Ironhack Payments

Maria Aguilar, Nancy, Patricia Giménez, Taynã Appel

Exploratory Data Analysis (EDA) **Report**

01 Objective of the EDA

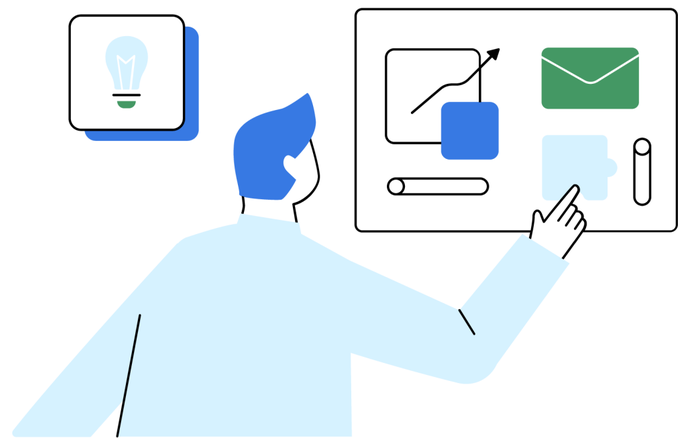
02 Dataset Overview

03 Key Distributions & Visualizations

04 Key Insights and Analysis

05 Data Quality

06 Observations & Limitations



01

Objective of the EDA

The goal of the Exploratory Data Analysis (EDA) is to gain an **initial understanding of the structure, content, and patterns within the dataset**. Before performing the cohort analysis, it is essential to explore the data to:

- Understand how **users interact** with Ironhack Payments' cash advance services.
- **Identify trends** in user activity over time.
- Examine the **distribution of request** statuses and amounts.
- Detect **potential outliers** or unusual behavior in user transactions.
- Highlight any early indicators of **service engagement** or **drop-off**.

This step also provides **valuable context** for designing the cohort **segmentation logic and helps ensure the quality and reliability** of the data used in subsequent analysis.

IRON HACK 02

Dataset Overview

Cash Advance Requests

Contains individual records of cash advance requests made by users.

- Rows: Each row represents a single cash request.
- Key columns:
  - **id:** Unique ID for the cash request
  - **user_id:** ID of the user who made the request
  - **amount:** Amount requested
  - **status:** Status of the request (e.g., accepted, rejected)
  - **created_at:** Timestamp when the request was created
  - **reimbursement_date**: Date the user is expected to repay
  - **transfer_type**: Type of transfer (e.g., regular)
  - **money_back_date**: Date the money was returned (if applicable)

Fees and Incidents

Contains financial and operational information related to the cash requests, including fees and incident records.

- Rows: Each row represents a fee or incident linked to a cash request.
- Key columns:
  - **id**: Unique ID for the fee/incident
  - **cash_request_id**: ID of the related cash request
  - **type**: Type of record (e.g., instant_payment, incident)
  - **status**: Status of the fee/incident
  - **total_amount**: Fee amount charged (used to calculate revenue)
  - **category**: Incident type, if applicable
  - **created_at**: Timestamp of fee creation
  - **paid_at**: Date the fee was paid

Together, these datasets provide a comprehensive view of user behavior, transaction activity, and the resulting revenue or service incidents. This overview informs the design of the cohort analysis.

IRON HACK

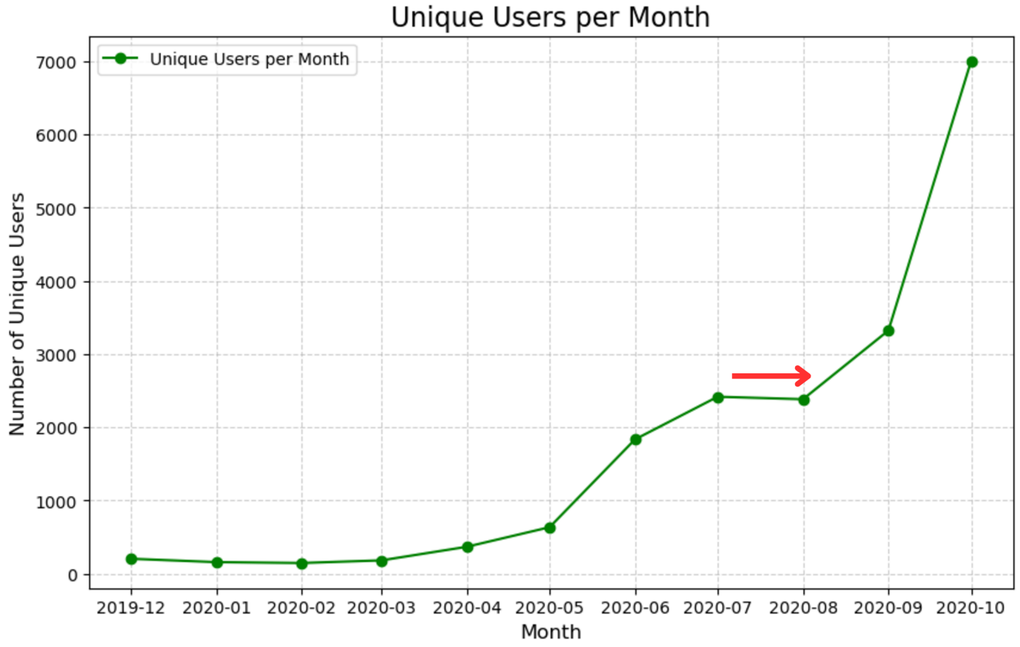
03

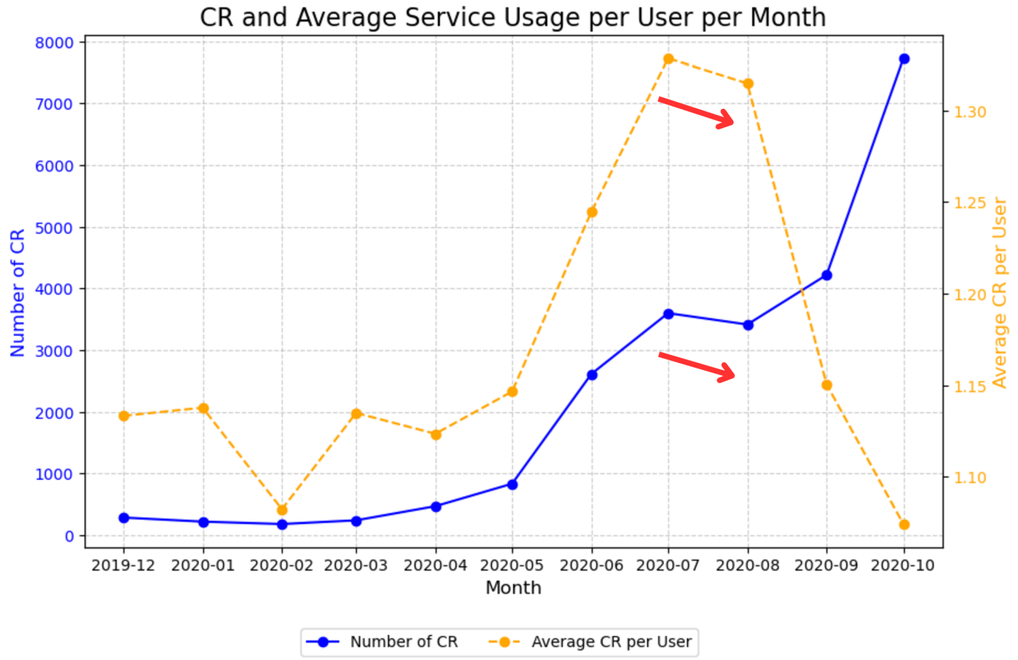
Key Distributions & Visualizations

A good indicator of **service performance** is the calculation of Cash Request per month. When this was calculated we wanted to go deeper to understand if the increase in CR was due to an increase of users or an increase of the amount of CR per user.

The graphs show **both tendency for users and cash requests** with the same distribution.
And this also reflect the **increase in CR** being mainly **due to an increase in users** (new users using our service).

For example, between July and August the amount of cash request decreased, but in this case was because the number of CR per user decreased (no new users during this period).



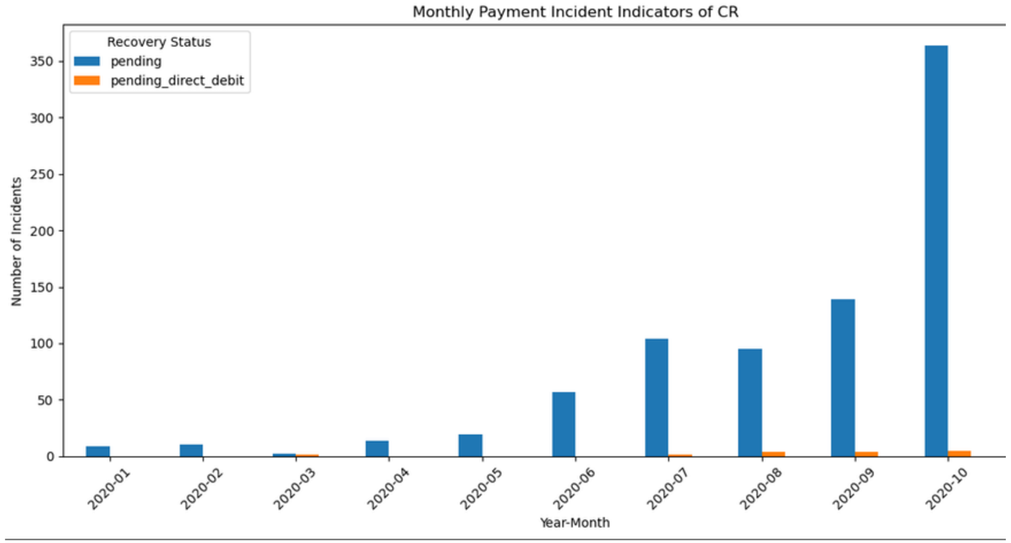


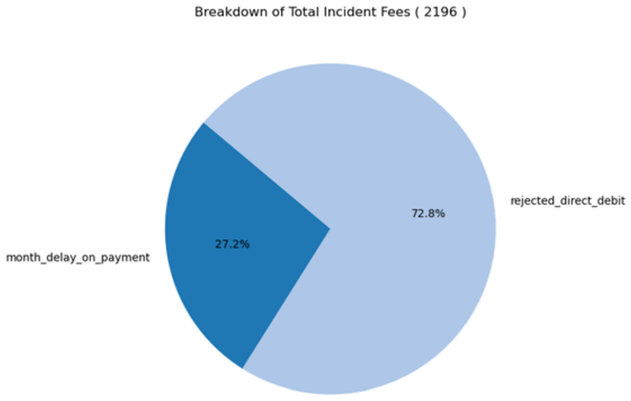
IRON HACK 03

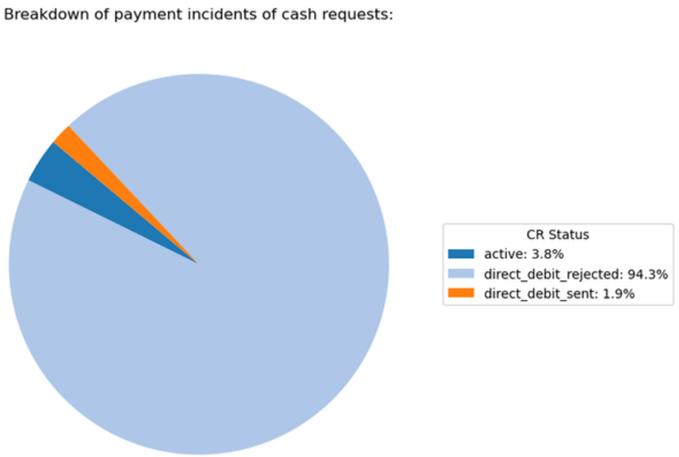
Key Distributions & Visualizations

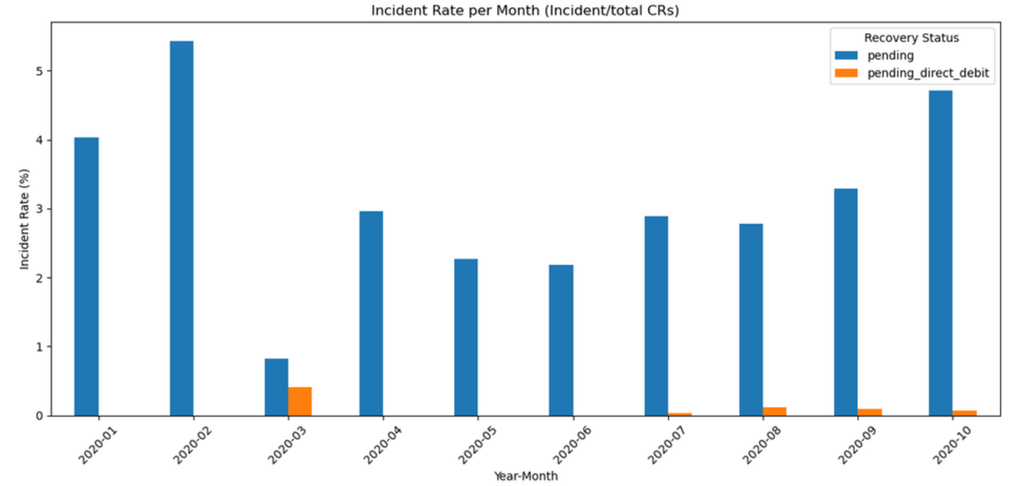
For the business to be suistainable, it is necessary to have an insight into the incidence rate of customer loan reimbursments.

In the top graph we can see that the amount of payment incidents is increasing due to the higher amount of cash requests/users and a “constant” incident rate.









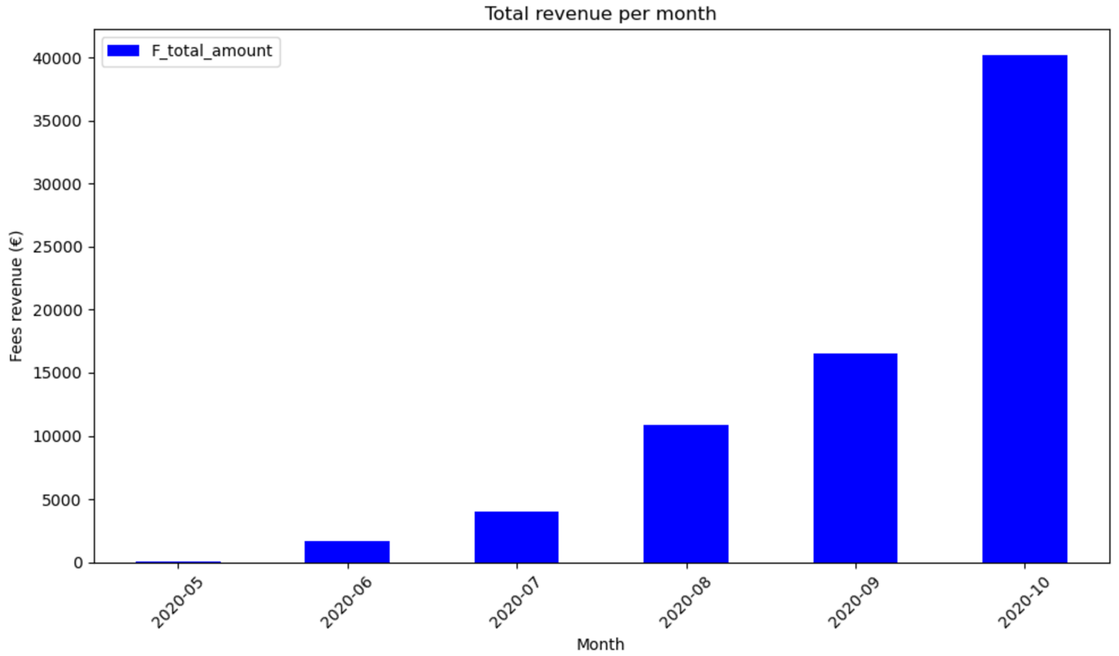
IRON HACK

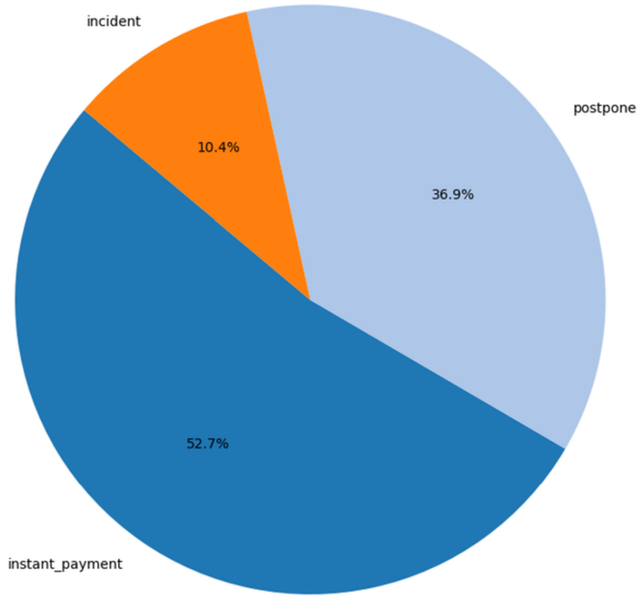
03

Key Distributions & Visualizations

Analyzing the distribution of total revenue fee amounts per month throughout the year was important to check growth or decline trends and to test other hypotheses about what could be affecting Ironhack's growth.
Hypothesis: This analysis showed that October was responsible for the higher fees revenues, following the trend analyzed before on user growth.

We also analyzed what **kind of fees** provided the higher total amounts of revenues. In that way, we can understand weather to charge higher or lower fees depending on their frequencies and returns.





IRON HACK 03

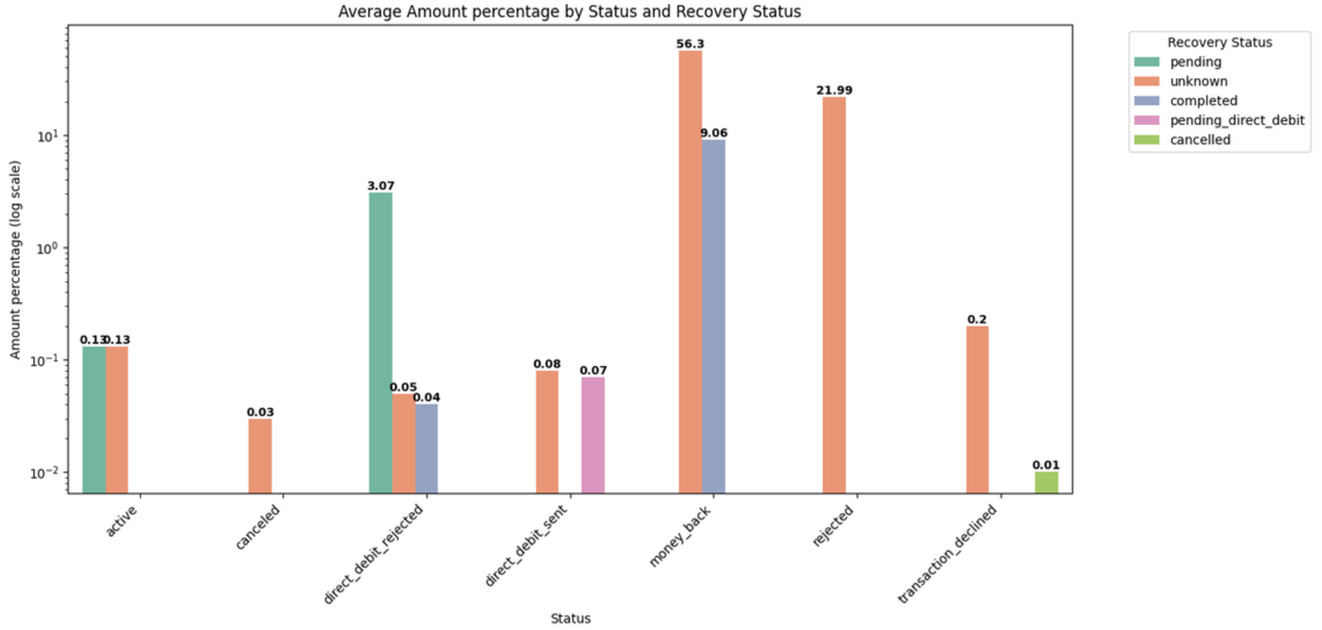
Key Distributions & Visualizations

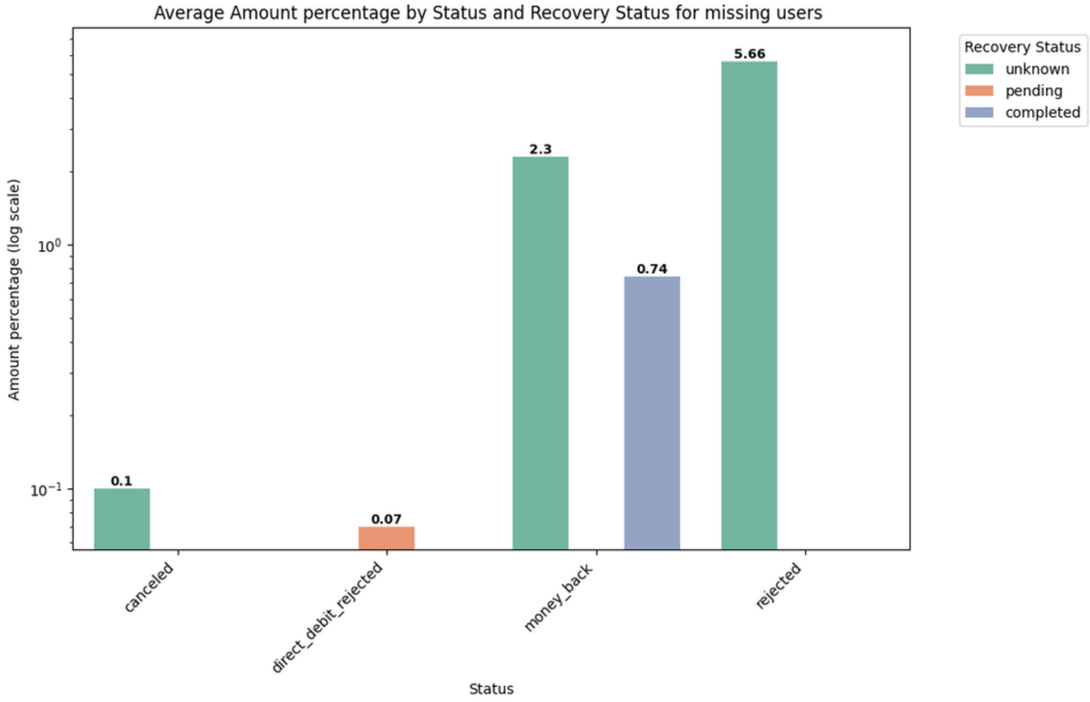
Revenue distribution

- Missing users
  - ~91% amount is tagged with users
  - ~9% amount is mapped with deleted users

- Status
  - ~68% of amount is returned : (65+3)
  - ~27% of amount is rejected :(22+5)
  - ~5% is due payment

Assumption: all unknown recovery status are same as status

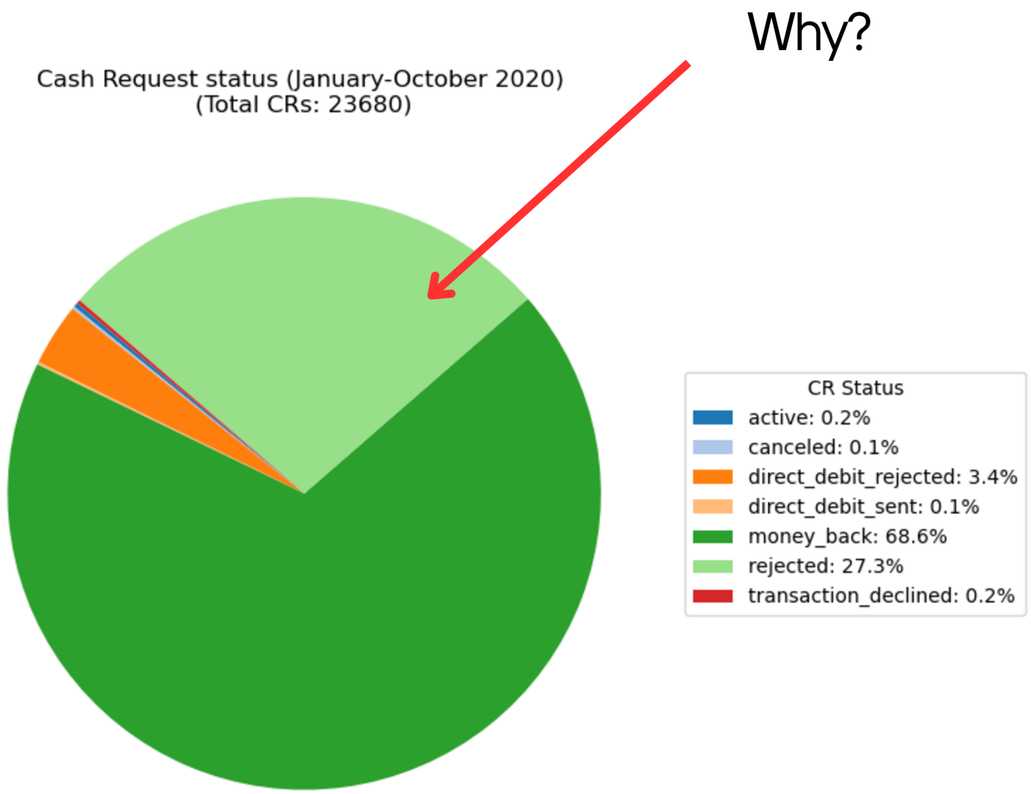




Data Visualization & Analysis

CR status indicators for the year

- ~27% Cash requests are rejected with unkown reasons



Data Visualization & Analysis

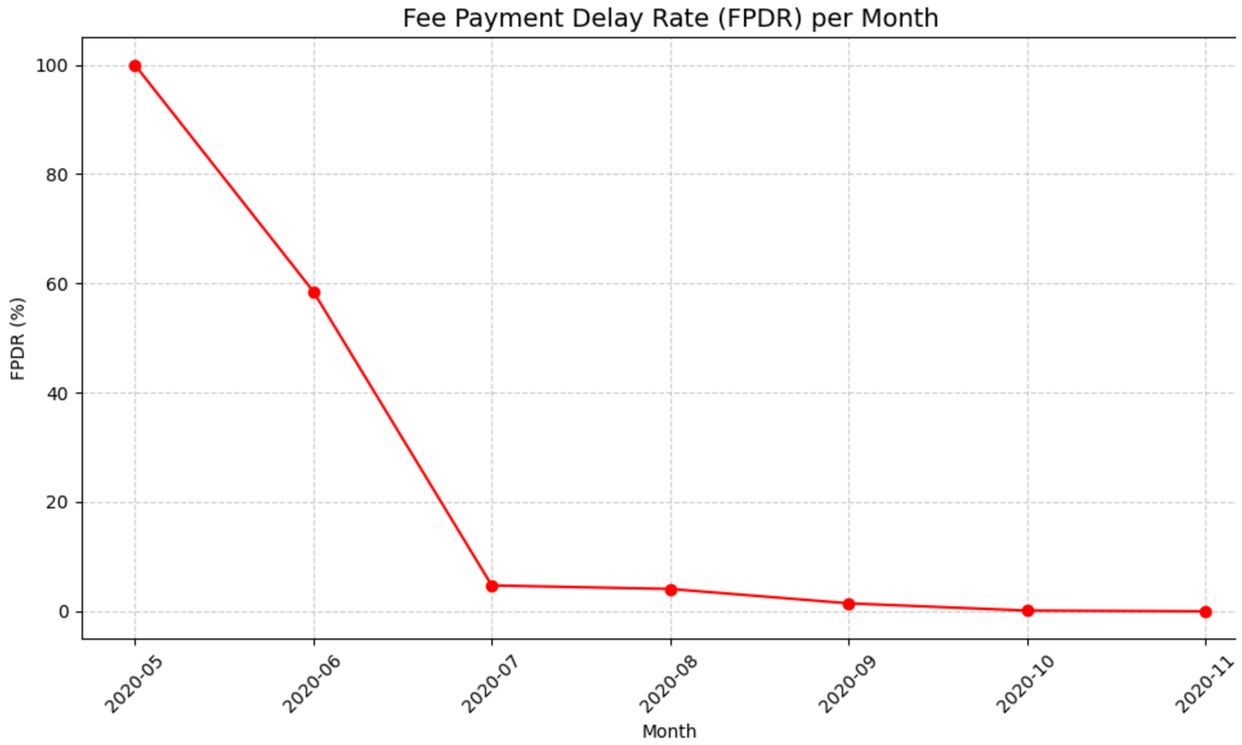
Fee Payment Delay Rate (FPDR) per Month

To ensure the sustainability of the platform, it is important to understand how users behave after accepting service fees. The **Fee Payment Delay Rate (FPDR)** helps us measure the percentage of accepted fees that remain unpaid over time.

This metric allows us to evaluate whether users who agree to pay a fee actually follow through, which is crucial for revenue stability and user trust.
In the early stages (e.g., May 2020), we observed a high FPDR. This is likely due to the launch of the platform, where payments were pending or still in processing. Over the following months, the FPDR steadily declined, suggesting improved user compliance and system maturity.

A decreasing FPDR is a positive signal: it indicates that users are paying their accepted fees more consistently, and the platform is becoming more **reliable in terms of fee recovery.**

Contrary to the initial hypothesis, the **Fee Payment Delay Rate (FPDR) has significantly decreased over time**, even as service usage increased.
This suggests a positive evolution in user payment behavior and potentially better platform processes or communication.



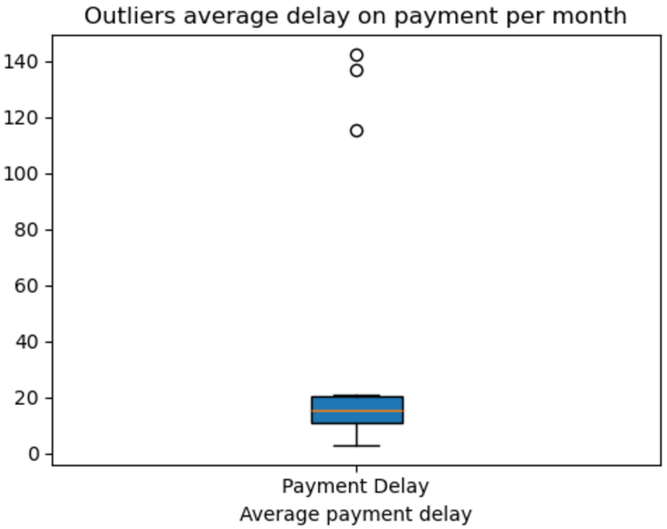
Hypothesis:
As service usage increases, the percentage of unpaid accepted fees might also increase, revealing possible liquidity problems or decreasing user engagement.

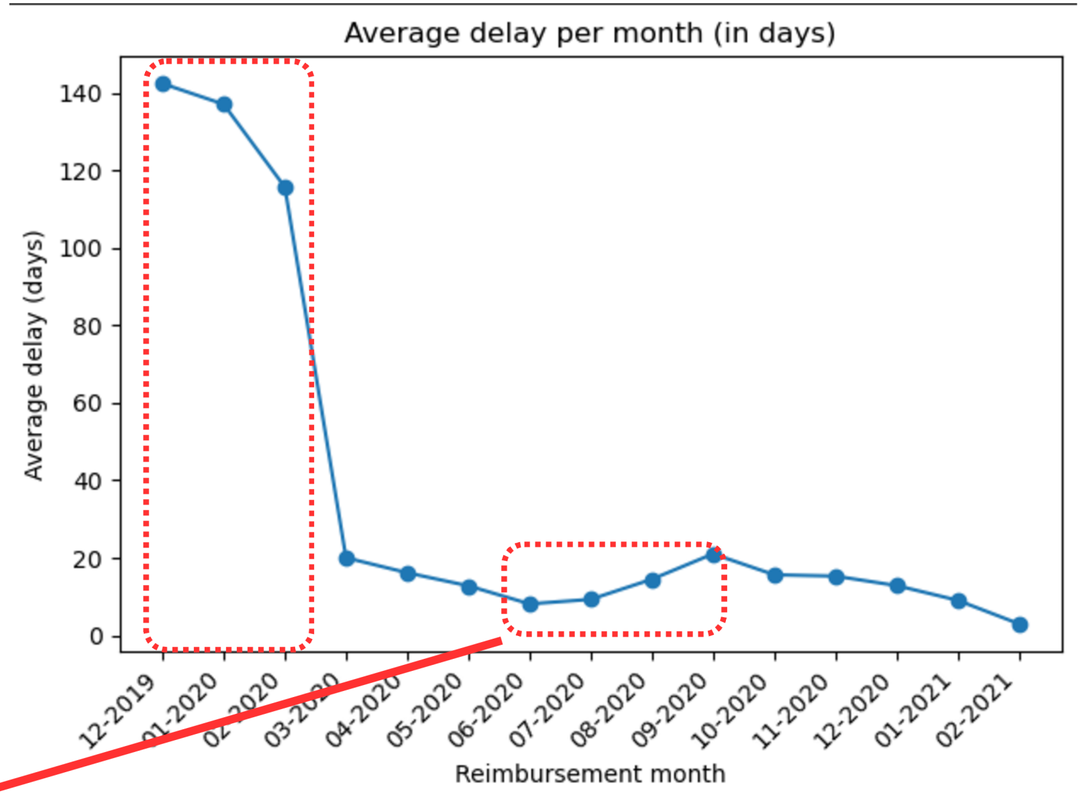
FPDR (%) = (accepted fees with no paid_at date) / (Total Accepted Fees) * 100

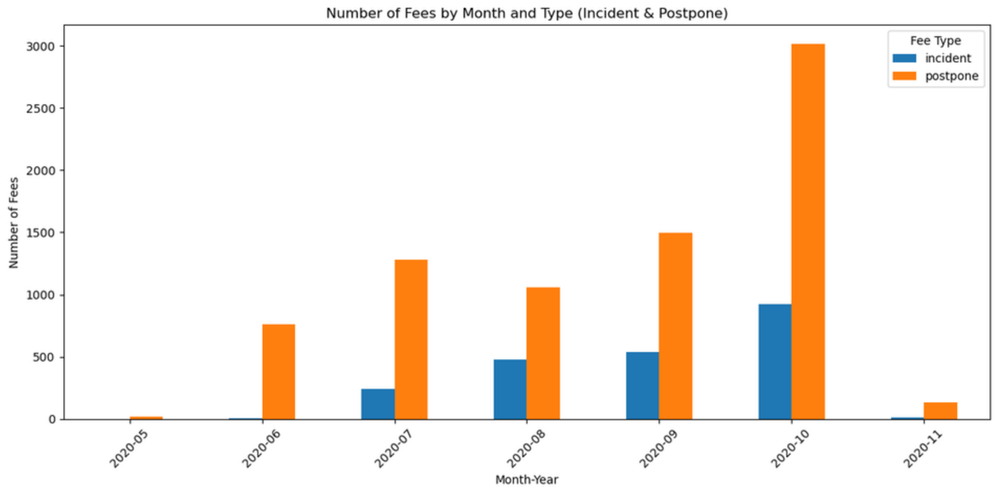
Data Visualization & Analysis

Delay con Cash Request Reimbursement

Only money back CR and late payments are considered



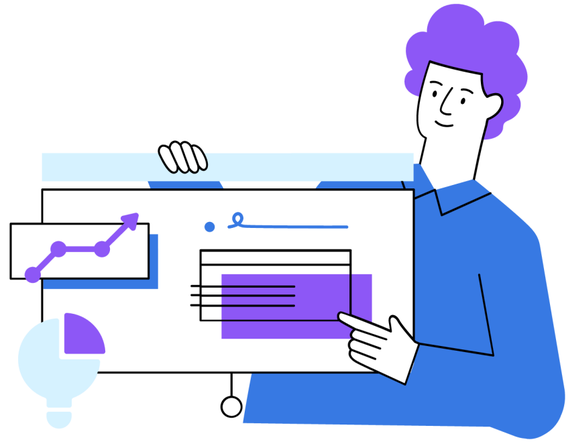




IRON HACK

05

Data Quality



1. Some missing values of Cash Request in the fees information.

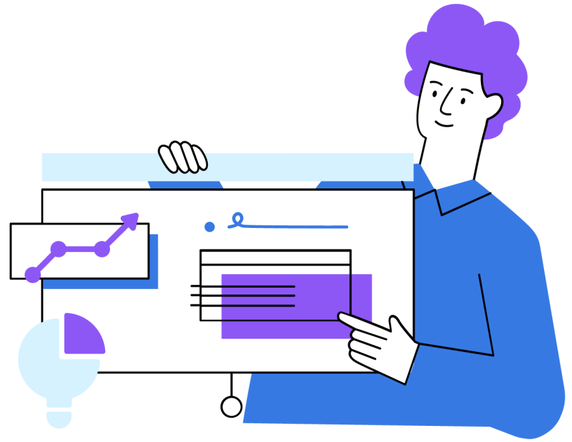
Solution: Completing data with the information of the id found in the reason column

```
id                   0
cash_request_id      4
type                 0
status               0
category         18865
total_amount         0
reason               0
created_at           0
updated_at           0
paid_at           5530
from_date        13295
to_date          13295
charge_moment        0
dtype: int64
```

	id	cash_request_id	type	status	category	total_amount	reason	created_at	updated_at	paid_at	f
1911	2990	NaN	instant_payment	cancelled	NaN	50	Instant Payment Cash Request 11164	2020-08-06 22:42:34.525373+00	2020-11-04 16:01:17.296048+00	NaN	
1960	3124	NaN	instant_payment	cancelled	NaN	50	Instant Payment Cash Request 11444	2020-08-08 06:33:06.244651+00	2020-11-04 16:01:08.332978+00	NaN	
4605	5185	NaN	instant_payment	cancelled	NaN	50	Instant Payment Cash Request 11788	2020-08-26 09:39:37.362933+00	2020-11-04 16:01:36.492576+00	NaN	
11870	3590	NaN	instant_payment	cancelled	NaN	50	Instant Payment Cash Request 12212	2020-08-12 14:20:06.657075+00	2020-11-04 16:01:53.106416+00	NaN	

IRON HACK 05

Data Quality



2. Some missing paid fees date when fee is accepted (financial department). We assumed fee is paid but the data is missing.

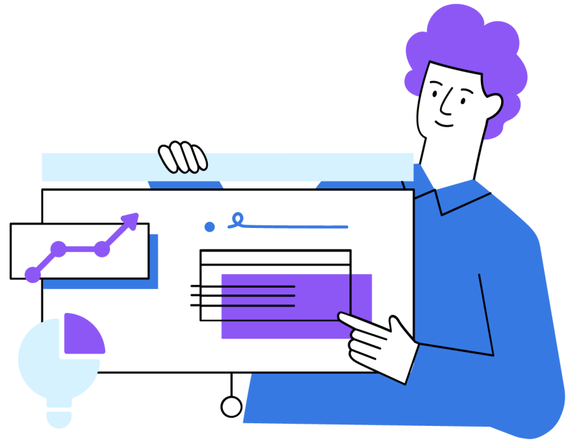
CR_recovery_status	...	F_type	F_status	F_total_amount	F_reason	F_created_at	F_updated_at	F_paid_at	F_from_date	F_to_date	F_charge_moment
NaN	...	postpone	accepted	5.0	Postpone Cash Request 1554	2020-06-01 19:29:27.603878+00	2020-10-13 14:25:00.797171+00	NaN	2020-06-18 22:00:00+00	2020-07-06 22:00:00+00	before
NaN	...	postpone	accepted	5.0	Postpone Cash Request 1554	2020-07-01 23:07:59.917774+00	2020-10-13 14:25:01.81737+00	2020-07-01 23:08:03.927568+00	2020-07-06 22:00:00+00	2020-08-05 22:00:00+00	before
NaN	...	postpone	cancelled	5.0	Postpone Cash Request 1554	2020-06-27 00:44:09.957168+00	2020-10-13 14:25:15.822258+00	NaN	2020-07-06 22:00:00+00	2020-08-05 22:00:00+00	after
NaN	...	postpone	cancelled	5.0	Postpone Cash Request 1554	2020-07-01 23:05:57.08772+00	2020-10-13 14:25:16.002938+00	NaN	2020-07-06 22:00:00+00	2020-08-05 22:00:00+00	after

In this example, fee paid_at column is missing the date. However F_status is accepted and F_charge_moment is set as “before”.

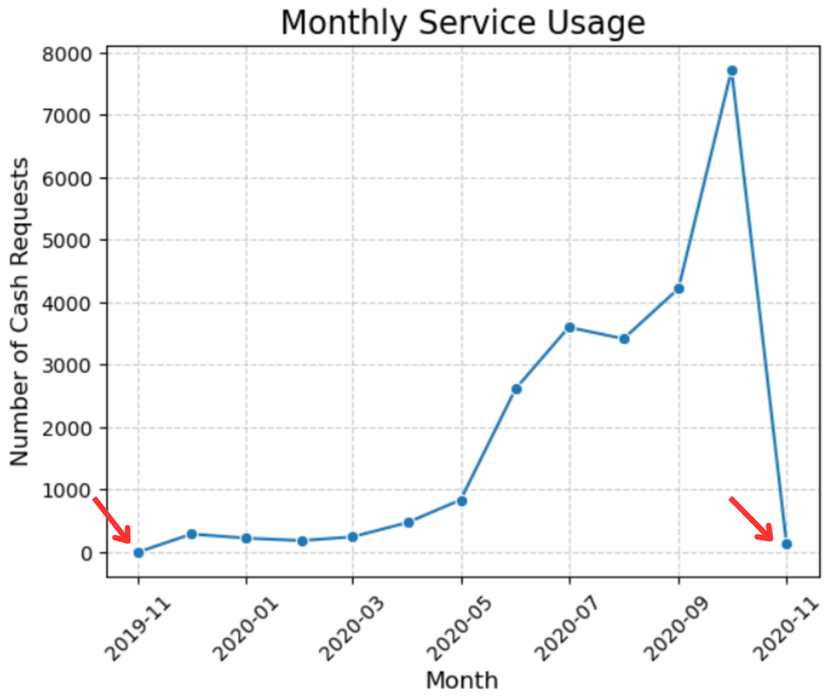
IRON HACK

05

Data Quality



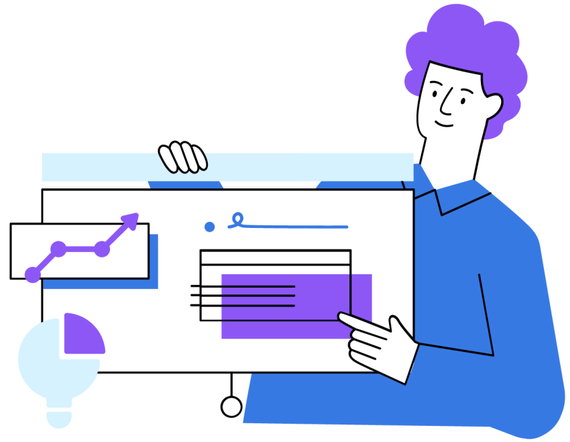
3. First and last month of the dataset were not fully completed with data, so we removed them for the analysis.



	CR_id_count
count	13.000000
mean	1843.846154
std	2344.271914
min	1.000000
25%	223.000000
50%	473.000000
75%	3417.000000
max	7725.000000

IRON HACK 05

Data Quality



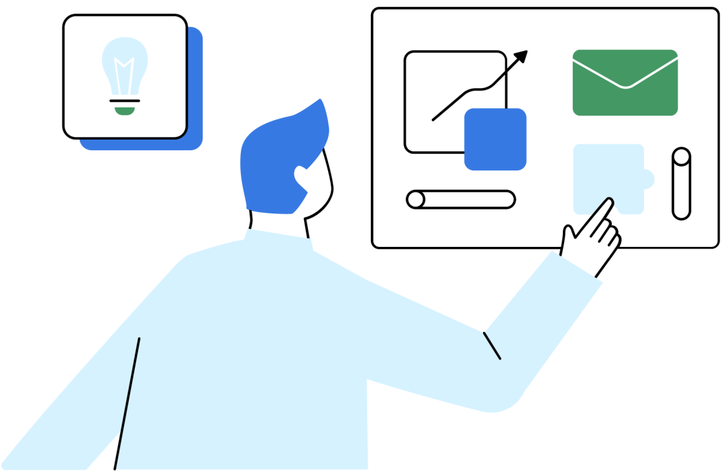
4. All **date fields** appeared to be in a specific datetime format and for the analysis we should:
Standardize time zones or remove for consistency.
Columns like *created_at*, *updated_at*, and *paid_at* need to be **converted to datetime** types.

5. **Duplicates** were also checked to make sure we would have the correct amount of users and guarantee data quality.

IRON HACK

06

Observations & Limitations



Point 01

There are sufficient events per user so It's Ideal for our cohort analysis, but understanding the data was our big challenge to the analysis

Point 02

We could have more data on the users to make further analysis as modelling risk per profile of each user or region etc.

Point 04

To understand seasonality of the cash requests more data is required

Point 05

Data structure could be simplified and improve, we can't see different stages of CR_status as it is a live variable

Point 06

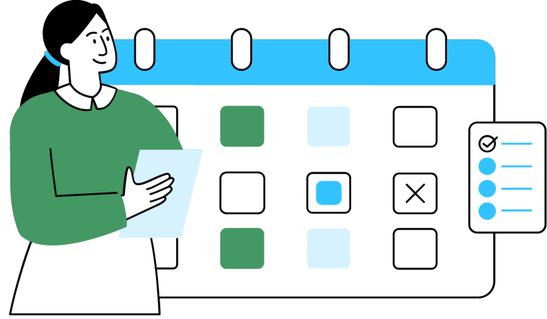
We missed more information on the reasonability behind the status of requests rejected and the recovery status.

SLIDES FOR PRESENTATION

Ironhack Payments

Maria Aguilar, Nancy, Patricia Giménez, Taynã Appel

Data Analysis Process



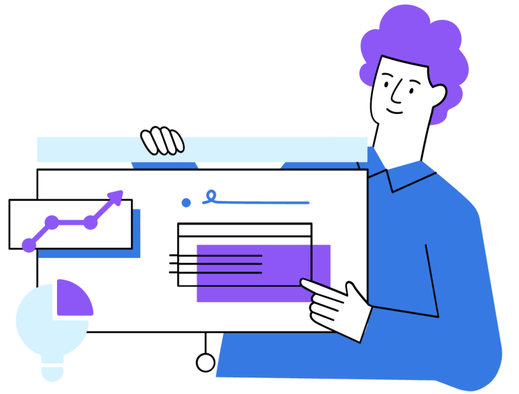
Cleaning/quality → EDA → Report

Make sure our **data has quality**, It's properly clean and will provide accurate information to the analysis. We should check for **missing values, outliers, inconsistencies ....**

In order to properly do that, we start cleaning the databases while we **investigate and understand** the meaning of each variable and how they are impacting each other, properly exploring the data and being able to **create hypothesis** (aligning with the cleaning of it)...

And finally we can provide **Insights** and raise some **analytical solutions** to our raised hypothesis. In our report we were able to explain **each step on the way**, to make sure the full picture was provided.

Data Quality Review



Identifying issues and cleaning process

1. Some missing values of Cash Request in the fees information.

Solution: Completing data with the information of the id found in the reason column.

2. Some missing paid fees date when fee is accepted (financial department).

Solution: Flag the problem to the finance team and drop the lines that are not consistent

3. First and last months of the dataset were not fully filled with data.

Solution: we removed them for the analysis.

4. All date fields appeared to be in a unconsistent datetime format

Solution: errors were handled and all formats were considered.

5. Duplicates were also checked to guarantee data quality.

Data Visualization & Analysis

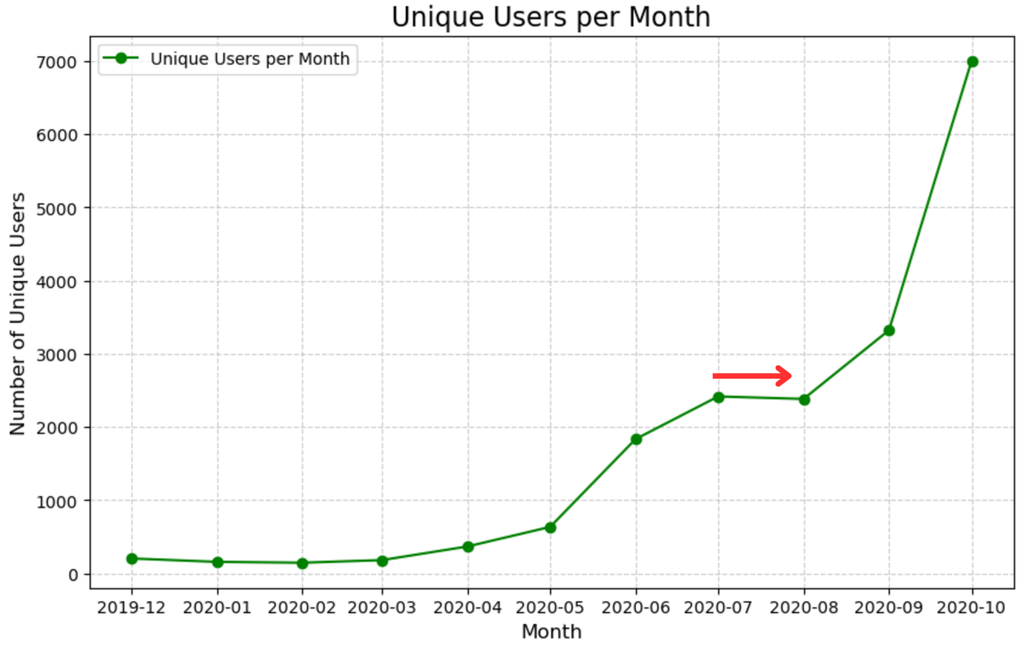
Frequency of Service Usage

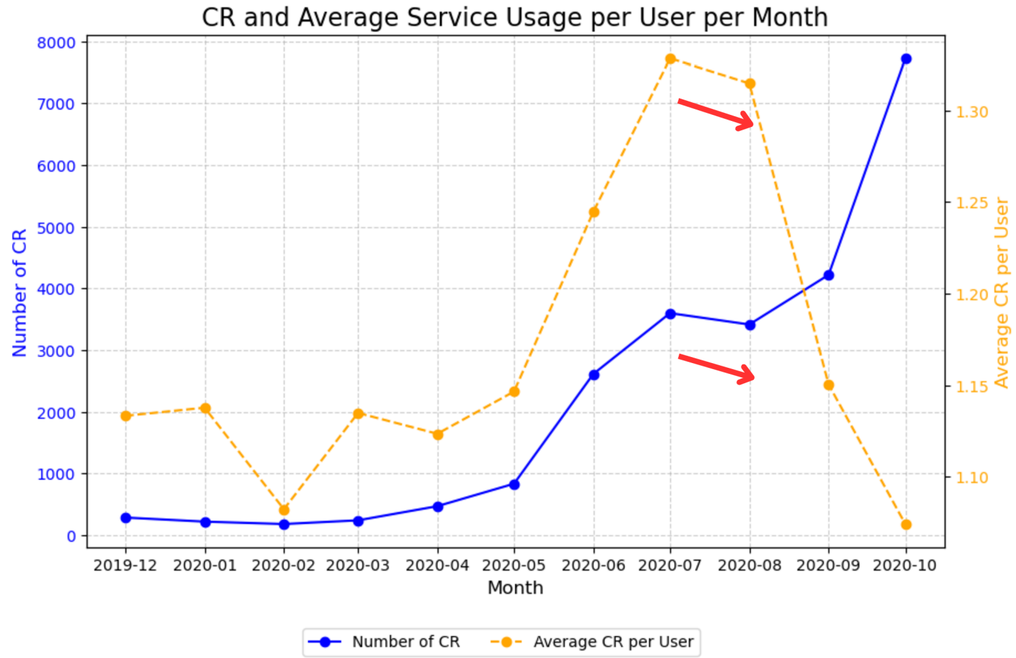
Service performance --> calculation of Cash Request per month. Understand if the increase in CR was due to an increase of users or an increase of the amount of CR per user.

The graphs show **both tendency for users and cash requests** with the same distribution.

And this also reflect the **increase in CR** being mainly **due to an increase in users** (new users using our service).

For example, between July and August the amount of cash request decreased, but in this case was because the number of CR per user decreased (no new users during this period).



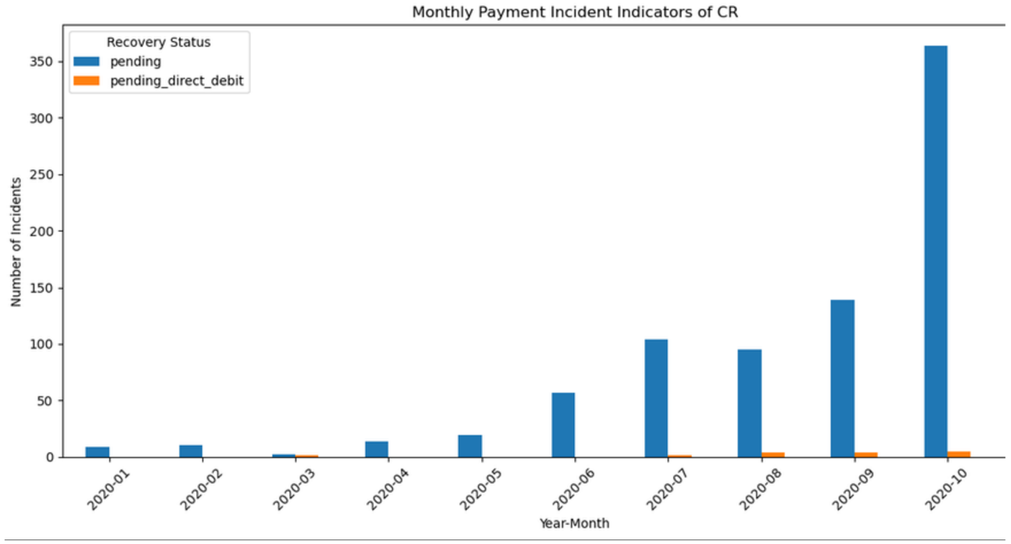


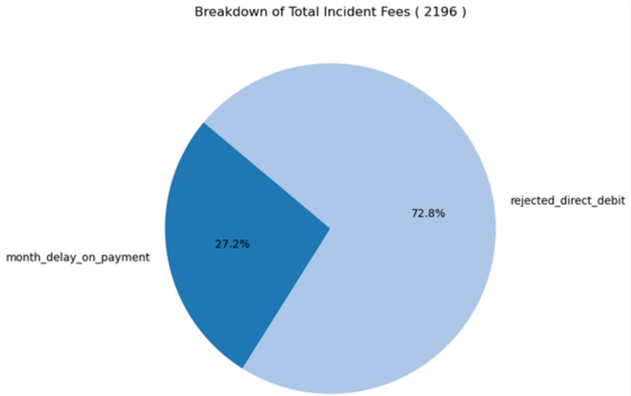
Data Visualization & Analysis

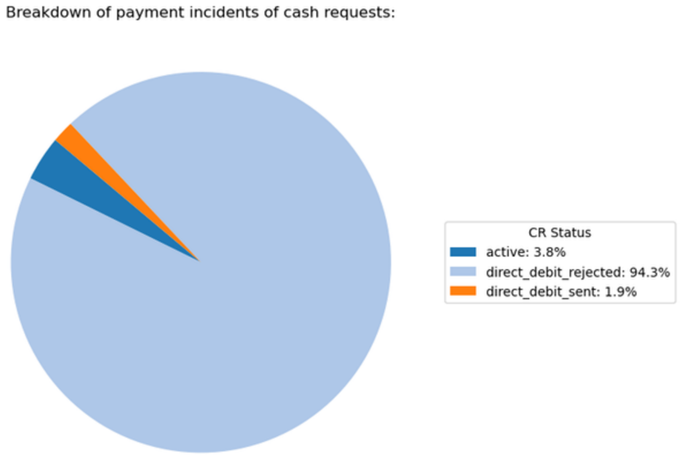
Incident Rate

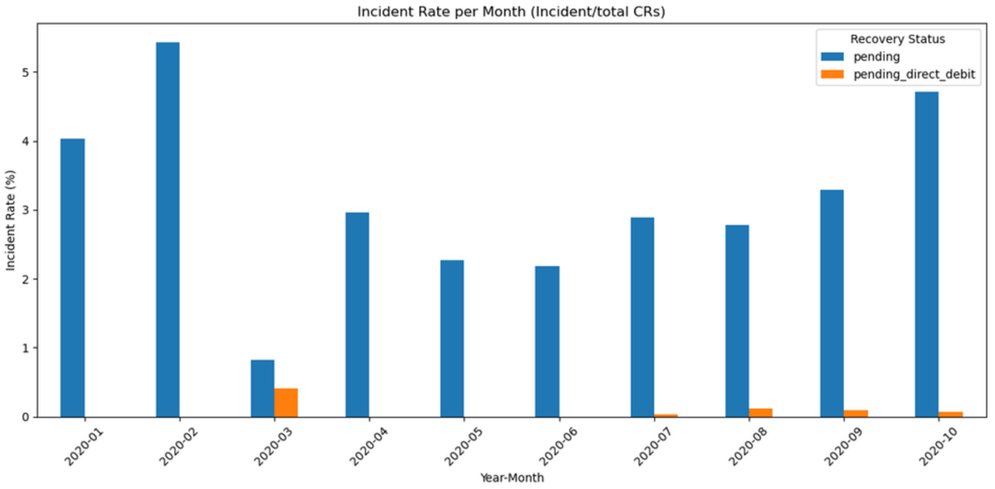
Sustainability of the business→ demands insights into the incidence rate of customer loan reimbursements.

In the top graph we can see that the **amount of payment incidents is increasing** due to the **higher amount of cash requests/users** and a "constant" incident rate.





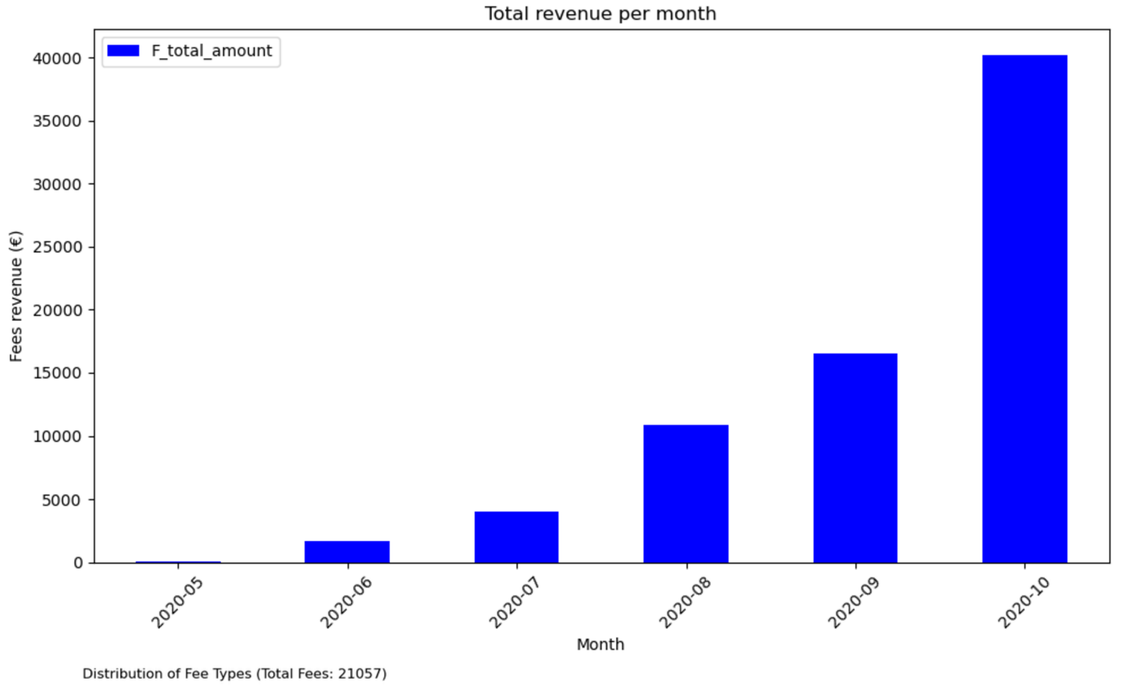




Data Visualization & Analysis

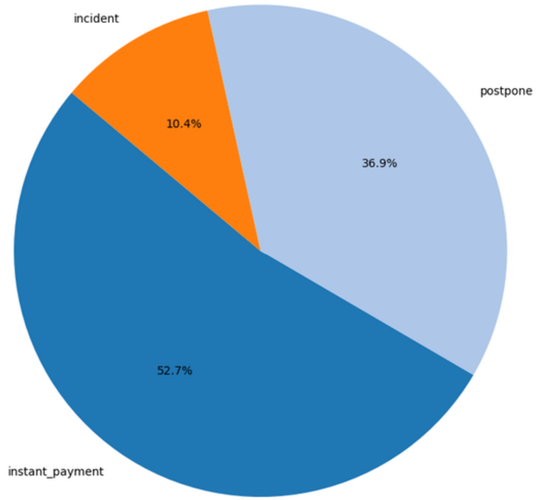
Revenue Generated by the Cohort

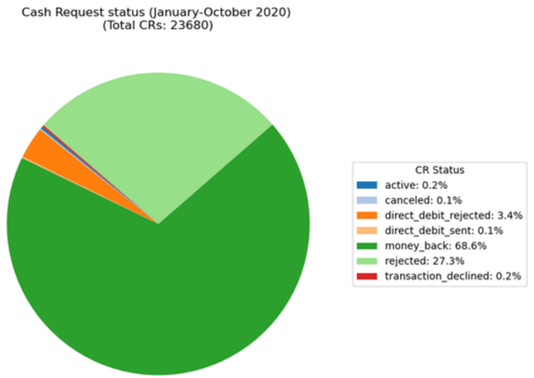
Analysis of the distribution of total revenue fee amounts per month throughout the year
→ Check growth or decline trends and test other hypotheses about what could be affecting Ironhack's growth.
Hypothesis: This analysis showed that October was responsible for the higher fee revenues, following the trend analyzed before on user growth.



We also analyzed what **kind of fees** provided the higher total amounts of revenues. In that way, we can understand weather to charge higher or lower fees depending on their frequencies and returns.

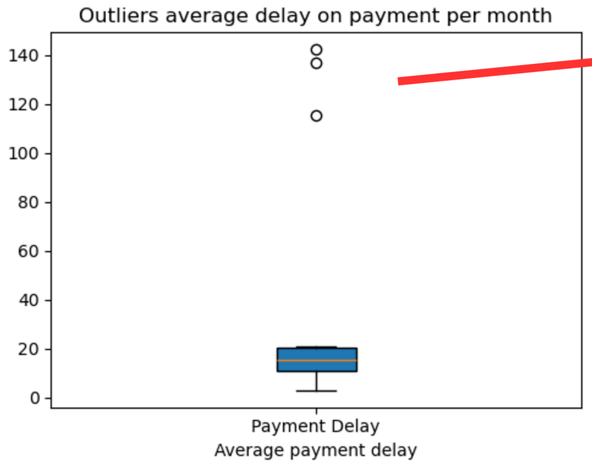
~53% of fees revenues comes from instant_payment (increase in revenue by little increase for the fee)



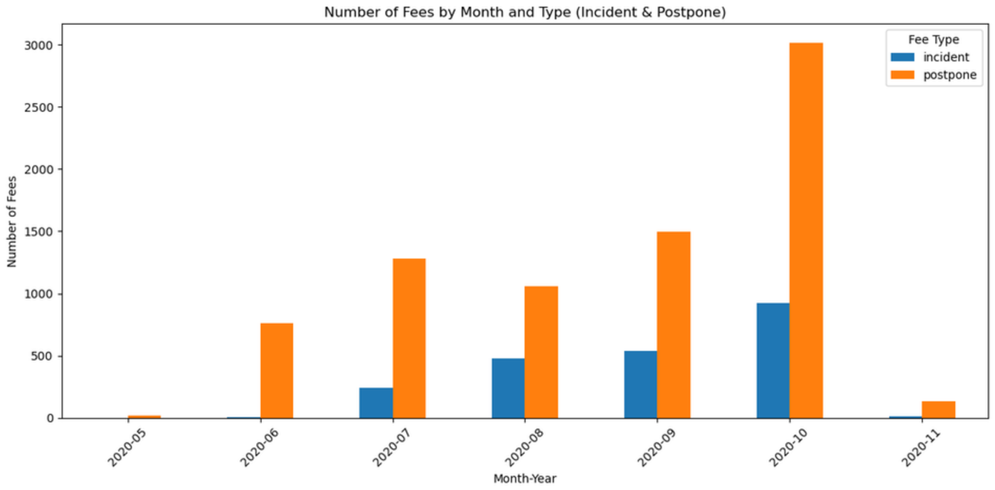


Data Visualization & Analysis

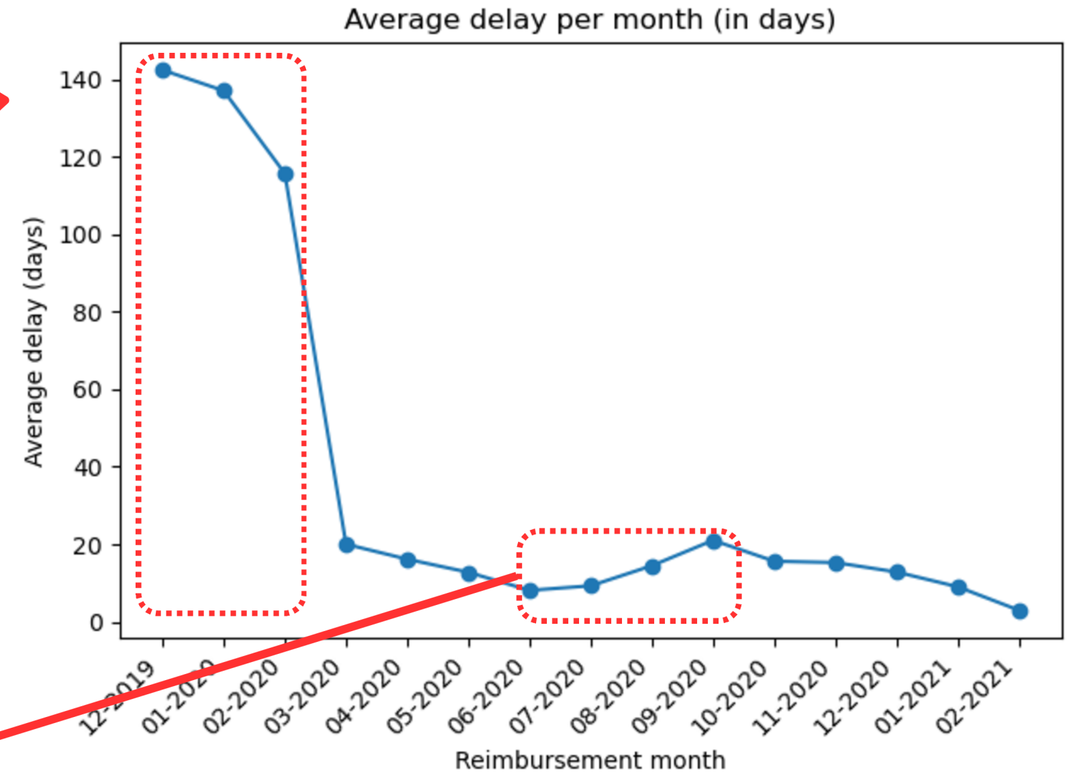
Metrics to understand



Looks like the company announced fees for incidents on february 2020 and decreased the average delay on payments as no fees are recorded till May.



Only money back CR and late payments are considered



From November 2021 is either not enough fees data or users are behaving :)

Data Visualization & Analysis

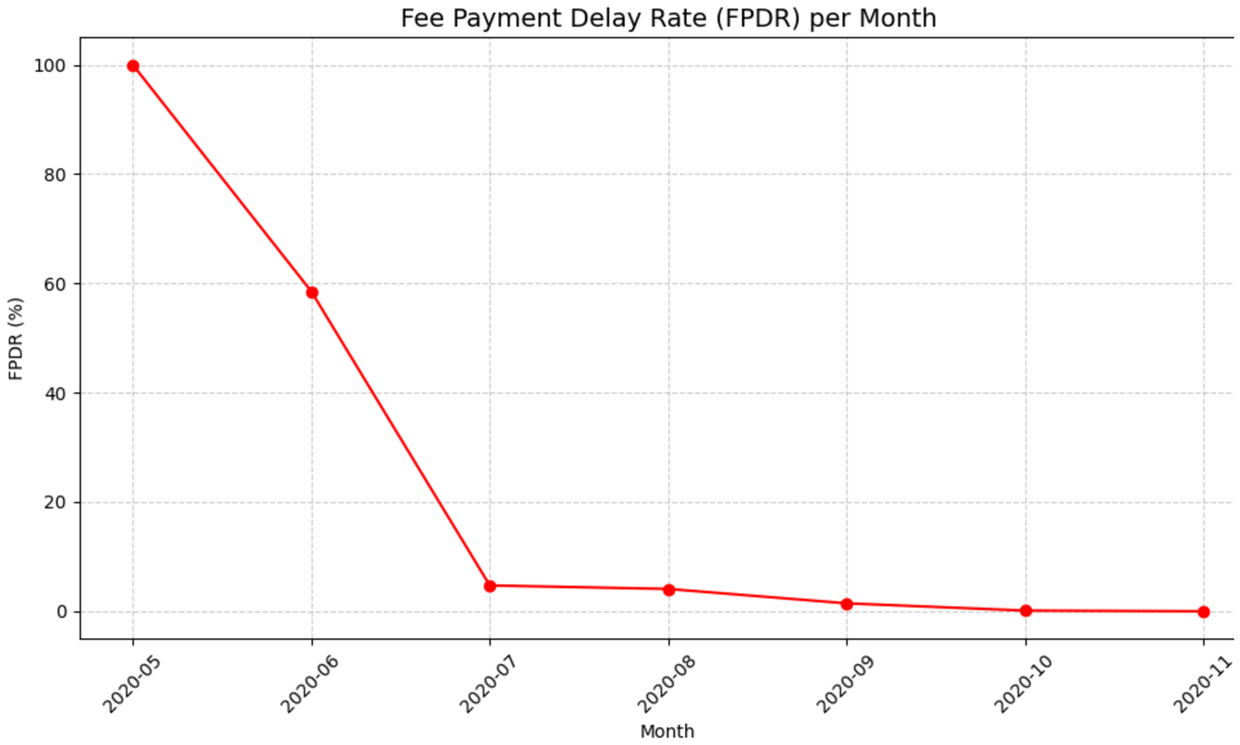
Fee Payment Delay Rate (FPDR) per Month

The **Fee Payment Delay Rate (FPDR)** helps us measure the percentage of accepted fees that remain unpaid over time.

→ Evaluate whether users who agree to pay a fee follow through (**revenue stability and user trust**).

A decreasing FPDR is a positive signal: it indicates that users are paying their accepted fees more consistently, and the platform is becoming more **reliable in terms of fee recovery.**

Contrary to the initial hypothesis, the **Fee Payment Delay Rate (FPDR) has significantly decreased over time**, even as service usage increased.
→ Positive **evolution in user payment behavior** and potentially **better platform processes** or communication.



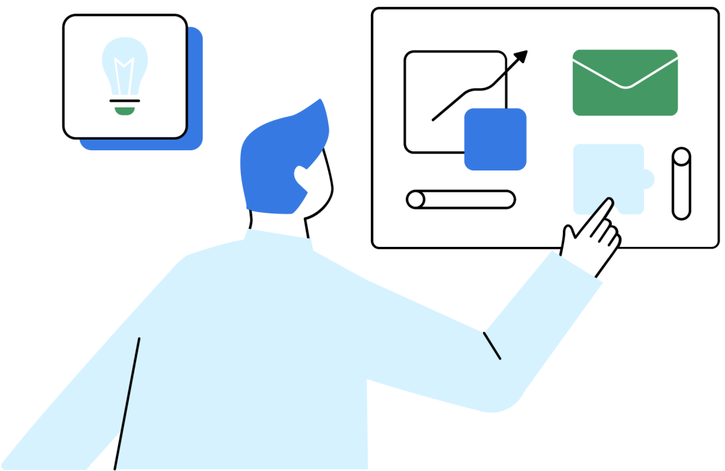
Hypothesis:
As service usage increases, the percentage of unpaid accepted fees might also increase, revealing possible liquidity problems or decreasing user engagement.

FPDR (%) = (accepted fees with no paid_at date) / (Total Accepted Fees) * 100

IRON HACK

06

Observations & Limitations



Point 01

There are sufficient events per user so It's Ideal for our cohort analysis, but understanding the data was our big challenge to the analysis

Point 02

We could have more data on the users to make further analysis as modelling risk per profile of each user or region etc.

Point 04

To understand seasonality of the cash requests more data is required

Point 05

Data structure could be simplified and improve, we can't see different stages of CR_status as it is a live variable

Point 06

We missed more information on the reasonability behind the status of requests rejected and the recovery status.

Point 07

Fees amount are the same regardless the type of fee

Thank You