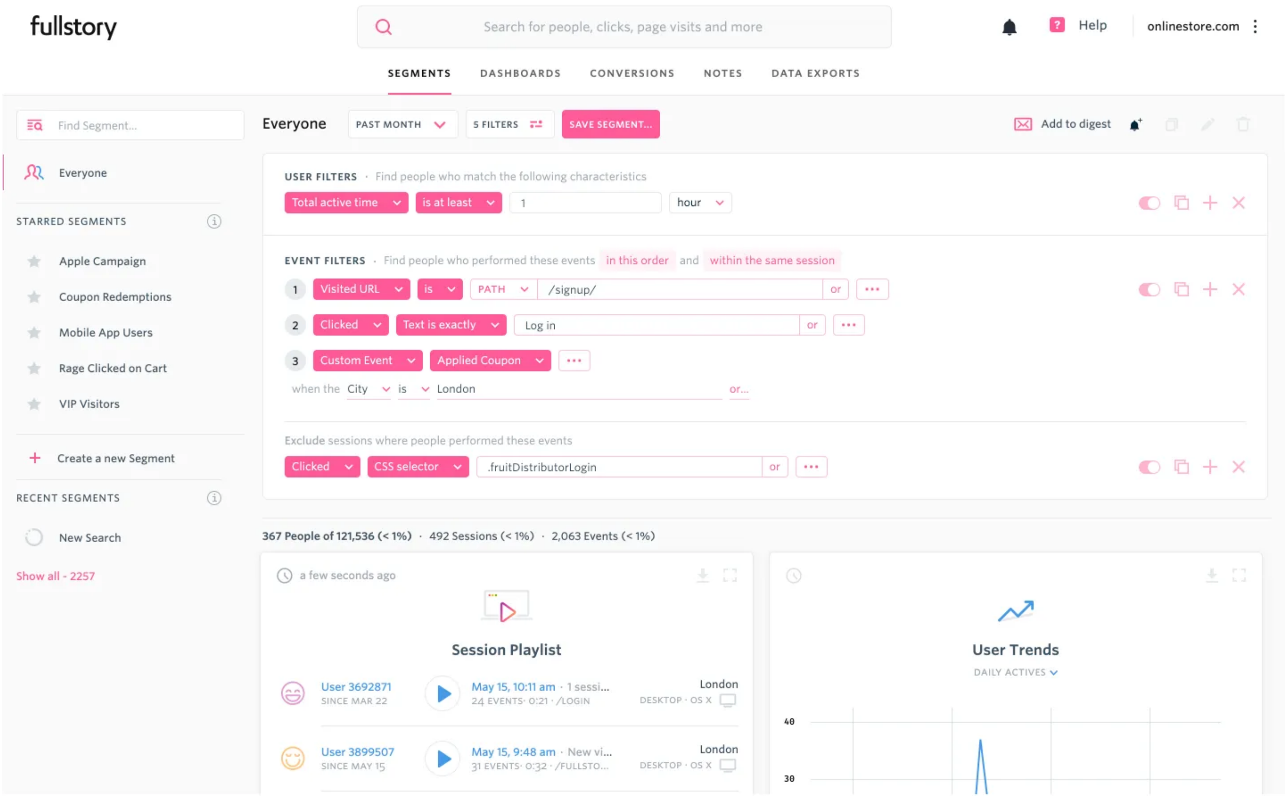
Metrics based on subscription growth and engagement.

Conversion of new to paying users within 14 days of creating their account

The nice thing about Amplitude is it lets you avoid the *transformation* step for certain questions. E.g. If you want to know what the most popular pages visited in your product are, you can just do that in the Amplitude UI, instead of needing your data team to build a table with that information. As mentioned though, for more complex explorations, this is probably not the tool.

### FullStory / HotJar – session recording

FullStory and HotJar are session recording tools: they look at the actions your users take and replay them to you, like you're watching a video of their screen. Logistically, they're not *actually* recording a user's screen – that would be creepy – they're instead looking at how the user's actions impact the HTML of the page, and then rendering what we imagine that probably looked like when the user did it.



USER FILTERS · Find people who match the following characteristics

Total active time | is at least | 1 | hour

EVENT FILTERS · Find people who performed these events in this order and within the same session

1 Visited URL | is | PATH | /signup/ | or

2 Clicked | Text is exactly | Log in | or

3 Custom Event | Applied Coupon

when the City is London or...

Exclude sessions where people performed these events

Clicked | CSS selector | .fruitDistributorLogin | or

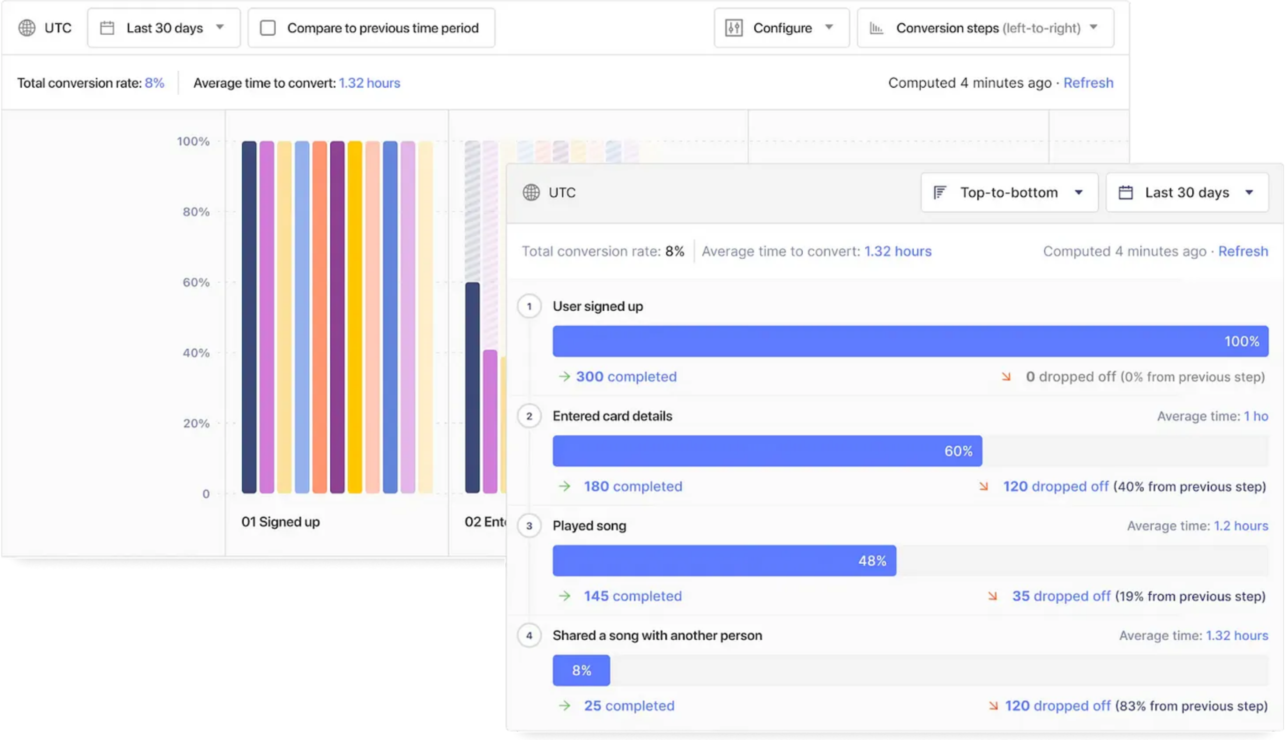
367 People of 121,536 (< 1%) · 492 Sessions (< 1%) · 2,063 Events (< 1%)

Since it’s mostly HTML tagging under the hood, you can filter for things like which page a user visited, what they clicked on, etc.

I have not heard of teams building this functionality on their own, but if you know of any, leave a comment on this post!

### PostHog – full stack, everything

PostHog is probably the most unique tool here, because it obviates most of the need for **the entire** process of building a product analytics practice. Their product extends from event capture all the way to visualization and exploration; they even have functionality around heatmaps and session recording, something you’d normally need to use Fullstory/HotJar for. Their bet is that product analytics should all happen in one place, and that non-technical teams should be empowered to answer *almost all* of their own questions, with little or no help needed.



UTC | Last 30 days | Compare to previous time period | Configure | Conversion steps (left-to-right)

Total conversion rate: 8% | Average time to convert: 1.32 hours | Computed 4 minutes ago · Refresh

UTC | Top-to-bottom | Last 30 days

Total conversion rate: 8% | Average time to convert: 1.32 hours | Computed 4 minutes ago · Refresh

1 User signed up
100%
300 completed | 0 dropped off (0% from previous step)

2 Entered card details | Average time: 1 ho
60%
180 completed | 120 dropped off (40% from previous step)

3 Played song | Average time: 1.2 hours
48%
145 completed | 35 dropped off (19% from previous step)

4 Shared a song with another person | Average time: 1.32 hours
8%
25 completed | 120 dropped off (83% from previous step)

PostHog supports all kinds of visualizations – like funnels, user paths, etc. – specifically built for product analytics. They even have support for feature flags, where you release a new feature to a segment of your user base to test how they react to it. Perhaps the most interesting piece of the puzzle: PostHog is open source, which means it's completely free to run on your own infrastructure (all other products on this list are paid and expensive).

Thanks to the legendary Claire Carroll for help on this post.

11 Comments

Write a comment...

Ferruccio Apr 6, 2022

Great article! There's definetely not enough content around product analytics.

As the founder of a product analytics tool built on top of Segment (https://june.so/) I think it would be relevant to share about our approach to Product analytics.

June allows people to connect their Segment account and automatically generate relevant metrics and reports for Acquisition, Retention, Active usage and Power users

♡ 2 Reply Gift a subscription Collapse •••

Timothé Rey Apr 18, 2022

It would be interesting to see where Google Analytics / Google Tag Manager fit in this space. As a data analyst in the mobile apps/games space I am familiar with product analytics tools and data warehouses as they share a common event-driven architecture.

The problem is that Google Analytics / Google Tag Manager seem weird to me. I really don't understand how their architecture work and it's quite frustrating, as I feel like these are tools with similar purposes, yet they feel so different.

♡ 1 Reply Collapse •••

1 reply

9 more comments...