Based on the initial analysis, we were able to find number of the active customers, new customer, and churning customers per month (*see spreadsheet Excel file*). Besides, the monthly SaaS was calculated, which represents the whole revenue from all customers. As shown in Figure 1, the relationship between the number of active customers with SaaS revenue is depicted. There is a similar pattern found in this chart. As number of users increased, the revenue is getting higher. However, it is worth noting that on some points, these two trends are not identical. For example, on 04-2022 a huge drop is coming while the number of customers has not changed dramatically. We would be able to find the reason in contracts to determine how well we are generating money from our core revenue-driving operations. The question is why we could not boost our revenue. Maybe the cost of production has been changed, or the price of the products got lower than before. To get an accurate financial picture, we have to see how expenses and operating costs impact our revenues.

**Figure 1 – Active Customers versus SaaS Revenue**

In Figure 2, the cumulative number of customers is compared to the cumulative revenue. If we assume that the rate of growth in these two trends are linear, we can notice that growth rate of revenue is significantly higher than the number of total customers acquired. One action that might be required is to consider the price elasticity. It means that we need to figure out the demand for our products in reaction to changes in the price for those products. Another point is that the number of new customers turned to zero for a long period of time. As the demand is strictly a function of changes in price, we would need to adjust the prices accordingly. if the price goes down, people will buy more. In this case, I would study total revenue, marginal revenue and elasticity of demand to find the optimum conditions.

**Figure 2 – Total Customers Acquired versus Total Revenue**

**Code (in PostgreSQL)**

with

activity as(

Select active as months, count(\*) as Active\_Customers, sum(saas\_monthly) as total\_monthly\_saas\_amount

from (Select name, generate\_series(DATE\_TRUNC('month', origin\_date::date), origin\_date::date + (saas\_contract\_duration-1)\*'1 month'::interval , '1 month'::interval) as active, saas\_contract\_total\_value\_in\_usd/saas\_contract\_duration as saas\_monthly from challenge\_companies) as act

where active<'2024-01-01'

group by active),

startdate as(

Select start\_date, count (\*) as new\_customers

from (Select DATE\_TRUNC('month', origin\_date::date) as start\_date from challenge\_companies) as STRT

group by start\_date),

churning as(

Select churned\_date, count (\*) as churning\_customers

from (Select DATE\_TRUNC('month', churned\_date::date) as churned\_date from challenge\_companies) as CHRN

group by churned\_date)

Select months::date, active\_customers, coalesce(new\_customers,0) as new\_customers, coalesce(churning\_customers,0) as churning\_customers, total\_monthly\_saas\_amount, coalesce(sum(new\_customers) over (order by months),0) as total\_customers\_acquired, coalesce(sum(churning\_customers) over (order by months),0) as total\_customers\_churned

from activity

left join startdate on months=start\_date

left join churning on months=churned\_date

order by months

Graphical user interface, application, table

Description automatically generated