



How to use Sigma

Welcome, explorer! This guide will help you use Sigma to navigate the vast realms of Qeepsake's data warehouse with ease.

[Qeepsake's Sigma Homepage](#)

- For written learners, there's a detailed walkthrough of the basics of Sigma below.
- For visual learners, I've recorded some video walkthroughs where I go through and explain all of our Sigma folders, and build datasets, charts, and dashboards
 - 📄 [Sigma video walkthroughs](#)
- For those who learn by doing, feel free to login to Sigma as a creator and explore! You can use Philip's account which is in the shared 1password. Please make sure to create a personal folder and save your work in there while you're learning instead of in the team folders.

You can also check out Sigma's help center [here](#). I've written this guide because I think it's easier to learn Sigma for our purposes with Qeepsake data instead of the sample data they provide in [this tutorial](#), but there are tons more resources in the help center than I can provide here. Definitely bookmark it and use it as a reference when you're getting stuck trying to build something!

What is Sigma?

- Qeepsake's business intelligence tool. It's meant to be a central place for information about the business that *all teams* can use.
- A collection of *datasets (tables)* and *workbooks (dashboards)* that store important information about the business and display it in ways that are easy to visualize, monitor, and analyze

- A tool for working with tables in our database like you would work with a spreadsheet! With Sigma you can:
 - See any table in our database as if it were a spreadsheet (you can't edit the cells to change the data in our database, this is a setting that we have turned off)

The screenshot shows the Sigma SQL editor interface. At the top, there is a search bar and navigation tabs for 'onboards', 'ENDORSED', and a star icon. Below this, a menu bar includes 'Overview', 'Columns', 'Links', 'Lineage', 'Worksheet' (which is selected), 'Materialization', and 'Permissions'. The main area displays a table with the following columns: 'User Id', 'Day Onboarded', 'Referral Code', 'Onboarding Channel', 'Oauth Id', 'Utm Campaign', 'Utm Term', 'Utm Source', 'Utm Content', and 'App Version'. The table contains multiple rows of data, including user IDs like 766352, 766034, and 766318, with various referral codes and onboarding channels.

- Write formulas to calculate *new* metrics based off of different columns in your table

The screenshot shows the Sigma SQL editor with a formula bar at the top containing the formula: `Concat([Utm Term], [Utm Source], [Utm Campaign])`. Below the formula bar, there are sections for 'FILTERS' and 'PARAMETERS', both currently empty. The main table view shows the same columns as the previous screenshot, but with a new column added: 'UTM String'. This column contains concatenated values of 'Utm Term', 'Utm Source', and 'Utm Campaign' for each row, such as 'acq_int_abc_motherhoodfacebook' for the first row.

- Summarize data over different dimensions (like counting users onboarded by date or journals by user ID)

[onboards/day_onboarded]

FILTERS ⓘ +		BASE COLUMNS			
No filters		123 User Id	📅 Day Onboarded	abc Referral Code	abc Onboard
PARAMETERS ⓘ +		766352	2021-12-14 00:00:00	Add New Column	paid_soc
No parameters		766034	2021-12-14 00:00:00	Rename Column	intention
		766318	2021-12-14 00:00:00	Duplicate Column	paid_soc
		766350	2021-12-14 00:00:00	Add Description	mobile
		766425	2021-12-14 00:00:00	Column Details...	mobile
		766387	2021-12-14 00:00:00	Hide Column	paid_soc
		766422	2021-12-14 00:00:00	Hide Other Columns	mobile
		766440	2021-12-14 00:00:00	Delete Column	intention
		766309	2021-12-14 00:00:00	Add Filter...	mobile
		766372	2021-12-14 00:00:00	Group Column	paid_soc
		766324	2021-12-14 00:00:00	Sort Ascending	paid_soc
		766336	2021-12-14 00:00:00	Sort Descending	paid_soc
		766340	2021-12-14 00:00:00	Sorting...	organic
		766358	2021-12-14 00:00:00		mobile
		766363	2021-12-14 00:00:00		paid_soc
		766181	2021-12-14 00:00:00		paid_soc
		766487	2021-12-14 00:00:00		mobile
		766499	2021-12-14 00:00:00		organic
		766407	2021-12-14 00:00:00		paid_soc
		766449	2021-12-14 00:00:00		mobile
		766455	2021-12-14 00:00:00		paid_soc

fx CountDistinct([User Id])

FILTERS ⓘ +		LEVEL 1 ⋮	BASE COLUMNS			
No filters		📅 Date	123 Users Onboarded	123 User Id	abc Referral Code	abc Onboarding Ch...
PARAMETERS ⓘ +		2021-12-14 00:00:00	181	766323	IntOrg	intentional_organic
No parameters				766397	IntOrg	intentional_organic
				766499	OFDA	organic
				766448	OFDF	mobile
				766494	OFDF	mobile
				766181	OFDB	paid_social
				766403	OFDF	mobile
				766369	OFDF	mobile
				766329	OFDB	paid_social
				766381	OFDF	mobile
				766383	OFDF	mobile
				766371	OFDD	google
				766492	OFDD	google

- Link different sources of data together in an intuitive way (imagine using "vlookup" but way easier)

onboards

User Id	User Id (Conversions)	Day Onboarded	Referral Code	Onboarding Channel
766352	null	2021-12-14 00:00:00	OFDB	paid_social
766034				
766318				
766350				
766425				
766387				
766422				
766440				
766309				
766372				
766324				
766336				
766340				
766358				
766363				
766181				
766487				
766499				
766407				
766449				
766455				

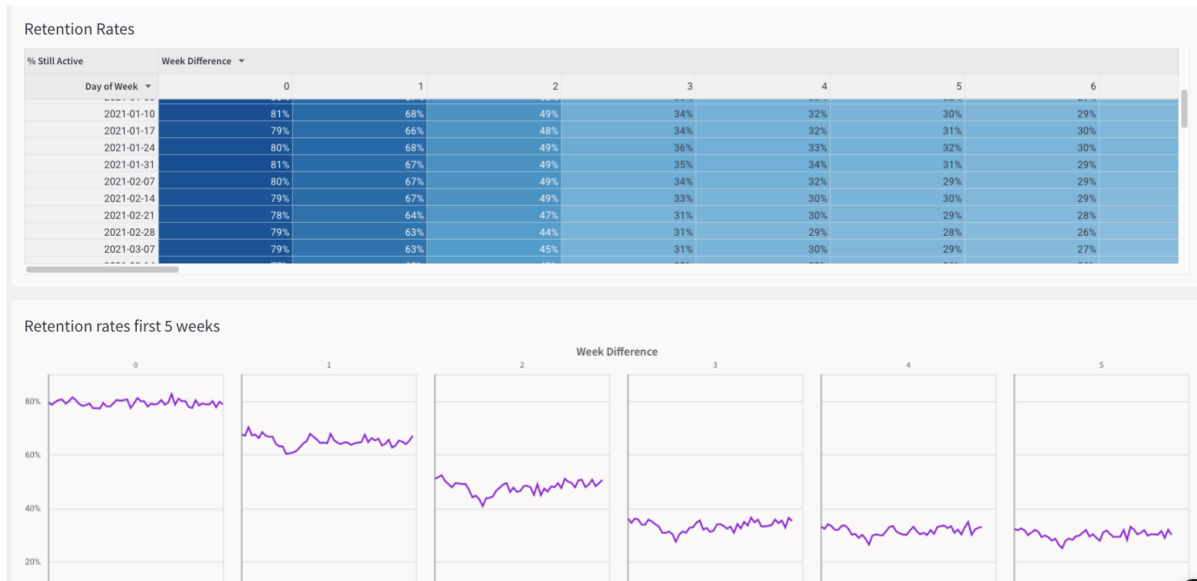
LINKED RECORD

conversions

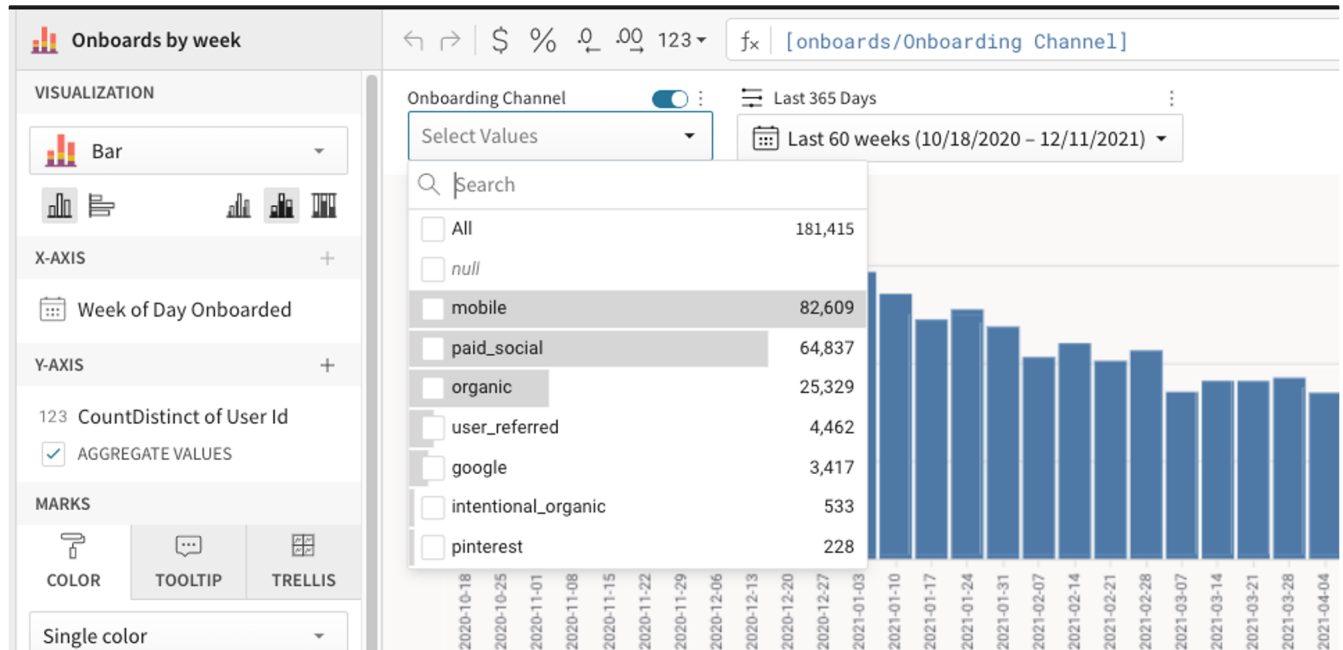
766034

- User Id: 766034
- Subscription Name: Qeepsake Plus
- Payment Provider: ios_iap
- Day Subscription Starts: 2021-12-14 00:00:00
- Is First Paid Sub: 1
- Day of Conversion: 0
- Discount Amount: 0
- Revenue Transacted: 47.88
- Day Onboarded: 2021-12-14 00:00:00
- Referral Code: IntOrg

- Create charts based off of your tables



- Filter any column in a table or a chart to drill down into a specific subset of your data.



How do I get started?

- Sign up for a "viewer" account by asking one of the administrators (Ryan or Philip) to invite you via email.
- Check out the "Workspaces" folder and the "All Endorsed Datasets" folder within it
 - **Workspaces** is where all team-shared data should live. There are folders for each business team, plus some additional folders like Financial modeling, All Endorsed Datasets, and Teamwide.
 - **All Endorsed Datasets** is essentially our "database" in Sigma. Data team curates this folder to make sure it contains the tables each team would need to do analysis and build dashboards, while maintaining a structure that makes sense and documenting it [here](#)
- Log into an "explorer" or "creator" account via the team's shared 1password (we have limited accounts here because we have to pay for them). Right now, the accounts we have are:
 - Creators (can create & edit folders, workbooks and datasets, can invite other users): Ryan, Philip
 - Explorers (can break down the data within workbooks and save different versions): Jeff, Duncan

- Try creating your own datasets and workbooks!
 - If you have a business analysis you need to do and it's important to do it quickly and correctly, set aside some time with Ryan to do a shared session where you'll build it together.
 - If not, you should still spend some time exploring our data and answering your own questions. **This is valuable time to the business — Being comfortable in Sigma will make you a more effective decision maker and a more informed team member!** What are you curious about? Here are some questions you could try to answer:
 - | How many users have we added per year of Qeepsake's existence?
 - | What is the average number of journals a Qeepsake user has?
 - | What's the most common journal name?
 - | How many questions has the average user gotten over their lifetime? How many have they answered?
 - | What is the average price of a subscription counting discounts from sales vs. not counting them?
 - | What % of users answer a question in their Xth (1st, 6th, 18th, etc) week on Qeepsake?
 - | How many questions are answered with a response that contains the word "love"?
 - | How many times does the average user tap the "new entry" button in the app during their first week?
 - | Who has more journals on average - People whose youngest journal is less than a year old, or people whose youngest journal is 1-2 years old?

Okay, I'm poking around, but still don't really know what I'm supposed to be doing. Can you help me out?

Sure thing!



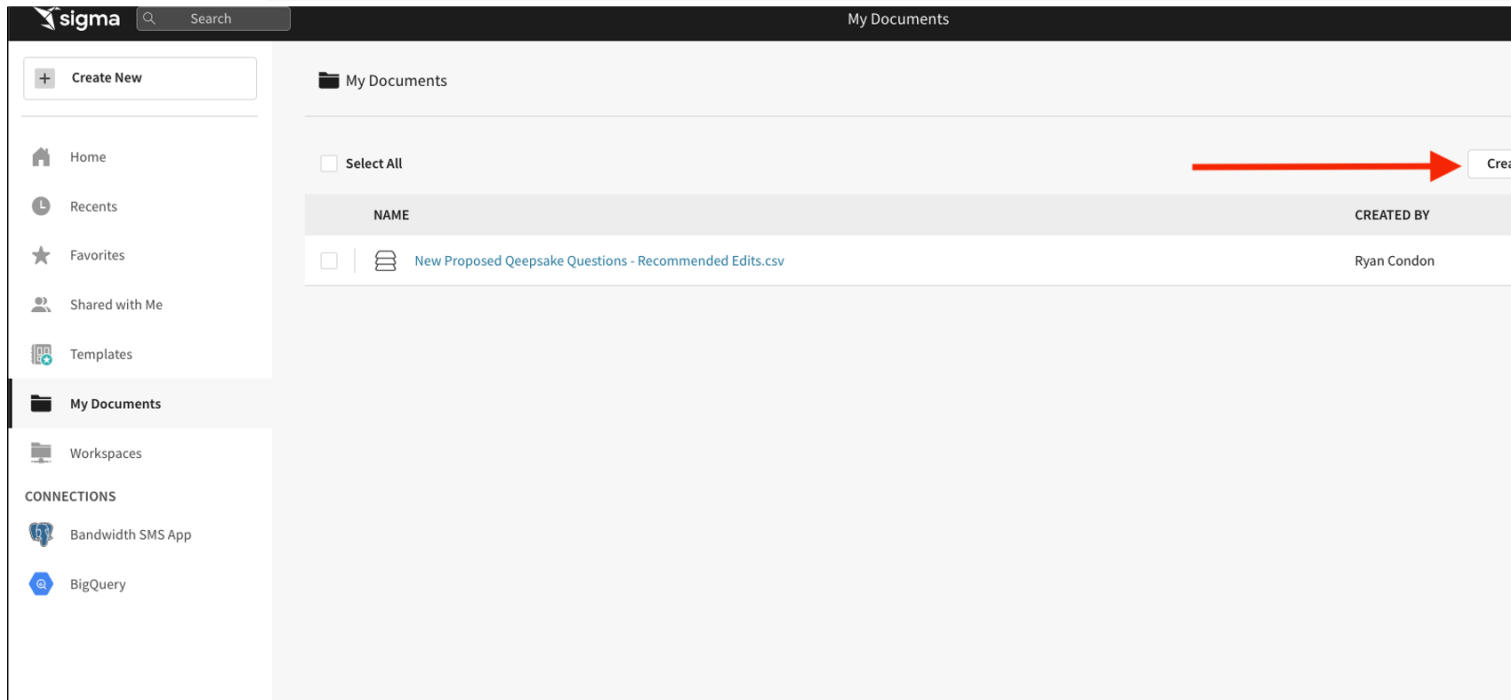
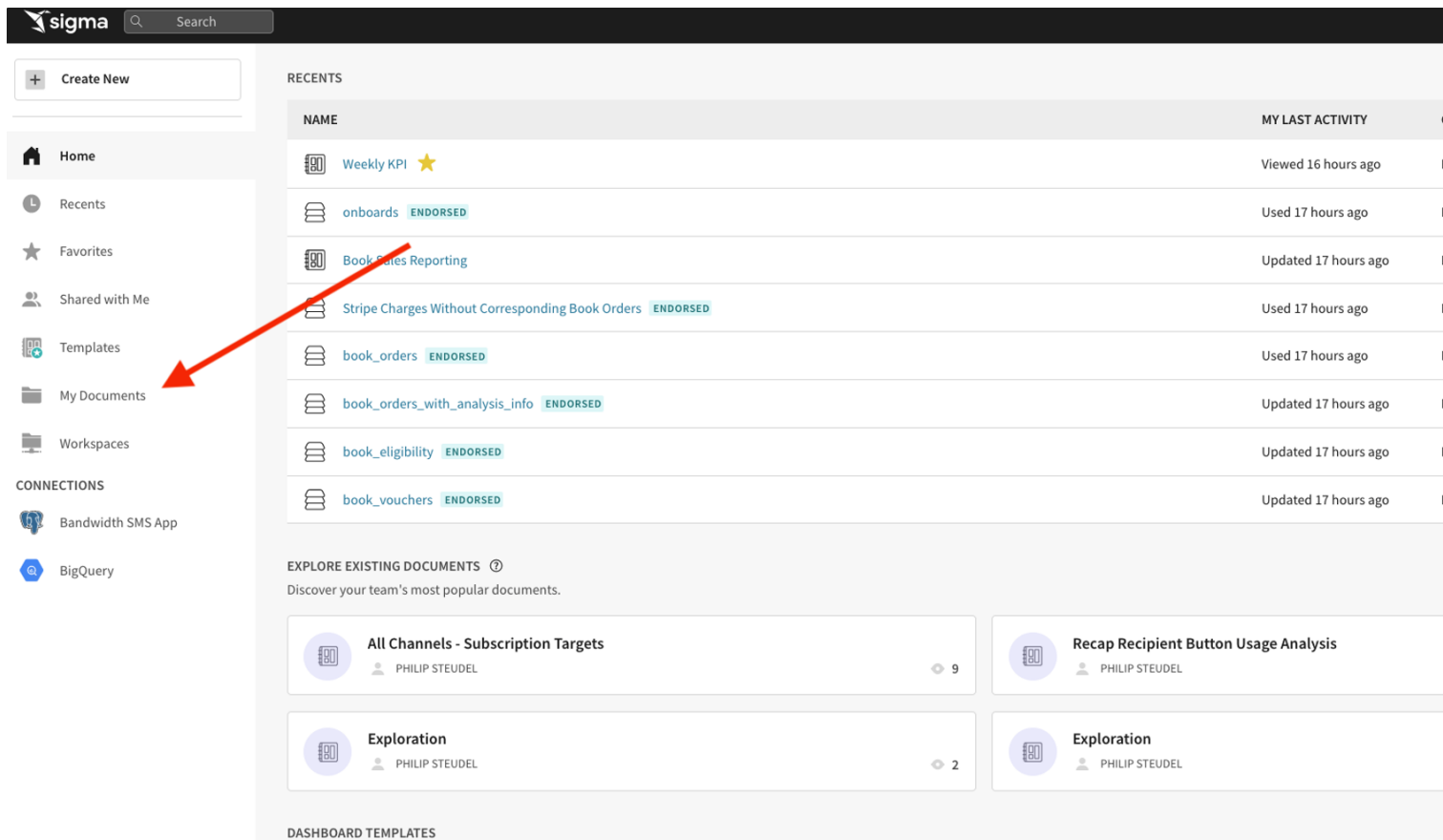
First thing, you might want to watch some of the walkthrough videos:

[📄 Sigma video walkthroughs](#)

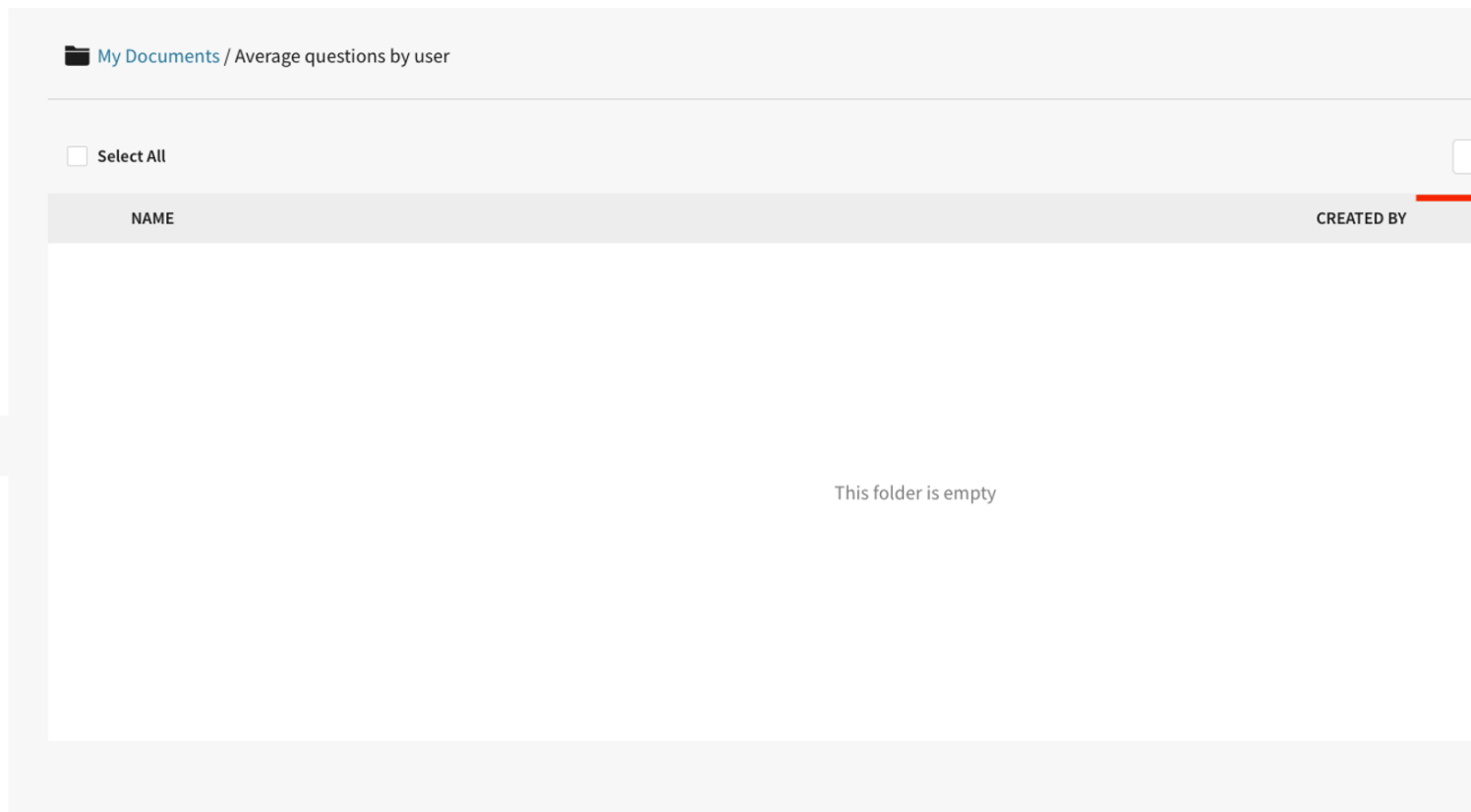
But if you're more of a written learner, let's go through some of the basic functions in Sigma and how to do them. Let's pretend we're answering one of the questions above:

"How many questions has the average user gotten over their lifetime? How many have they answered?"

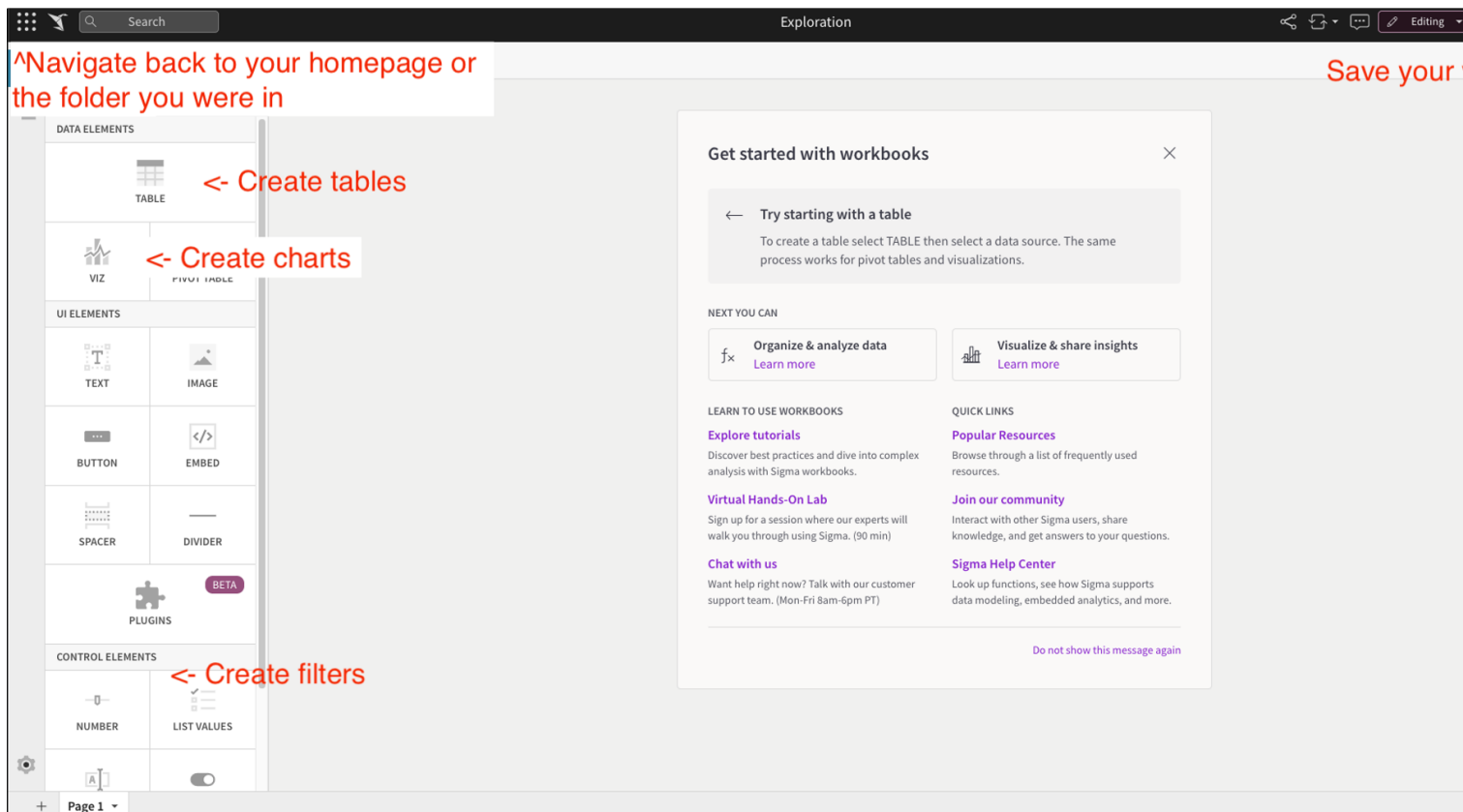
And we're starting on our creator account's homepage. The first thing we're going to do is go to "my documents" and create a folder to store our analysis:



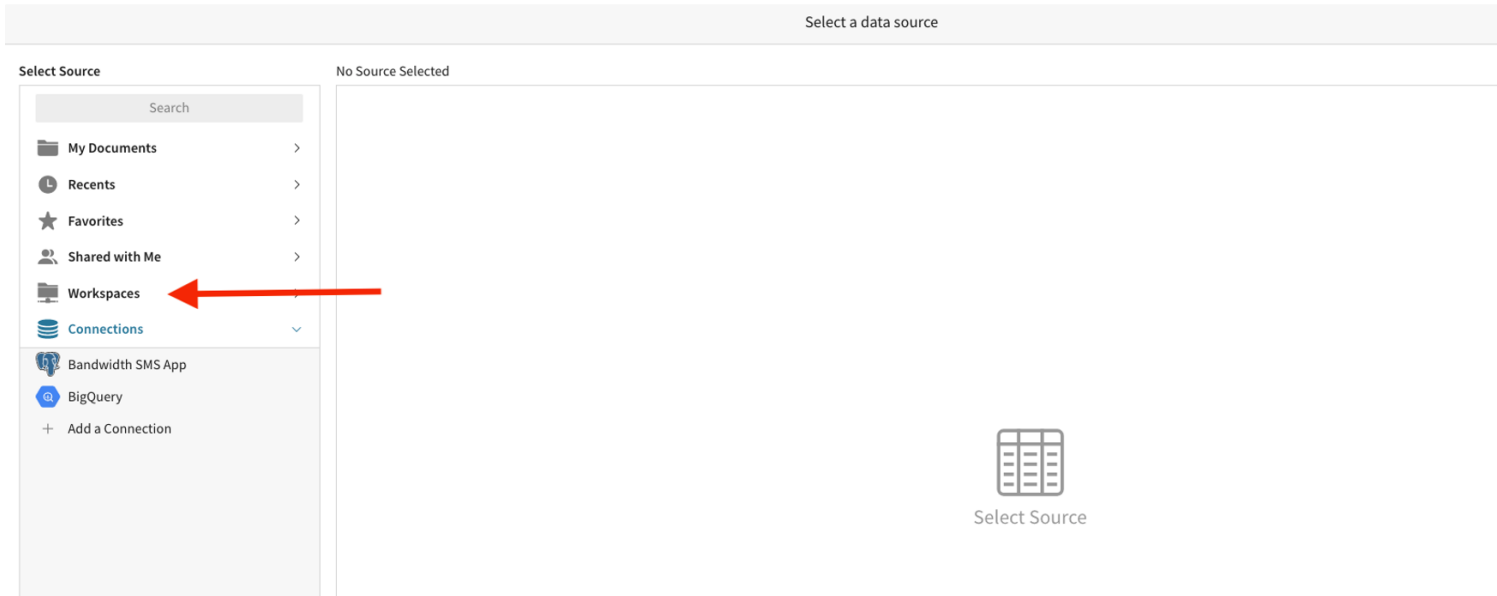
Then we'll create a new workbook in this folder. Workbooks (AKA dashboards) are generally where your charts and final analysis will live. If we need to create a custom table to build our analysis off of, we'd create a dataset in this same folder for that purpose.



This is the workbook opening screen. I've labeled a few key places in the UI where most of the important things happen:

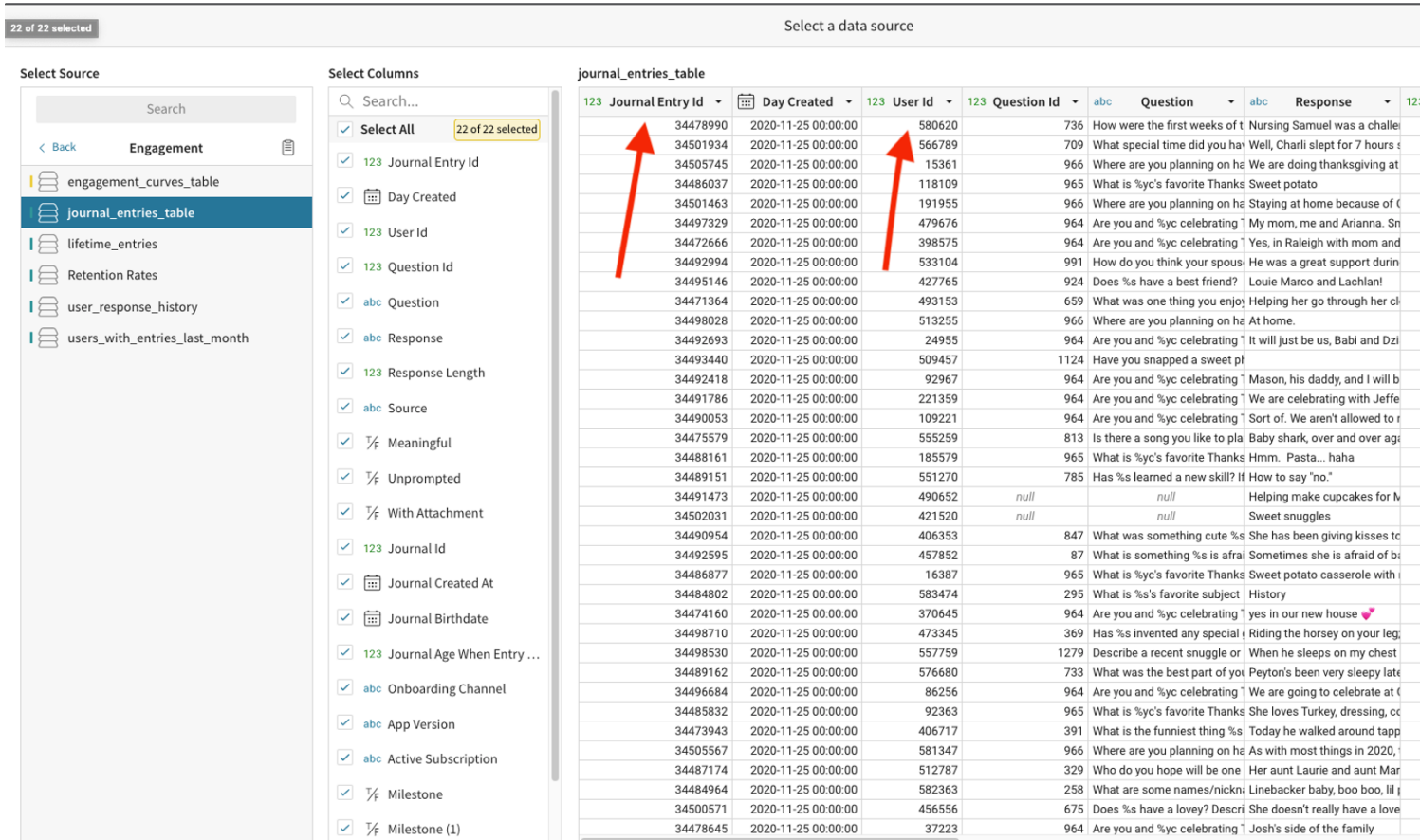


Usually it's good to start by creating a table. Let's bring in the journal entries table. Sigma gives you an option to select from your existing tables, write SQL to pull directly from our database, or upload a CSV. This table already exists (technically it's saved as a dataset in Sigma, we'll use the terms "table" and "dataset" pretty interchangeably), so we select "Tables and datasets". That opens up this screen:

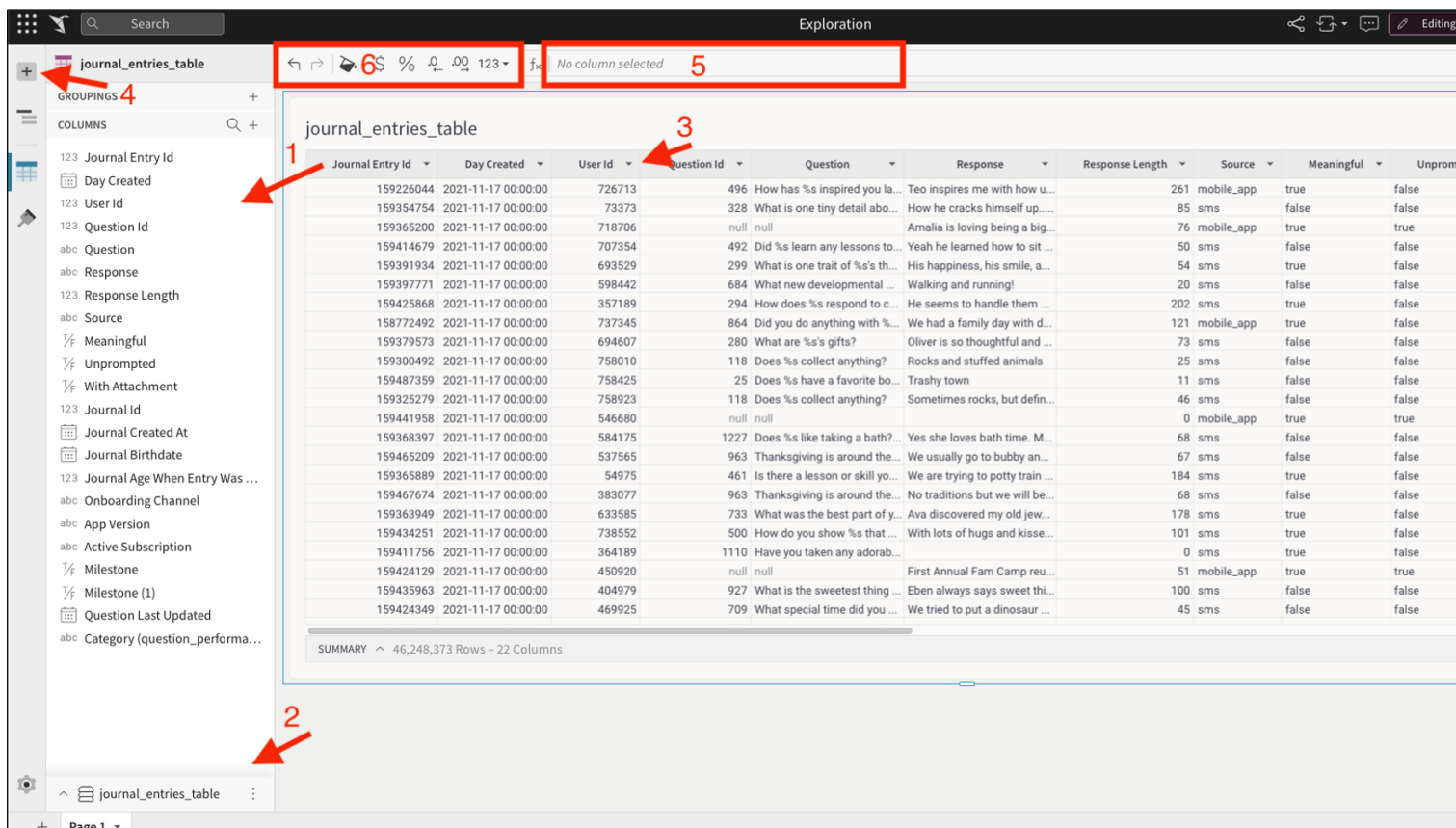


You will **almost always** want to navigate to **Workspaces**→**Analytics database** so that you're working with existing data that the data team has already curated. The journal entries table is in the "Engagement" folder, but if I didn't know that I could look it up here (even if I was looking for something as vague as "entries", I could control-F and search the page)

Selecting the table lets us preview it before we decide it's the one we want:



The first column in every table should (almost always) be some sort of ID column that represents the basic item that each row in this table represents. In this case, each row of this table is a unique journal entry created by a user. We'll be able to summarize this data by users, since we see that there's also a column identifying which user ID created each entry. So we click "done" and see this table back in our workbook:



Some key areas to notice:

1. You can view the columns in the table either in the table itself or on the left while you have the table selected.
2. You can view the underlying dataset that your table is pulling from in the bottom left. **The new table you created in your workbook is not the same as the dataset! It is a "child object" of it. What that means is: any edits you make to your new table will not be applied to the dataset itself, but edits to the dataset will apply to the table. For example, you can choose to only bring in certain columns from the dataset into your table, and any formatting you do on those columns will not affect the original columns in the dataset.**
3. You can select/edit any individual column by clicking on its dropdown.
4. You can still add new objects to the workbook with the "+" button in the top left.
5. You can write custom formulas to create new columns (or edit existing ones) in the formula bar up top
6. Just like in Excel, there's a formatting bar as well with buttons for common formatting options.

Let's create a new column in our table with a custom formula. Remember the question we're trying to answer is

"How many questions has the average user gotten over their lifetime? How many have they answered?"

So let's create a column to see if the user answered the question. To create a new column, we click the "+" button next to columns in the top right of the bar on the left where all the columns are listed, and select "new column". It will show up at the end of our table as "Calc" (we can rename it by double-clicking on the column name). We can write a formula for this column. We'll count it as a response if a user sends in a text response, or even if they only send in a photo. Something isn't a response if and only if both of those columns are "Null".

Here is the function index for Sigma - If you want to do a certain kind of calculation on a column but aren't sure how, start here!

Our function will be

```
isnull(response) and withattachment = false
```

This will return "true" if the response column is null and the withattachment column is false.

The screenshot shows the Sigma Analytics interface. At the top, a formula bar contains the expression: `f_x | IsNull([Response]) and [With Attachment] = False`. A red arrow points to this formula. Below the formula bar, a table titled "journal_entries_table" is displayed. The table has columns: Journal Birthdate, Journal Age When Entr..., Onboarding Channel, App Version, Active Subscription, Milestone, Milestone (1), and Question Last Updated. The table contains multiple rows of data. On the left side, a sidebar shows a list of columns, with a red arrow pointing to the "Calc" option at the bottom.

Let's look at what we can do with our new column by selecting the dropdown menu (I renamed it "Responded?" but it should actually be "Didn't Respond?" because a true value indicates a question that WASN'T responded to)

The screenshot shows the dropdown menu for a column named "Responded?". The menu items are numbered 1 through 7:

1. SORT
2. Filter
3. Add new column
4. Duplicate column
5. Delete column
6. Group column
7. Column details...

Other options visible in the menu include "Add column via lookup..." (with a BETA badge), "Rename column", "Hide column", and "Transform".

1. We can sort the column in ascending or descending order (this will sort the entire table by this column)
2. We can filter the table by the values in this column
3. We can always add a new column via this dropdown menu
4. We can duplicate the column (especially useful if you have written a long, complicated formula that you want to iterate on but don't want to have to write all over again)
5. We can hide or delete the column (hiding just removes it from visibility in the table in the workbook, deleting it removes it from the table altogether (*but not from the underlying dataset!*))

6. We can group the entire table by this column. More on this soon!

7. We can examine the values in the column by looking at "column details". Let's do that so we can see what our formula is showing!

Column Details ✕

About	Summary		
$\frac{1}{f}$ Responded?	True	0.0%	0
f_x IsNull([Response]) and [With Attachment] = False	False	100.0%	46,248,373
	Nulls	0.0%	0
	Row Count	100.0%	46,248,373

[Learn more about Columns](#)

It looks like there isn't a single "true" value in the entire 46-million row table. That's crazy! Well, it turns out that we've already filtered out non-responses from this table since we don't count them as "journal entries". You can see this in the [table documentation](#):

Journal Entries

This is a cleaned-up version of the `journal_entries` table in our [Heroku](#) database that to only non-null entries from users who are currently on one of our standard plan types additional calculated fields that flag useful properties of the journal entry.

- Journal entry ID (used for count distincts)
- User ID
- Question ID (null = Unprompted)
- Day the entry was created
- Rendered text (This column is buggy but will sometimes display the text of the question asked)
- Response (The user's response)

It's always important to remember that in Sigma, you're working with curated data, not a direct representation of our database. Always check the documentation if you're seeing something you don't expect! And feel free to reach out to a data team member if you have questions about why data is filtered in a certain way, or whether it can be changed.

So it looks like we won't be able to answer the first part of our question,

"How many questions has the average user gotten over their lifetime?"

with this table. However, we should be able to answer part 2, **How many have they answered?".** All we need is to find the total number of users, and the total number of journal entries lifetime, then divide the second by the first!

Whenever we want to find data that summarizes an entire table, what we'll want to do is GROUP the table by a column with only one value in it. That way, any metrics we calculate based off of that column's one value will be based off of the entire table! Luckily, we already have such a column. Let's group by it!

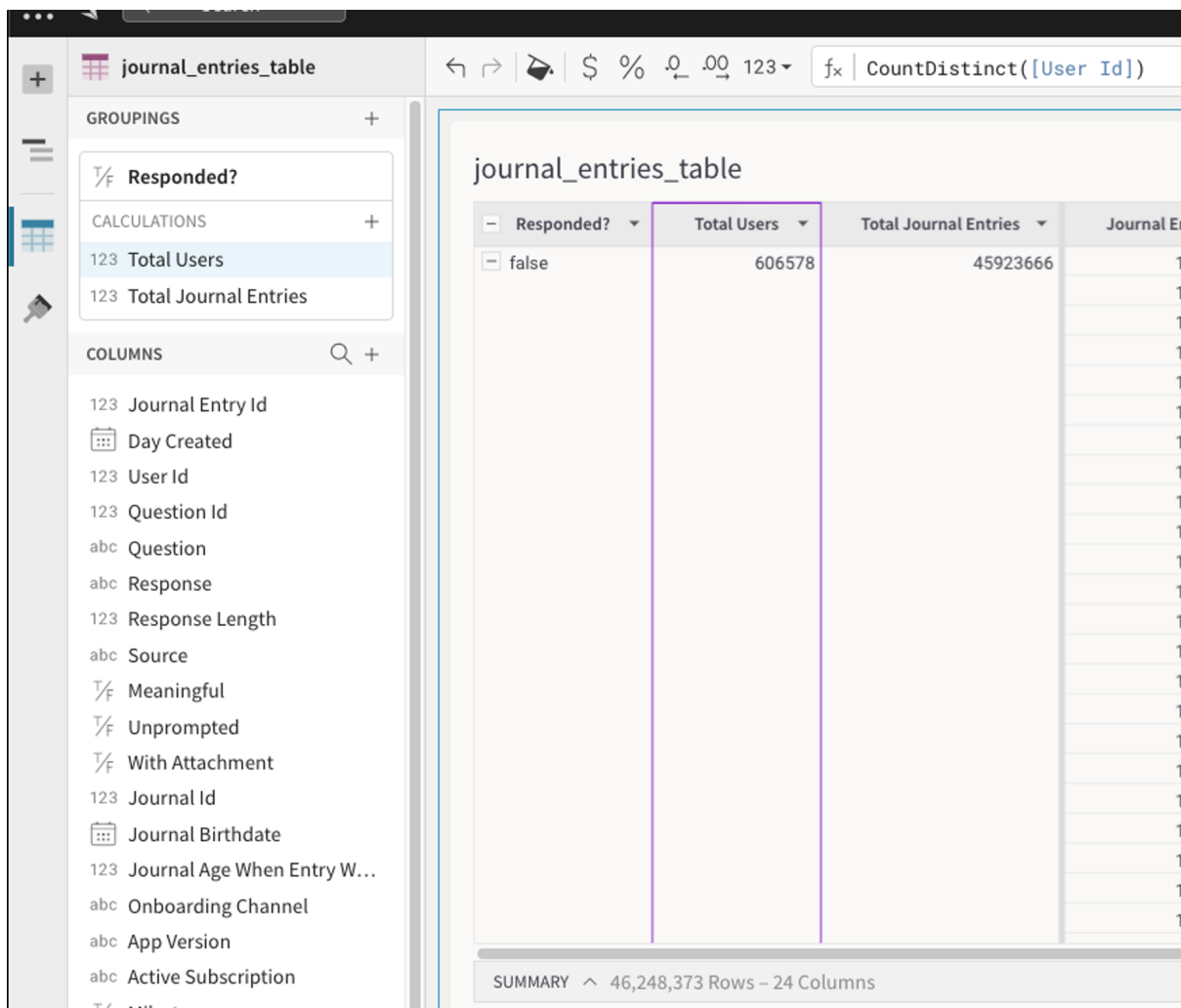
The screenshot shows the Sigma SQL editor interface. On the left, the 'journal_entries_table' is configured with a 'GROUPINGS' section containing 'Responded?'. Below it are 'CALCULATIONS' and 'COLUMNS' sections. The 'COLUMNS' section lists various fields like 'Journal Entry Id', 'Day Created', 'User Id', etc. The main table view shows a table with columns: Responded?, Journal Entry Id, Day Created, and User Id. The 'Responded?' column is highlighted in purple and contains the value 'false'. A red arrow points to the 'Responded?' column in the GROUPINGS section, and another red arrow points to the 'false' value in the table.

Responded?	Journal Entry Id	Day Created	User Id
false	133244679	2021-05-03 00:00:00	6
false	133192433	2021-05-03 00:00:00	6
false	133190454	2021-05-03 00:00:00	1
false	133133787	2021-05-03 00:00:00	6
false	133152603	2021-05-03 00:00:00	6
false	133103157	2021-05-03 00:00:00	3
false	133162729	2021-05-03 00:00:00	6
false	133171410	2021-05-03 00:00:00	4
false	133210486	2021-05-03 00:00:00	6
false	133220268	2021-05-03 00:00:00	1
false	133194474	2021-05-03 00:00:00	6
false	133177643	2021-05-03 00:00:00	6
false	133213559	2021-05-03 00:00:00	6

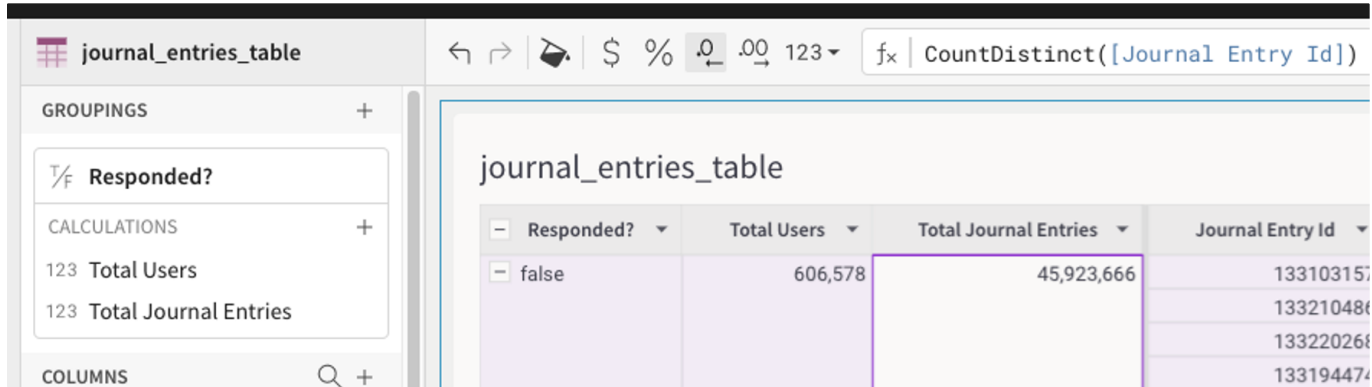
Now we can see that all the way on the left of the table, we've taken this column and added it to a new "grouping level" in the table (separated by the gray divider line). Any columns we add at this level will be calculated as aggregations for each value in the "Responded?" column (in this case, there is only one value, "false", so any calculation will be over any row that has value "false", which is the entire table)

We can add new columns to calculate how many user IDs show up in this table, and how many journal entries there are. (You could also find the total journal entries by seeing how many rows are in the table, because each row corresponds to a single entry).

```
countdistinct(user id) countdistinct(journal entry id)
```



We can also format the columns so that they show commas in the thousands places between digits by selecting the dropdown next to the function box, and picking "number", then reducing the decimal places twice.



Now we have total users and total journal entries! We can also make a calculation based off the new columns we made. So we divide entries by users to get average entries per user.

There we go! *How many have they answered?* Looks like 76 on average.

Responded?	Total Users	Total Journal Entries	Avg entries per user	Journal Entry Id	Day Created	User Id	Question Id
false	606,578	45,923,666	76	34492418	2020-11-25 00:00:00	92967	964 A
				34488161	2020-11-25 00:00:00	185579	965 W
				34489151	2020-11-25 00:00:00	551270	785 H
				34491473	2020-11-25 00:00:00	490652	null ni
				34492595	2020-11-25 00:00:00	457852	87 W
				34486877	2020-11-25 00:00:00	16387	965 W
				34484802	2020-11-25 00:00:00	583474	295 W
				34498710	2020-11-25 00:00:00	473345	369 H
				34498530	2020-11-25 00:00:00	557759	1279 D

But let's try to answer the question in a different way, that might be slightly more useful. It would be great to see not just the average number of journal entries per user, but the *distribution*. To do this, we'll need to calculate the number of entries each user has in this table, and then graph each of those numbers. Luckily, this is incredibly easy to do in Sigma! All we have to do is GROUP again by user ID, and we can sum up the number of entries that each user has. So we select the dropdown on the user ID column, and group:

Responded?	Total Users	Total Journal Entries	Avg entries per user	User Id	Journal Entry Id	Day Created	Question
false	606,578	45,923,666	76	61	4895849	2017-07-13 00:00:00	
					7521164	2017-12-23 00:00:00	
					735334	2016-06-30 00:00:00	
					22238587	2019-09-23 00:00:00	
					5449626	2017-08-19 00:00:00	
					13642878	2018-10-02 00:00:00	
					430284	2016-05-13 00:00:00	
					26983871	2020-04-08 00:00:00	
					26983832	2020-04-08 00:00:00	
					430345	2016-05-13 00:00:00	
					8491181	2018-02-13 00:00:00	
					2670884	2017-01-27 00:00:00	
					18235398	2019-04-10 00:00:00	
					6557495	2017-10-28 00:00:00	
					1530661	2016-09-28 00:00:00	

Sidenote: At this point I realized my "responded?" column should really say "true" instead of "false", so I added a not() surrounding the entire formula. The will change in the following screenshots!

Notice that the grouping by user ID doesn't remove the grouping by "Responded?" — It goes *under* it, as a subset of the original grouping. Since "Responded?" is grouping the entire table, that doesn't particularly matter here, but for example, if we added another grouping by "day created", what we would actually be grouping by is the combination of both *user* and *day*.

Next, we'll add a "journal entries for this user" column at our "user" grouping level.

journal_entries_table

GROUPINGS

- Responded?

CALCULATIONS

- Total Users
- Total Journal Entries
- Avg entries per user

123 User Id

CALCULATIONS

- Journal entries for this user

COLUMNS

- Journal Entry Id
- Day Created
- Question Id
- Question
- Response
- Response Length
- Source
- Meaningful
- Unprompted
- With Attachment
- Journal Id
- Journal Birthdate

Responded?	Total Users	Total Journal Entries	Avg entries per user	User Id	Journal entries for this ...	Journal Entry Id
true	606,578	45,923,666	76	61	991	42
						15099
						43
						382
						43
						43
						43
						2221
						1548
						43
						877
						43
						43
						43
						2698
						12
						43
						2698
						43
						1396
						73

SUMMARY ^ 46,248,373 Rows - 26 Columns

User ID 61 happens to be Jeff, and we can see that he has 991 lifetime entries. Impressive!

It would be nice if we didn't have to see all the journal entries associated with each user, and instead could just see a list of all our user IDs. Sigma lets us do that! We just have to collapse the lowest grouping level by clicking on the "-" sign next to user ID:

journal_entries_table

Responded?	Total Users	Total Journal Entries	Avg entries per user	User Id	Journal entries for this ...	Journal Entry Id	Day Created	Question Id	Question	Response
true	606,578	45,923,666	76	+ 61	991					
				+ 62	1356					
				+ 64	94					
				+ 66	72					
				+ 67	3					
				+ 70	414					
				+ 71	16					
				+ 72	19					
				+ 74	107					
				+ 75	142					
				+ 76	17					
				+ 77	165					
				+ 78	29					
				+ 79	11					
				+ 80	8					
				+ 81	40					
				+ 82	15					
				+ 84	33					
				+ 85	1					
				+ 86	58					
				+ 87	31					

SUMMARY ^ 606,578 Rows - 26 Columns

Notice that you can still see the columns of the underlying table (and you can still do things like group by them or filter by them), but the underlying data has been wrapped up so that we can see things at the level of user IDs.

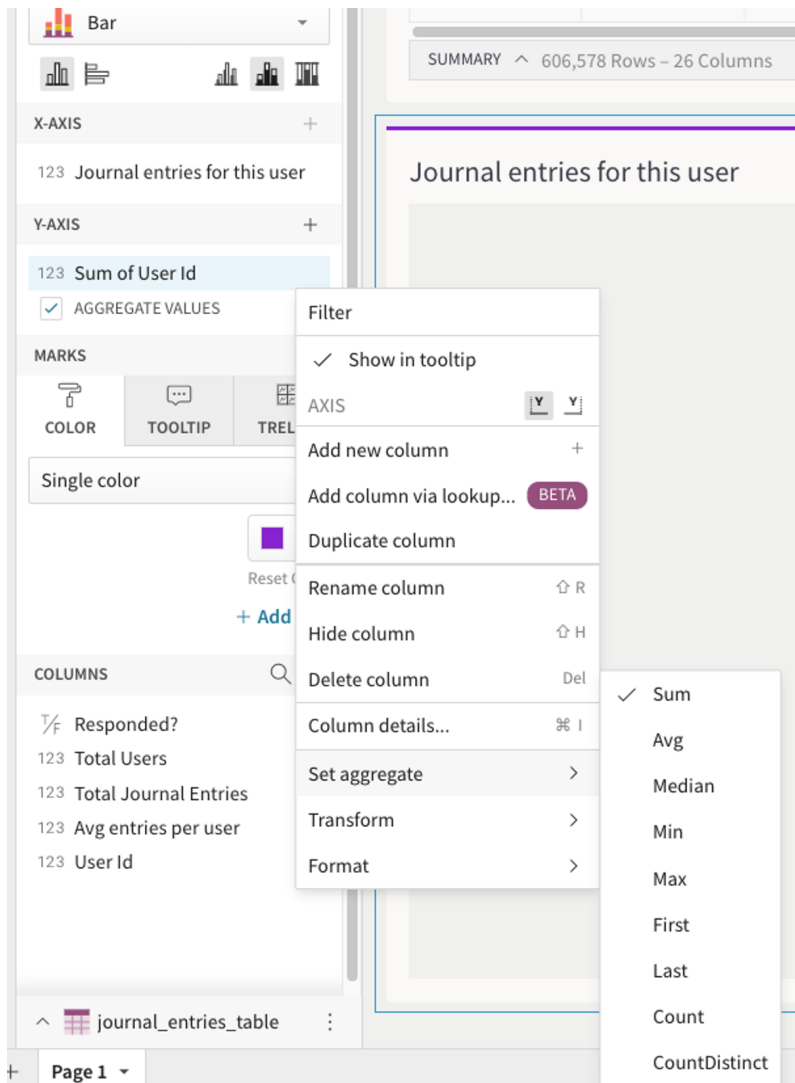
Awesome. Now let's graph it! To create a visualization based off of a table, we just click the "+chart" icon in the top right of the table.

Responded?	Total Users	Total Journal Entries	Avg entries per user	User Id	Journal entries for this ...	Journal Entry Id	Day Created	Question	Question	Response
true	606,578	45,923,666	76	+ 61	991					
				+ 62	1356					
				+ 64	94					
				+ 66	72					
				+ 67	3					
				+ 70	414					
				+ 71	16					
				+ 72	19					
				+ 74	107					
				+ 75	142					
				+ 76	17					
				+ 77	165					
				+ 78	29					
				+ 79	11					
				+ 80	8					
				+ 81	40					
				+ 82	15					
				+ 84	33					
				+ 85	1					
				+ 86	58					
				+ 87	31					

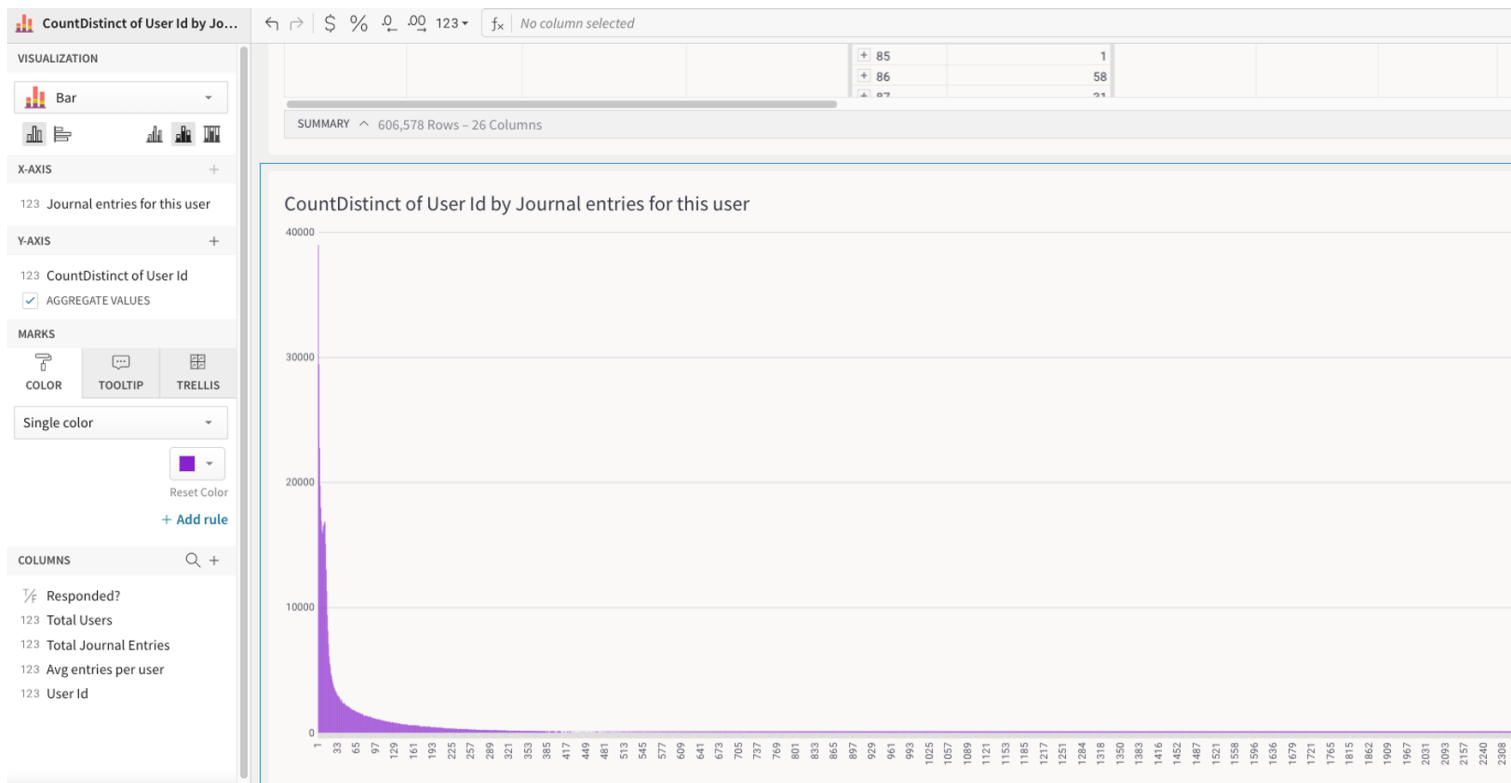
This creates a "child element" of the table that will reflect the values in the table as the table gets changed, filtered, etc.

1. This is where you can choose which type of visualization you're creating. For our purposes, we're fine with a bar chart to show a distribution of integer values.
2. These are the columns from your original table. You can drag them up into...
3. The X and Y axes of your chart! You can drag existing columns into these areas to display them on the graph, but you can also write new columns using formulas.
4. The chart formatting section can be found by clicking on the paintbrush.

We want to see the distribution of user IDs based on how many entries they've made. So our X-axis will be "journal entries for this user", and our Y-axis will be counting the number of user IDs with that number of entries, like so:

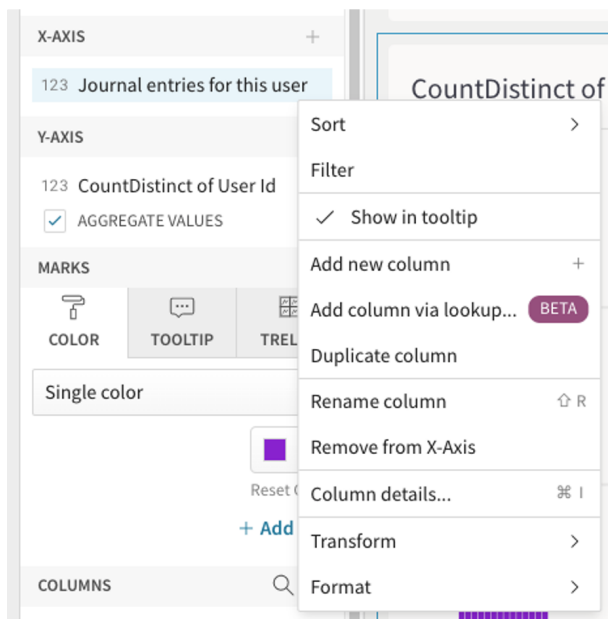


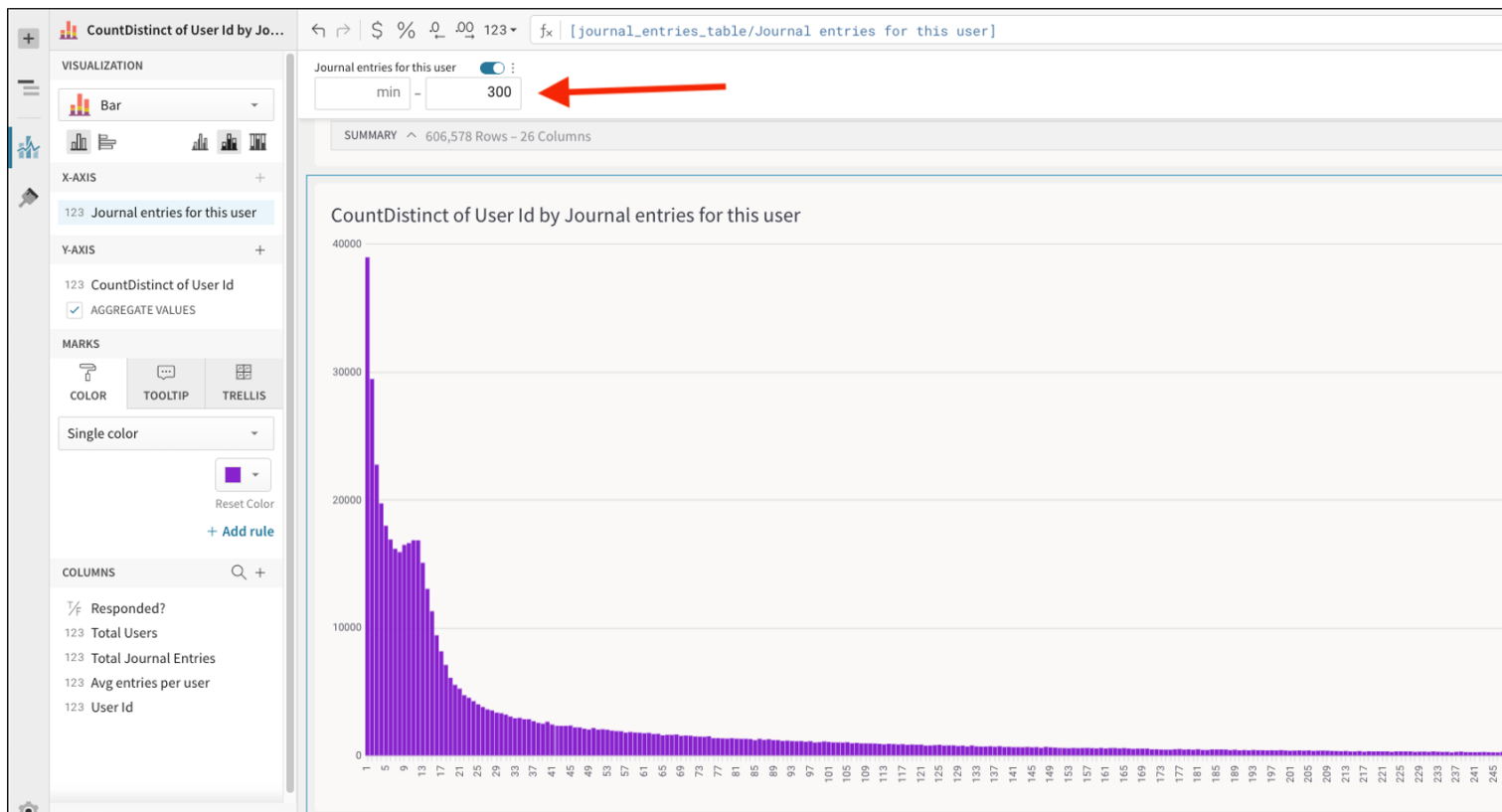
Notice that when we first drag "user ID" into the Y-axis, Sigma tries to automatically choose an aggregation for that variable. It's chosen sum, but we want countdistinct, so we select the dropdown and set aggregate → countdistinct. That gives us this chart:



This is getting closer — It is technically the distribution of journal entries per user, but there's a few ways we could make it more readable. We could filter out our "long tail" by limiting to only values less than, say, 300 entries. Or we could "bucket" the data into clumps of 10 entries. I'll show how to do both.

To filter on any of our variables, we just have to select the dropdown for that variable. So I'll filter on "journal entries for this user" and set the max to 300:

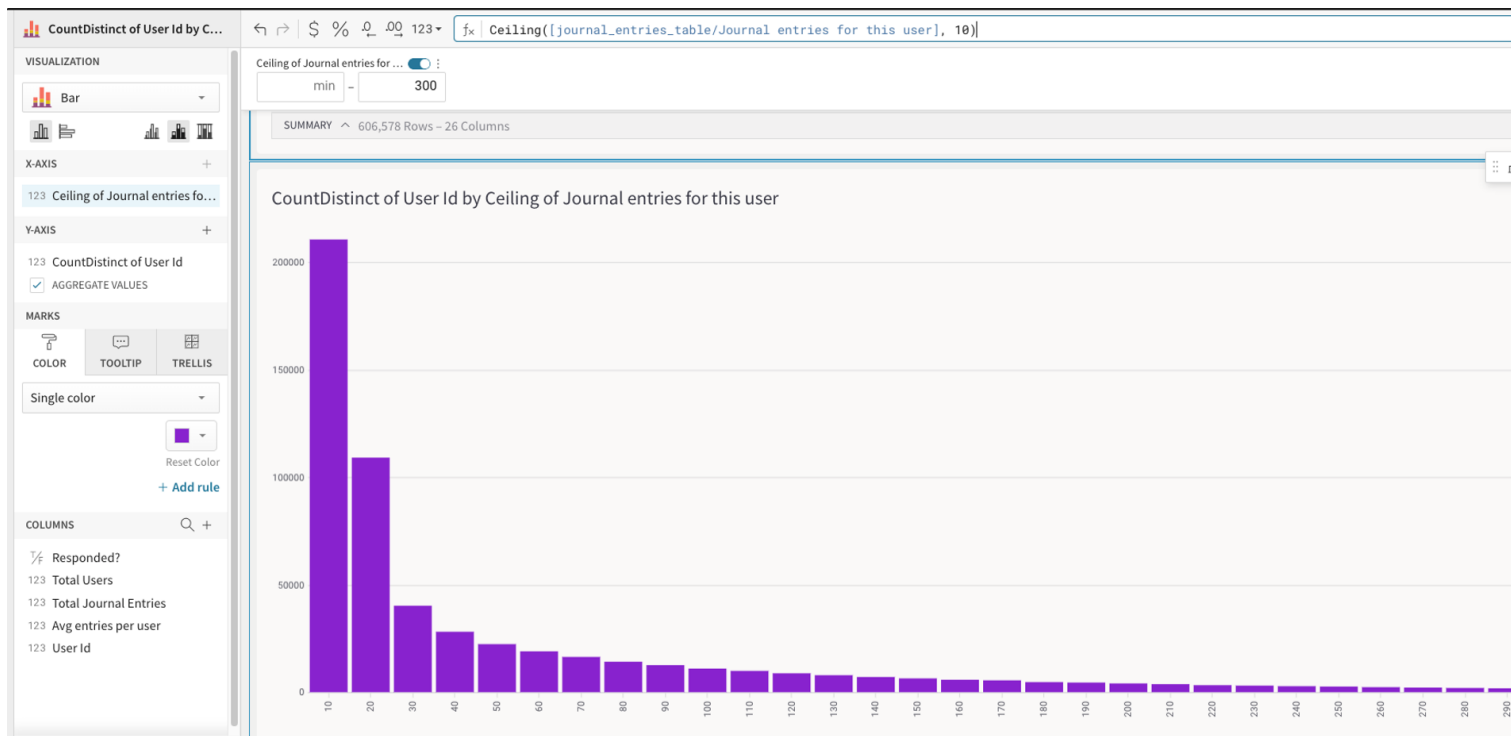




There, that looks a bit more manageable. We can see a slight bump in the curve around 13-14 entries, which makes sense if a lot of users stop after their free trial. But let's make it look even nicer. Instead of bucketing by each single increment up, let's bucket by 10s. For this, I'll use a formula called ceiling:

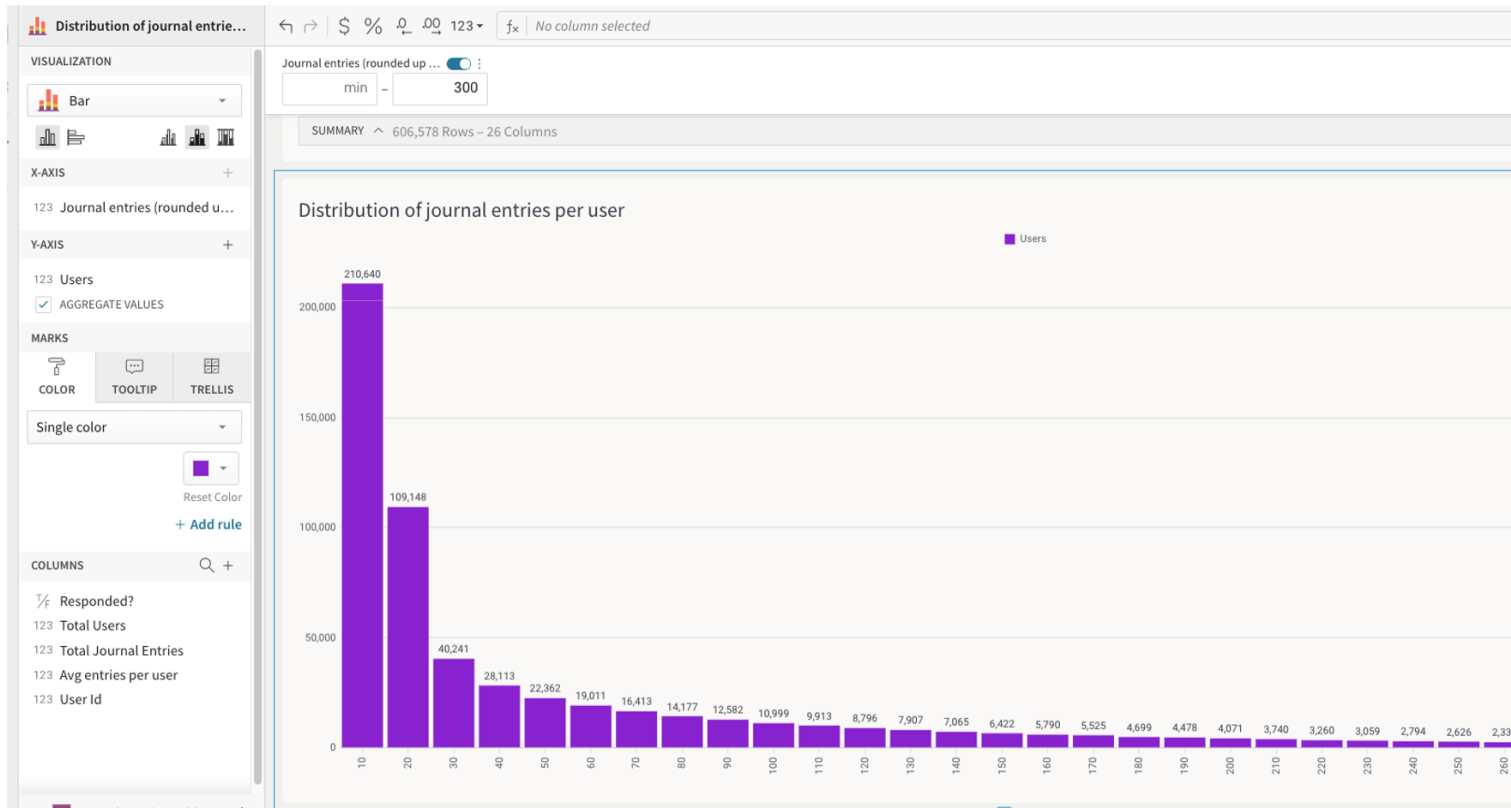
```
Ceiling([journal_entries_table/Journal entries for this user], 10)
```

This will set our Y-axis to the closest 10s value above the number of journal entries a user has. In other words, the "10" represents all users who have between 1-10 entries.



There we go, a nice steep engagement curve! (Although of course, we'd love for it to be flatter 😊)

Finally, we can title the chart and the axes whatever we want by double-clicking into their names, add a legend and data labels, and call it done!



What if we wanted to know the exact same thing, except for only photo entries? Do we have to rebuild this chart from an entirely new table? Am I asking leading questions to highlight the magic of dynamic filtering? I am!

If we want to filter this chart to only photo entries, all we have to do is filter the table it's based off of! I'll go back to the table and click on the dropdown for the "With Attachment" column (even though the data in it is hidden because we've rolled up to the user level, this still works!)

The screenshot shows the Sigma Analytics interface. On the left, there's a sidebar with 'GROUPINGS', 'CALCULATIONS', and 'COLUMNS'. The main area displays a table named 'journal_entries_table' with columns: 'ion', 'Response', 'Response Length', 'Source', 'Meaningful', 'Unprompted', 'With Attachment', 'Journal Id', 'Journal Birthdate', and 'Journal Age When Entr...'. The 'With Attachment' column is highlighted with a purple border, and its dropdown menu is open, showing options like 'SORT', 'Filter', 'Add new column', 'Add column via lookup...', 'Duplicate column', 'Rename column', 'Hide column', 'Delete column', 'Group column', 'Column details...', 'Transform', and 'Conditional formatting'. Below the table is a bar chart titled 'Distribution of journal entries per user' showing a single bar for 'Users' with a value of 210,640.

And there we go... We can see that the distribution of photo entries is even steeper.

