Exploratory Data Analysis

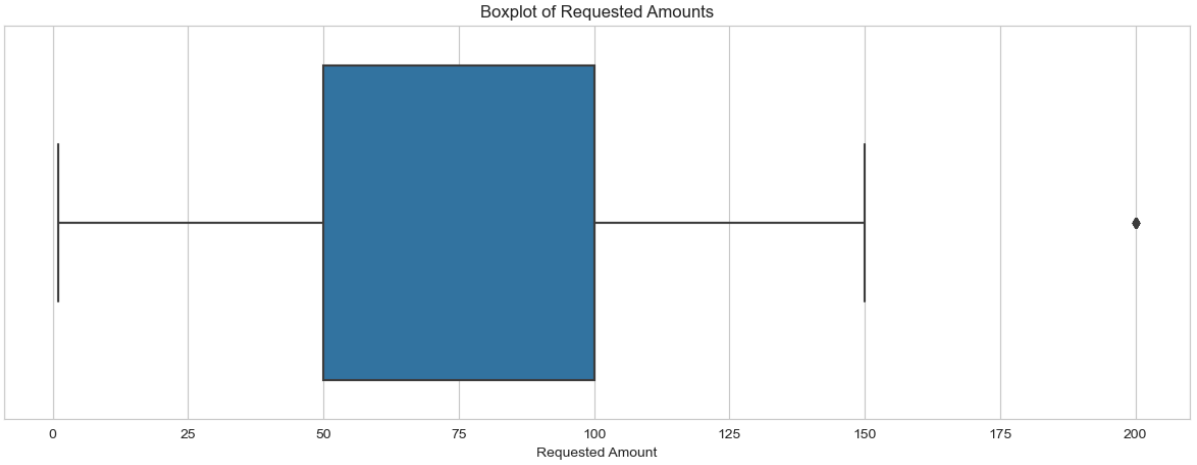
Before diving into cohort analysis, conduct an exploratory data analysis to gain a comprehensive understanding of the dataset. Explore key statistics, distributions, and visualizations to identify patterns and outliers. EDA will help you make informed decisions on data preprocessing and analysis strategies.

For the *Extract - Cash Request* dataset, the statistical summary showed that the file contained 23970 rows and 16 columns, of types float, int and objects. As for the columns with numerical values:

Statistical Summary:

	id	amount	user_id	deleted_account_id
count	23970.0	23970.0	21867.0	2104.0
mean	13911.0	82.7	32581.3	9658.8
std	7788.1	26.5	27618.6	7972.7
min	3.0	1.0	34.0	91.0
25%	7427.2	50.0	10804.0	3767.0
50%	14270.5	100.0	23773.0	6121.5
75%	20607.8	100.0	46965.0	16345.0
max	27010.0	200.0	103719.0	30445.0

- The variables *id, user_id* and *deleted_account_id* have numerical values associated, however, as these represent *id's*, only the count of total values is relevant;
- In a first approach, possibly the concatenation of *user_id* and *deleted_account_id* could represent the entire list of users that made cash requests, this hypothesis needs to be further explored;
- The average *amount* is 82.7$, but this may be impacted by the min and max amount requested (min = 1, max = 200, std = 26.5). This variable was further explored in the boxplot below, showing that the max amount = 200 is an outlier, and the min amount =1 is an amount requested multiple times.



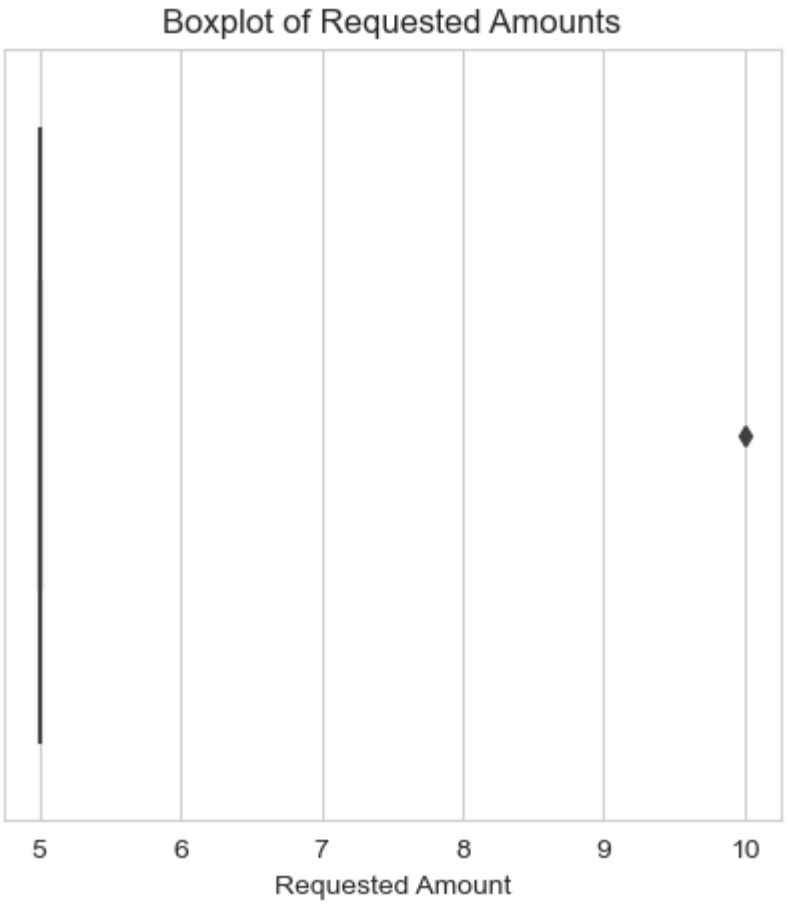
Several columns of type *object* contain information about dates and status, that may allow to see the progression overtime of cash requests and other relevant metrics, depending on the data quality of the dataset, to be analyzed ahead. However, the dates are limited and there isn't enough data to perform an analysis for the entire year of cash requests.

For the *Fees Request* dataset, the statistical summary showed that the file contained 21060 rows and 13 columns, of types float, int and objects. As for the columns with numerical values:

Statistical Summary:

	id	cash_request_id	total_amount
count	21061.0	21057.0	21061.0
mean	10645.0	16318.0	5.0
std	6099.0	6656.0	0.0
min	1.0	1456.0	5.0
25%	5385.0	11745.0	5.0
50%	10652.0	17160.0	5.0
75%	15925.0	21796.0	5.0
max	21193.0	27010.0	10.0

- The variables *id* and *cash_request_id* have numerical values associated, however, as these represent *id's*, only the count of total values is relevant;
- The mean *amount* is 5$, and std = 0, meaning that almost all values for the fee are the same, this is also reinforced by the min value = 5. The max amount = 10 is an outlier, based on the boxplot for this variable:



Data Quality Analysis

Assess the quality of the dataset by identifying missing values, data inconsistencies, and potential errors. Implement data cleaning and preprocessing steps to ensure the reliability of your analysis. Document any data quality issues encountered and the steps taken to address them.

Overview of the DataSet

The dataset for this analysis consisted in three different files: *Extract - Cash Request* and *Extract - Fees*, with relevant data for the analysis, and *Lexique - Data Analyst*, with the description of the information available in each column, for each file.

Cash Request consists of a dataset with 16 columns and 24k rows, with relevant data for all the cash requests that were done by the cohort students since 2020. This dataset stores information for each request (request_id, date of the request creation and update, the status of the request, type of transference, recovery status, and others).

Fees consists of a dataset with 13 columns and 21k rows, with relevant data for all the cases where a cash request was done and a fee was applied due to a specified reason. This dataset stores information for associated cash request and user, as well as the status, type, category and amount of the fee applied, and others.

Both datasets can be connected using the cash request id variable, which is unique for each data request and is the same in both exports.

Missing values, Data Inconsistencies & Potential errors

Both of the datasets have missing values in some of the columns. In some of the cases, the missing values had a direct impact on the analysis. The most critical values missing that may have impact on the analysis are listed below.

On the *Cash request* export, the most critical values missing were for the *user_id* variable, where 8.8% of values were missing in the dataset. After further analysis, it is possible to assume that the user_id's missing are found in the deleted_account_variable. It was decided to remove these 8.8% of data from the analysis, not only because there was no user id associated, but also because it is not relevant for this analysis the data from inactive users.

On *Fees* export, the variable cash_request_id is missing 4 values, even though there is a fee and other info associated with this fee id. It was decided to remove these 4 fees' registrations from the analysis. These represent a total amount of 20$ income, but won't have a significant impact on the final outcome analysis.

Data cleaning and preprocessing

Both files had variables of types int, float and objects. The variables with type float were converted to type int.

For the most important variables that had values missing, it was decided to exclude the correspondent rows from the analysis (already mentioned in the previous topic). For other variables that were to be considered for the analysis, such as *Fee_amount* in the *Fees* export, all the missing values were replaced with “0”.

A new variable was created, “Cohort”, to identify the cohorts of each student, based on the month/year when the cash request id was created (created_at on *Cash requests* dataset).

It was necessary to check if the variable that would be the connection between both datasets had unique or duplicated values. On *Cash requests* dataset, this variable had unique values, but had duplicated values on *Fees* dataset (as each cash request could have several fees associated). Due to this, the *Fees* dataset was reorganized to contain only the relevant variables needed for the analysis, and have only a unique cash request id with the total sum of the fees associated with that cash request id.

After joining all the variables in a unique dataset, all the float variables were converted to int and empty values replaced with 0 (for the numeric variables). Also, some columns had their names updated for a better understanding.

Metrics Calculation

Frequency of Service Usage

Understand how often users from each cohort utilize IronHack Payments' cash advance services over time.

Assumptions:
Frequency of service usage is the number of cash requests made by the users in each cohort overtime.

Calculation:

1. The dataset was grouped by cohort;
2. Returned the count of unique cash requests made by each cohort;

Incident Rate

Determine the incident rate, specifically focusing on payment incidents, for each cohort. Identify if there are variations in incident rates among different cohorts.

Assumptions:
Payment incidents are all cases where *Status* is: rejected, direct debit rejected, transaction declined, or canceled.

Calculation:

1. On column *status,* it was counted the amount of incidents (considering the assumption above) found in the total of status.
2. The dataset was grouped by cohort and:
    a. counted the total of cash request made in each cohort
    b. counted the amount of payment incidents for each cohort
3. Calculated the incident rate by dividing the incident count by the total cash requests and calculating the percentage

Revenue Generated by the Cohort

Calculate the total revenue generated by each cohort over months to assess the financial impact of user behavior.

Assumptions:
Considering that the IronHack Payments provides money advancements for free, the revenue generated is the sum of all fees applied to the users (due to the reasons described in the *reason* variable).

Calculation:
1. column *fees_total_amount* represents the sum of all fees for each cash request;
2. Grouped the dataset by cohort and summed up the fees for each cohort, as the total revenue.

New Relevant Metric: Fees application rate

Propose and calculate a new relevant metric that provides additional insights into user behavior or the performance of IronHack Payments' services.

As an extra metric, it was calculated the *Fees rate*, to understand how many of the cash requests have fees associated or not. All the cohorts with fees associated generated an income to the company.

Calculation:
1. The dataset was grouped by cohort and:
    a. counted the total of cash requests made in each cohort
    b. counted the amount of cash requests with fees associated in each cohort
2. As not all of the cash requests had fees associated, replaced the NaN values with 0
3. Calculated the fees rate by dividing the total fees requests by the total cash requests and calculating the percentage