How To Calculate Customer Retention Rate — A Practical Approach

November 21, 2019

Retention is arguably the new marketing gold, but it's not always easy to calculate it. With this simple Python script, that will change!

If you don't have a way to measure Customer Retention, how will you know if you're "leaking" customers? (photo: Vryheid Herald)

So many metrics, so little time...

I get it. You are a busy person with no time to read the intros! Unfortunately, I do need to give a brief explanation about what we are measuring, since there are so many metrics around customer retention and loyalty! Contrary to most of my other articles, the coding here will be short, but I need to make sure the concepts are clear because **it is really easy to mess up the calculations.**

> I will go over the basics and the reasons to measure Customer Retention, and then finish with a Python script that will deliver the retention tables

If you already follow me, you know I usually write about Python projects and share the code in the end. This will be no exception! However, this time I decided to go for a project which some may actually apply to their own professional reality or even their own businesses. Let me know how it worked out for you in the comments!

The single most important thing you need to understand before attempting to calculate anything is that **each business has its own nuances when measuring retention**. For an online store, you may want to know how many customers keep purchasing your items from month to month. For a SaaS company, you may want to consider how many customers keep an active paying subscription. On the other hand, if you talk to an analyst working at a social platform like Reddit, or Facebook, they will probably be more focused on user visits instead of purchases. Same if you have a freemium App like a game, for instance.

It's not that we do not care about purchases and other money-related metrics, but **the main idea here is to measure if our users are coming back**! That is the ultimate way to know if you have a good product, right? *(hint: it's a rhetorical question!)*

These slight differences between industries also exist when measuring Recency, which is the *time passed since the last interaction* with a user. The term "interaction" can be a lot of things... But that is a topic for a whole new article about RFM analysis!

There are a few names I will throw at you throughout the article and you should know what they mean beforehand:

- **Churn Rate (or Attrition Rate)** — the percentage of customers that abandon your product over a period of time
- **Retention Rate** — the percentage of customers that continue using your product over a period of time
- **Cohort analysis**: Noticed the "period of time" above? Do you want to measure retention from day to day, week to week, month to month? In our example, I will measure **daily** retention. A cohort analysis is probably the best tool to analyze Retention and Churn. It's a specific type of table that will split the users into "buckets" (cohorts), and each cohort will contain users that signed up on the same day/week/month/etc. That said, each cohort refers to one period of time (one day, in our example).

Why is Customer Retention important anyways?

Evaluates the quality of your acquisition strategy

See the picture on top, with a leaking pipe? If you are not measuring customer retention, you may have a [customer] leakage problem. You may be losing existing users at a catastrophic rate while you spend your precious dollars acquiring new ones with AdWords or Facebook ads, or whatever acquisition strategy you may have.

If you do not keep your users coming back, you are essentially **burning your money away**.

Let that sink in.

> While you are spending your precious dollars acquiring new users, your product is failing at keeping them engaged, and that will kill your business in the long run.

My business model is solid because I get so many new users!

Maybe that was a little harsh... But it serves to make a point. **You need to know your retention rate if you want to make informed decisions about your business model**. You will also want to know how much is the lifetime value of your customers so you can evaluate if you are spending too much acquiring them, but that is out of the scope of this article!

Think about this GIF when you plan your next AdWords or FB ads campaign to get new users!

It allows you to test what works and what doesn't

Of course, using retention related metrics is not only useful to evaluate your acquisition campaign. You can experiment with several things, like A/B testing a signup offer, or an app feature. Give A to the users who sign up January, and give B to users who sign up in February. Measure the retention rate and check if there is a campaign that worked better than the other. Rinse. Repeat.

This is why Facebook introduced the News Feed to humankind. Do you remember when we only saw what our friends shared and that was more than enough? Then one day, Facebook decided to roll out the News Feed. Civil unrest ensued, which ironically ended up proving

Facebook was right to make that move. **They knew people were spending more time online, engaging with other people, and coming back more often**.

They did the testing and their data showed the retention was higher. So they went for it, regardless of public outcry. Online games perfect their missions to maximize user engagement (another fancy name for retention).

The same principle is valid for user journeys. If you decide you want to offer new users a specific journey (this is probably more related to Gaming apps), you can check which roadmap works best. You can see which specific journey kept the users longer in your app/platform. And if you cross the retention metric with some sales metric, you can also see if the users that stay longer are spending more or less money than before. And then, obviously, you can target them with campaigns, and evaluate their performance.

But I digress…

I could go on with a few other points that can benefit from knowing what your product retention rate is, but these two should be enough to convince you! The bottom line is that for some people, **Retention is king**. And this should be logical because if you keep your customers longer, they will eventually spend more money on your product, which means increasing their lifetime value.

I get the idea… but where do we start?

Well, we start by getting some data!

I wanted to go beyond the classic textbook example of an eCommerce store and its monthly sales from 2015, so I went ahead and created a fictitious dataset for a fictitious app. Imagine this app is a freemium game where users can play for free, but also spend some money.

> In order for this to work in your own context, you need to extract daily data about which users were active in the app. You don't need to extract it each day, but you must have a table that shows who was active, on each day of your defined timeframe.

> If you choose to analyze monthly retention, replace the word "*daily*" and "*day*", by "*monthly*" and "*month*" in the sentence above. Or "*weekly*" and "*week*" and so on. You are the master of your own analysis!

You can find my dataset in a *csv* format at the end of the article. Each row is related to a player's activity in one day. You can have duplicate player names if they were active in several days (hopefully!). But let me clarify what the columns are:

- **username**: unique user id
- **signup_date**: the date the player signed up
- **ref_date**: the date of the activity
- **money**: how much did the player spend on that ref_date
- **time**: how many minutes the user played on that ref_date
- **country**: categories like this allow you to segment players and calculate retention per country, for instance.

Here's a snippet of the file we will use:

	A	B	C	D	E	F
1	username,signup_date,ref_date,money,time,country					
2	U10000,01/10/2019,01/10/2019,2,89,Canada					
3	U10000,01/10/2019,07/10/2019,0,35,Canada					
4	U10000,01/10/2019,09/10/2019,2,4,Canada					
5	U10000,01/10/2019,10/10/2019,1,5,Canada					
6	U10000,01/10/2019,12/10/2019,1,17,Canada					
7	U10000,01/10/2019,15/10/2019,0,16,Canada					
8	U10001,01/10/2019,01/10/2019,1,134,Canada					
9	U10001,01/10/2019,02/10/2019,2,60,Canada					
10	U10001,01/10/2019,03/10/2019,0,55,Canada					
11	U10001,01/10/2019,04/10/2019,1,77,Canada					
12	U10001,01/10/2019,05/10/2019,0,117,Canada					
13	U10001,01/10/2019,06/10/2019,0,23,Canada					
14	U10002,01/10/2019,01/10/2019,0,110,United States					
15	U10002,01/10/2019,04/10/2019,1,51,United States					
16	U10002,01/10/2019,11/10/2019,1,81,United States					
17	U10002,01/10/2019,12/10/2019,0,21,United States					
18	U10003,01/10/2019,01/10/2019,2,112,United States					
19	U10003,01/10/2019,02/10/2019,1,25,United States					
20	U10003,01/10/2019,03/10/2019,1,12,United States					
21	U10003,01/10/2019,04/10/2019,1,137,United States					
22	U10003,01/10/2019,05/10/2019,1,109,United States					

Finally, some Python!

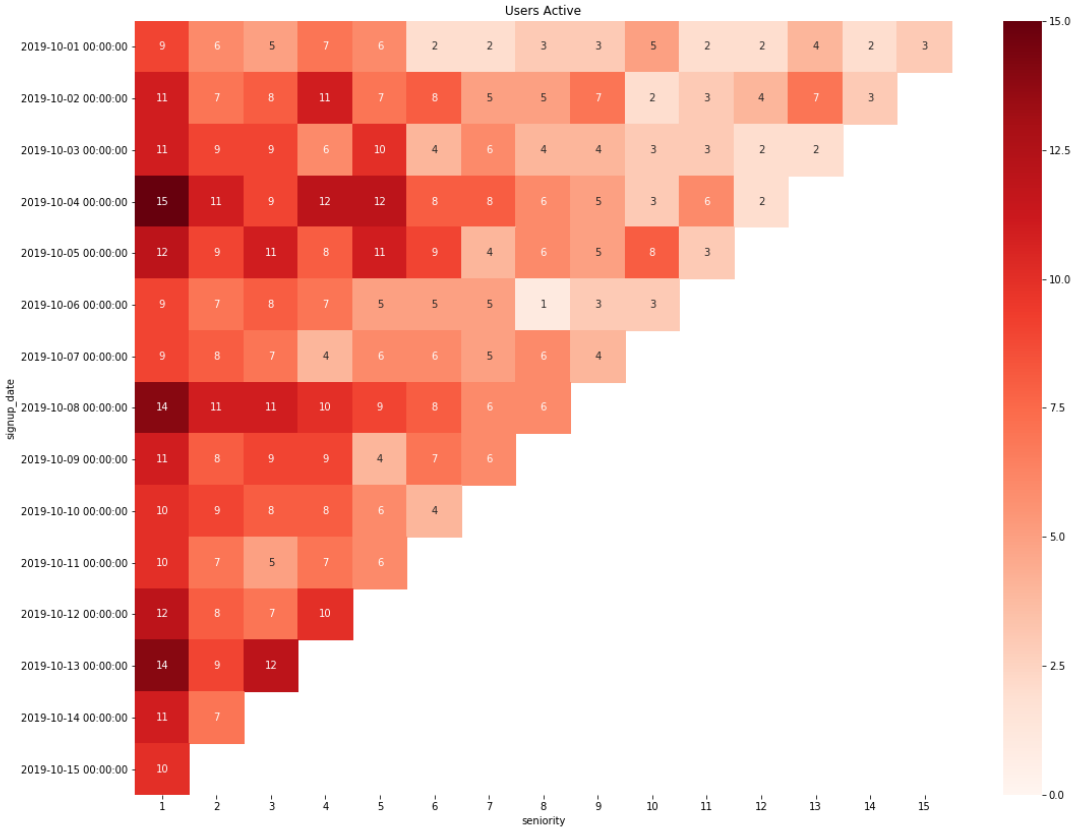
> The code below was based on the file structure of the *csv* I discussed earlier. If you are trying to use this for other projects, be aware that you may need to make some small adjustments — like with column names, for instance.

To manage your expectations (actually, to increase them), let me tell you what is supposed to be the end result of this project.

We will look at the data gathered from our app and apply some small changes to transform it into the right format. With this, we can then have a simple **pivot table with the number of users who were active each day**.

After that, we can also build the **retention rate table**, which is a variation of the first pivot table.

And finally, we build the **heatmaps** for the two tables and try to come up with any conclusions. I hope this is a strong enough incentive to keep you from *churning*. See what I did there?!



This heatmap displays how many users from each cohort (signup date on the left) were active during their lifecycle (their seniority).

Let’s get coding!

As always, we start by importing the packages we are going to use. Nothing fancy, but we will draw the heatmap with *seaborn* for our retention visualization, so we import *matplotlib* and *seaborn* for that. We will also read our dataset with *pandas.*

```
df.head(10)
```

	username	signup_date	ref_date	money	time	country
0	U10000	01/10/2019	01/10/2019	2	89	Canada
1	U10000	01/10/2019	07/10/2019	0	35	Canada
2	U10000	01/10/2019	09/10/2019	2	4	Canada
3	U10000	01/10/2019	10/10/2019	1	5	Canada
4	U10000	01/10/2019	12/10/2019	1	17	Canada
5	U10000	01/10/2019	15/10/2019	0	16	Canada
6	U10001	01/10/2019	01/10/2019	1	134	Canada
7	U10001	01/10/2019	02/10/2019	2	60	Canada
8	U10001	01/10/2019	03/10/2019	0	55	Canada
9	U10001	01/10/2019	04/10/2019	1	77	Canada

On the left, you can see the first 10 rows of the table we just imported.

According to this information, the user U10000 visited our app six times since he signed up.

We need to format the dates and add another column named *seniority*. Seniority basically tells us how old was the user on that day that he visited our app. We add the 1 to it because it makes it more readable, in the sense that seniority 10 means the user was on his/her 10th day since signup.

```
df.head(10)
```

	username	signup_date	ref_date	money	time	country	seniority
0	U10000	2019-10-01	2019-10-01	2	89	Canada	1
1	U10000	2019-10-01	2019-10-07	0	35	Canada	7
2	U10000	2019-10-01	2019-10-09	2	4	Canada	9
3	U10000	2019-10-01	2019-10-10	1	5	Canada	10
4	U10000	2019-10-01	2019-10-12	1	17	Canada	12
5	U10000	2019-10-01	2019-10-15	0	16	Canada	15
6	U10001	2019-10-01	2019-10-01	1	134	Canada	1
7	U10001	2019-10-01	2019-10-02	2	60	Canada	2
8	U10001	2019-10-01	2019-10-03	0	55	Canada	3
9	U10001	2019-10-01	2019-10-04	1	77	Canada	4

Now we are able to build the cohorts. But **what exactly is a cohort in this context**?

> Think of cohorts like buckets where groups of customers are placed according to a certain criteria.

With that, you can compare several metrics regarding the lifecycle of your product and your users. The important message here is “tracking their lifecycle”.

Our cohorts will be created from the signup dates (one for each date). That way they are **mutually exclusive**. We start by using the *groupby* method with *signup_date* and *seniority*, and getting their size. We also need to reset the index.

```
cohort_data
```

	signup_date	seniority	username
0	2019-10-01	1	9
1	2019-10-01	2	6
2	2019-10-01	3	5
3	2019-10-01	4	7
4	2019-10-01	5	6
5	2019-10-01	6	2
6	2019-10-01	7	2
7	2019-10-01	8	3
8	2019-10-01	9	3
9	2019-10-01	10	5
10	2019-10-01	11	2

On the left side, this is what the table should look like at this point.

It may not seem like much, but you now have the user count for each seniority (1 to 15 in our data) for each signup date.

But we are not done. We need to turn this into a pivot table.

As I mentioned before, the pivot table is what we will use to create the heatmap visualization later. It should have the *signup_date* in the index and *seniority* as columns. The values will be obtained from the *username* column, which after the previous step now contains the count of users for the cohorts.

I added a couple of things to the snippet above just so we have both tables ready for the next step. You might have noticed I did not share any formulas or some weird looking equations to calculate Retention. There is plenty of that online, and honestly, it's easier to get confused or apply them wrong if you don't understand what you are calculating.

> We are trying to find out how many customers we "retain" from one day to the other. First we get an absolute number, and second we calculate a percentage.

This percentage is nothing less than the ***number of active users from the same cohort in a day***, divided by the ***number of users the cohort started with on day 1***. If 10 people signed up on day N, and only 4 were still around after 7 days, the retention rate for that cohort after 7 days is 40% (4/10).

I've been told he is quite convincing...

The churn rate is the exact opposite. If we kept 40% of our users, it means that 60% of them have churned. To keep things simple, let's discuss just the retention in deep, but keep in mind that the churn rate is the exact opposite of retention rate!

The snippet above shows how to isolate the number of signups from each cohort, at the start of their lifecycle (seniority = 1). I used a variable aptly named *base*, and that is what I used to calculate the retention rate, dividing the whole *cohort_counts* matrix by the *base*. Check out the two pivot tables below:

```
cohort_counts
```

seniority	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
signup_date															
2019-10-01	9.0	6.0	5.0	7.0	6.0	2.0	2.0	3.0	3.0	5.0	2.0	2.0	4.0	2.0	3.0
2019-10-02	11.0	7.0	8.0	11.0	7.0	8.0	5.0	5.0	7.0	2.0	3.0	4.0	7.0	3.0	NaN
2019-10-03	11.0	9.0	9.0	6.0	10.0	4.0	6.0	4.0	4.0	3.0	3.0	2.0	2.0	NaN	NaN
2019-10-04	15.0	11.0	9.0	12.0	12.0	8.0	8.0	6.0	5.0	3.0	6.0	2.0	NaN	NaN	NaN
2019-10-05	12.0	9.0	11.0	8.0	11.0	9.0	4.0	6.0	5.0	8.0	3.0	NaN	NaN	NaN	NaN
2019-10-06	9.0	7.0	8.0	7.0	5.0	5.0	5.0	1.0	3.0	3.0	NaN	NaN	NaN	NaN	NaN
2019-10-07	9.0	8.0	7.0	4.0	6.0	6.0	5.0	6.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-08	14.0	11.0	11.0	10.0	9.0	8.0	6.0	6.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-09	11.0	8.0	9.0	9.0	4.0	7.0	6.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-10	10.0	9.0	8.0	8.0	6.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-11	10.0	7.0	5.0	7.0	6.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-12	12.0	8.0	7.0	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-13	14.0	9.0	12.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-14	11.0	7.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-15	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
retention = cohort_counts.divide(base, axis=0).round(3)
retention
```

seniority	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
signup_date															
2019-10-01	1.0	0.667	0.556	0.778	0.667	0.222	0.222	0.333	0.333	0.556	0.222	0.222	0.444	0.222	0.333
2019-10-02	1.0	0.636	0.727	1.000	0.636	0.727	0.455	0.455	0.636	0.182	0.273	0.364	0.636	0.273	NaN
2019-10-03	1.0	0.818	0.818	0.545	0.909	0.364	0.545	0.364	0.364	0.273	0.273	0.182	0.182	NaN	NaN
2019-10-04	1.0	0.733	0.600	0.800	0.800	0.533	0.533	0.400	0.333	0.200	0.400	0.133	NaN	NaN	NaN
2019-10-05	1.0	0.750	0.917	0.667	0.917	0.750	0.333	0.500	0.417	0.667	0.250	NaN	NaN	NaN	NaN
2019-10-06	1.0	0.778	0.889	0.778	0.556	0.556	0.556	0.111	0.333	0.333	NaN	NaN	NaN	NaN	NaN
2019-10-07	1.0	0.889	0.778	0.444	0.667	0.667	0.556	0.667	0.444	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-08	1.0	0.786	0.786	0.714	0.643	0.571	0.429	0.429	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-09	1.0	0.727	0.818	0.818	0.364	0.636	0.545	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-10	1.0	0.900	0.800	0.800	0.600	0.400	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-11	1.0	0.700	0.500	0.700	0.600	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-12	1.0	0.667	0.583	0.833	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-13	1.0	0.643	0.857	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-14	1.0	0.636	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2019-10-15	1.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

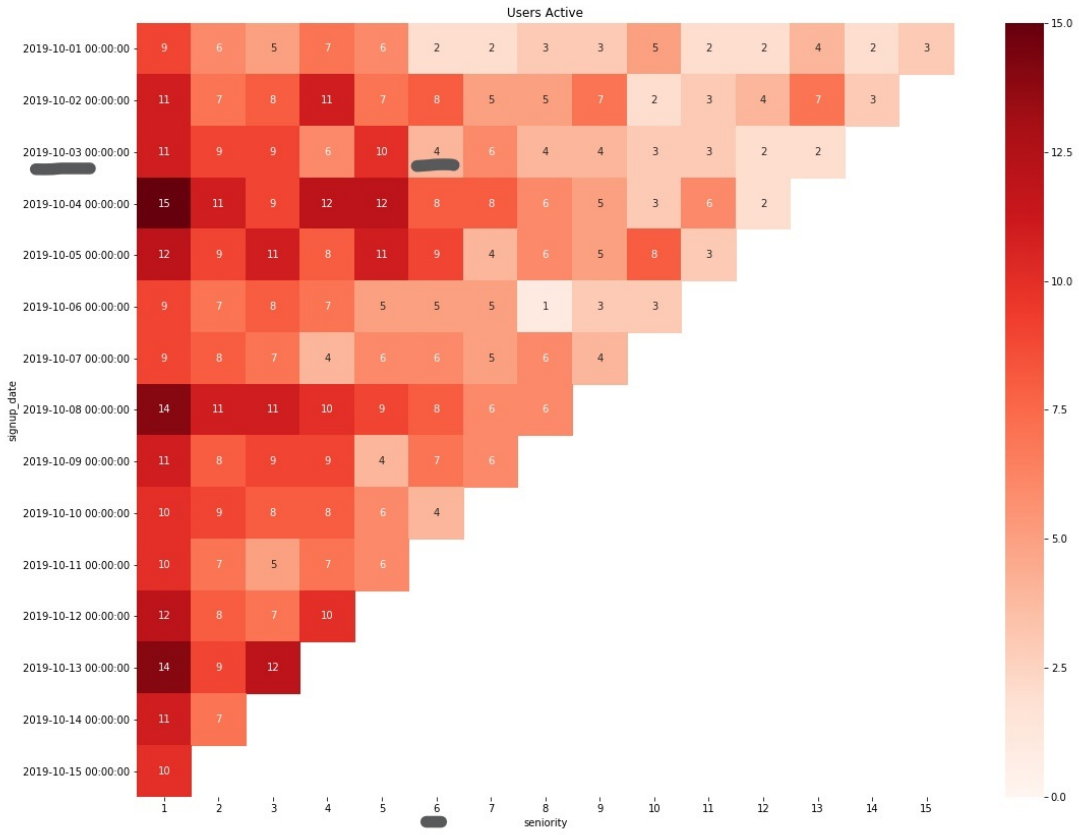
Typically, the retention rate should be 100% on the 1st day since we divide the whole table by the first column. It only makes sense, given that in our case every user was active on their signup date.

Heatmaps are great to visualize Cohort Analysis

As long as you know how to read them! Below you can find the code to get two heatmaps, one with the user count, and the other with the retention rate.

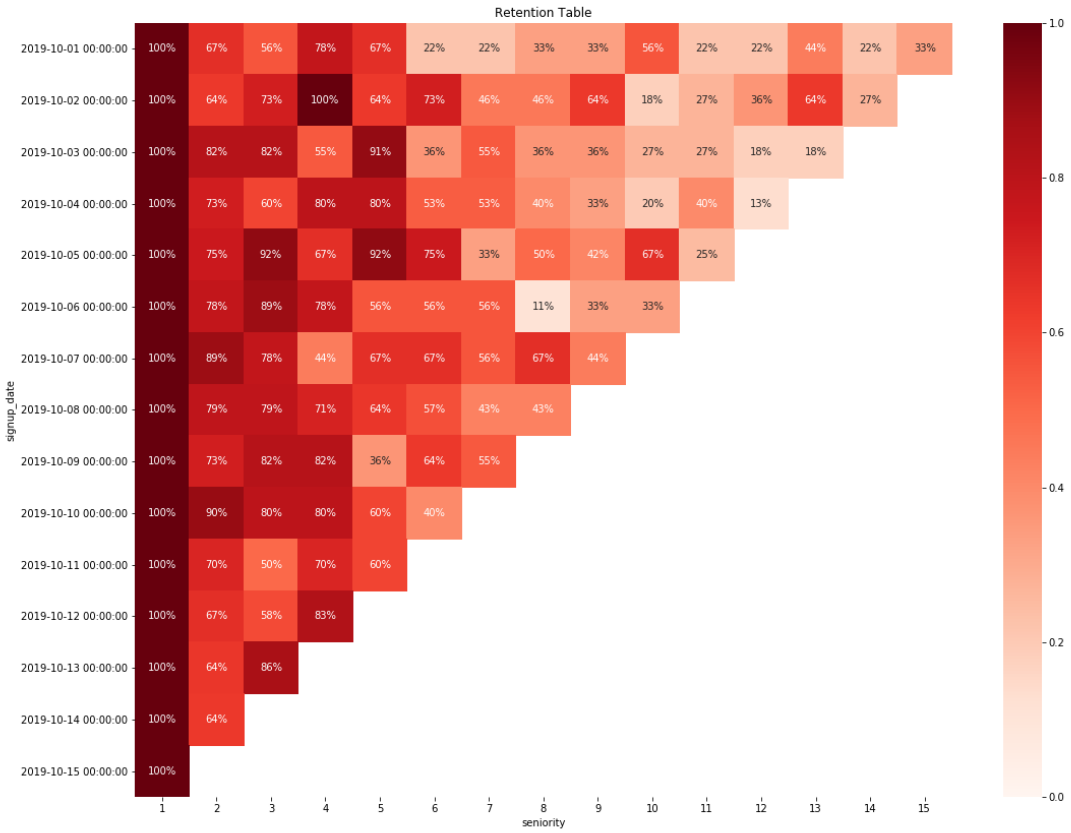
- Each row is one cohort of users who signed up on the same date.
- Each column represents where the user is at, in his/her lifecycle (measured in days, weeks, months, etc).

There are two main approaches to how you can extract some useful insights from this heatmap. The most obvious way is to look at **how a specific cohort behaves as they get "older"**. Let's call it a horizontal approach. In a typical heatmap, you will see the number of users dropping from left to right. Unless you have an outstanding product where every user keeps coming back every day/month/etc!



> To reinforce the point, the square I marked above shows that 4 users, of all the 11 that signed up on the 3rd of October, were active on their sixth day.

The second approach — and maybe a not so obvious one — is to evaluate **how our product is behaving in terms of retaining users at specific seniority stages**. Let's call this one a vertical approach. It's easier to understand using the retention table instead of the user count, because typically the cohorts may not have the same base to compare them vertically — whereas the rate is a percentage. Have a look at the heatmap and check out the example after.



The dataset is not large enough to draw big conclusions, but at least it looks cool...

As an example, imagine we updated our app to push a notification when users reach their 8th day. We would expect that some users would log in on the 8th day, maybe we can even offer them something as an incentive for them to be engaged again. We would then expect some sort of spike around the 8th day for the users that got the update. Looking at the seniority column equal to 8, we can then see how the retention changed from cohort to cohort.

> This is how you can create A/B tests for your new features, or new user experience, or whatever you feel like it could impact retention. This is how you actually monitor if it's working.

Final remarks

This article definitely grew beyond what I had in mind! I still wanted to show you how you could compare retention from different countries, and how you can also build a cohort analysis for a different metric than the user count.

The **country** comparison is very simple. You just need to split the dataframe *df* according to the countries you want to compare and follow the exact same code snippets for each new country dataframe.

In the file, we also have two columns “money” and “time”, which we can use to build a cohort analysis that will show us how much “money” or “time” each cohort spends along their lifecycle. One idea is to check if users are spending more or less as their lifecycle evolves. If you’re trying to increase a specific cohort spending, this is a good way to track it.

Unfortunately, the article got quite long so I will leave this last suggestion as a challenge for you. If you really really… *really*, want me to write another article about cohort analysis, lifetime value, or a follow up on this one, please let me know in the comments. I try to answer all of them, so I’ll notice if many people make the same request!