**Unlocking Insights from Various Cards Transactions**

**Project Overview**

This project presents a comprehensive analysis of a credit card transaction dataset, leveraging Power BI to transform raw data into actionable business intelligence. The primary goal is to provide a holistic view of credit card operations, encompassing transaction patterns, customer demographics, portfolio health, and potential fraud indicators. By creating a suite of interactive dashboards, this project aims to empower stakeholders with data-driven insights for improved decision-making, risk management, and strategic planning within the credit card domain.

Introduction

In today's dynamic financial landscape, the ability to derive meaningful insights from vast volumes of credit card transaction data is paramount. This project addresses the critical need for enhanced data visibility and analytical capabilities within the credit card domain. By transforming raw transaction information into actionable intelligence, this report aims to provide stakeholders with a clear understanding of key operational metrics, customer behaviours, and potential risk factors.

This analysis is designed to facilitate:

Strategic Decision-Making: By presenting clear trends and patterns in credit card usage and portfolio performance.

Risk Mitigation: Through the identification and analysis of suspicious transaction activities.

Customer Understanding: By segmenting and profiling customers based on demographic and behavioural data.

**Dataset Overview**

This project leverages a **multi-faceted credit card transaction dataset**, meticulously structured across three core tables and supplemented by a specialized JSON lookup file. This rich and interconnected dataset provides a comprehensive view of credit card operations, encompassing cardholder demographics, detailed transaction specifics, card account attributes, and merchant categorizations.

Spanning a historical period from **2010 to 2019**, the dataset enables in-depth analysis of trends, patterns, and anomalies, offering a robust foundation for identifying key insights into credit card usage and performance.

Below is a description of each data source and its key attributes:

Table 1: cards\_data (Card Account Details)

This table serves as the central repository for individual credit card account information. It captures static details about each card and its associated client, crucial for understanding the overall portfolio composition and client card holdings.

* id: A unique identifier for each entry/row in this table.
* client\_id: A unique identifier for each customer/cardholder, serving as a key to link with other tables.
* card\_brand: The brand of the credit card (e.g., Visa, Mastercard, American Express, Discover).
* card\_type: The type of card (e.g., Credit, Debit, Prepaid).
* card\_num: An obfuscated or masked identifier for the credit card number.
* expires: The expiry date of the credit card.
* cvv: The CVV (Card Verification Value) associated with the card.
* has\_chip: Indicates whether the card has a chip (e.g., True/False, 1/0).
* num\_cards: The total number of cards associated with a particular client (likely a pre-calculated count for client\_id).
* credit\_limit: The maximum credit limit assigned to this specific card.
* acct\_open\_year: The year when the credit card account was opened.
* pin: Likely a placeholder or indicator related to the PIN, not the actual PIN.
* card\_on\_date: The date the card was issued or activated.

Table 2: transactions\_data (Transaction Details)

This table holds the granular, event-level information for each credit card transaction. It is critical for analyzing spending patterns, identifying suspicious activities, and understanding transaction flow.

* id: A unique identifier for each individual transaction record.
* date: The date on which the transaction occurred.
* client\_id: The identifier for the customer who made the transaction (links to users\_data and cards\_data).
* card\_id: An identifier for the specific card used in the transaction (links to cards\_data).
* amount: The monetary value of the transaction.
* use\_chip: Indicates whether the transaction was processed using a chip card (e.g., True/False, 1/0).
* merchant: The name of the merchant involved in the transaction.
* merchant\_zip: The postal code of the merchant's location.
* mcc: Merchant Category Code, a four-digit number that classifies the business type of the merchant (links to the mcc code json file).
* errors: A field indicating any errors associated with the transaction, which can be crucial for identifying suspicious or failed transactions.

Table 3: users\_data (User/Customer Demographics)

This table contains detailed demographic and personal information for each client. This data is crucial for customer segmentation, profiling, and understanding the characteristics of the cardholder base.

* id: A unique identifier for each entry/row in this table (likely representing client\_id).
* current\_age: The current age of the customer.
* retirement\_birth\_year: The birth year of the customer, potentially used to calculate their retirement age or remaining working years.
* year\_birth: The year of birth of the customer.
* mon\_gender: The gender of the customer.
* address: The customer's address.
* latitude: The geographical latitude coordinate of the customer's address.
* longitude: The geographical longitude coordinate of the customer's address.
* per\_capita\_yearly\_inc: The customer's yearly per capita income.
* total\_debt: The total debt held by the customer.
* credit\_sco: The credit score of the customer.
* num\_credi: (Likely num\_credit\_cards or similar) The number of credit-related accounts or credit cards the customer holds.

Lookup File: mcc code json (Merchant Category Codes)

This JSON file serves as a lookup table, providing descriptions for the Merchant Category Codes (MCCs) found in the transactions\_data table. It allows for rich categorization of transactions by the type of business the merchant operates, enabling more granular analysis of spending habits.

**Data Preparation & Modeling**

The integrity and accuracy of the analysis depend heavily on the quality of the underlying data. Therefore, a rigorous data preparation and modeling phase was undertaken using Power BI Desktop's Power Query Editor and Data Model view. This process involved several critical steps to clean, transform, and integrate the raw data from cards\_data, transactions\_data, users\_data, and the mcc code json file into a cohesive and analytical-ready format.

Data Cleaning & Transformation

Initial data inspection revealed various aspects that required attention to ensure consistency, usability, and correctness of the raw datasets. The key cleaning and transformation steps performed were:

* Data Type Conversion:
  + Date columns, specifically date from transactions\_data, and AccountOpenDate and card\_on\_date from cards\_data, were meticulously converted to the 'Date' data type to enable proper chronological analysis and filtering.
  + Numerical columns, including amount (from transactions\_data), credit\_limit, num\_cards (from cards\_data), current\_age, per\_capita\_yearly\_inc, total\_debt, and credit\_sco (from users\_data), were accurately cast to appropriate numerical data types (e.g., 'Decimal Number' for monetary values and 'Whole Number' for counts and ages) to ensure precise calculations and aggregations.
  + Categorical fields such as card\_brand, card\_type (from cards\_data), mon\_gender, has\_chip, use\_chip (from transactions\_data), and errors (from transactions\_data) were confirmed as 'Text' data types to facilitate correct grouping and slicing.
* Handling Missing Values:
  + A thorough review was conducted to identify missing values across all tables. For instance, any missing values in critical identifier columns like client\_id or card\_id were addressed by either removing the affected rows if they represented a small, insignificant portion of the data, or by investigating the source for potential imputation if feasible.
  + Specifically for the errors column in transactions\_data, transactions where no explicit error was recorded were categorized as 'No Error' to ensure a complete and distinct differentiation from actual error types during analysis.
* Duplicate Removal:
  + Duplicate rows were meticulously identified and removed from each table based on their unique identifier columns (id in cards\_data, transactions\_data, and users\_data). This crucial step ensured that each record represented a distinct entity or event, preventing overcounting and skewed aggregations in the final reports.
* Standardization of Text Fields:
  + Text-based columns like card\_brand, merchant (from transactions\_data), mon\_gender (from users\_data), and the Description from the mcc code json were subjected to 'Clean' and 'Trim' transformations to eliminate any extraneous spaces or non-printable characters.
  + Consistency in casing was applied (e.g., 'Proper Case' for names, 'Capitalize Each Word' for descriptions) to enhance readability and ensure accurate grouping and filtering across visualizations.
* Geographical Data Preparation:
  + To enable accurate plotting on map visuals, geographical columns such as merchant\_zip (from transactions\_data) and latitude, longitude, address (from users\_data) were prepared. The 'Data Category' for these columns in Power BI Desktop was explicitly set (e.g., 'Zip Code', 'Latitude', 'Longitude', 'Address') to ensure Power BI's built-in mapping capabilities could correctly interpret and display the location data.

Data Modeling

Following the extensive cleaning phase, the disparate data sources were integrated into a robust and efficient data model within Power BI Desktop. This structured approach is fundamental for enabling seamless cross-table analysis, complex calculations, and hierarchical reporting within the dashboards.

* Establishing Relationships:
  + Clear and precise relationships were defined between the tables using their common key columns, adhering to a star schema design for optimal query performance:
    - cards\_data[client\_id] was linked to users\_data[id] with a Many-to-one cardinality, and a single cross-filter direction from users\_data to cards\_data.
    - transactions\_data[client\_id] was linked to users\_data[id] with a Many-to-one cardinality, and a single cross-filter direction from users\_data to transactions\_data.
    - transactions\_data[card\_id] was linked to cards\_data[id] with a Many-to-one cardinality, and a single cross-filter direction from cards\_data to transactions\_data.
    - transactions\_data[mcc] was linked to the mcc code json[MCC Code] with a Many-to-one cardinality, and a single cross-filter direction from the mcc code json to transactions\_data.
  + These relationships ensure that filters and slicers applied to one table correctly propagate and filter data in related tables, maintaining data integrity across the entire model.
* Schema Design:
  + The data model was meticulously structured following a Star Schema paradigm. The transactions\_data table serves as the central **fact table**, containing the quantitative measures (amount) and foreign keys to other tables.
  + cards\_data, users\_data, and mcc code json function as **dimension tables**, providing descriptive attributes about the cards, users, and merchant categories, respectively. This design significantly optimizes analytical query performance, simplifies the creation of insightful DAX measures, and enhances the overall user experience within the dashboards.

**Key Dashboards & Insights**

This project features a suite of interactive dashboards, meticulously designed in Power BI to translate complex credit card transaction data into clear, actionable insights. Each dashboard serves a distinct analytical purpose, allowing stakeholders to delve into specific aspects of the credit card ecosystem.

1. Suspicious Activity Dashboard

This dashboard is specifically designed to provide a vigilant overview of potential fraudulent or erroneous credit card transactions. By highlighting anomalies and suspicious patterns, it serves as a critical tool for risk assessment and rapid identification of unusual financial activities.

List of Visuals on this dashboard:

*Cards*: Total transaction amount, total number of transactions, Average transaction value, transaction error count.

*Slicer*: Time frame slicer to select specific time period

*Home button*: To return back to home page of dashboard

*Top 5 Clients by Unusual Transaction*:

This crucial bar and line chart identifies clients whose transaction behavior significantly deviates from the norm, indicating a higher likelihood of unusual or suspicious activity. This visual enables immediate focus on high-risk accounts for further investigation. The bars represent the count of unusual transactions for each client, while the overlaid line typically shows the total monetary amount involved in those unusual transactions for the selected clients.

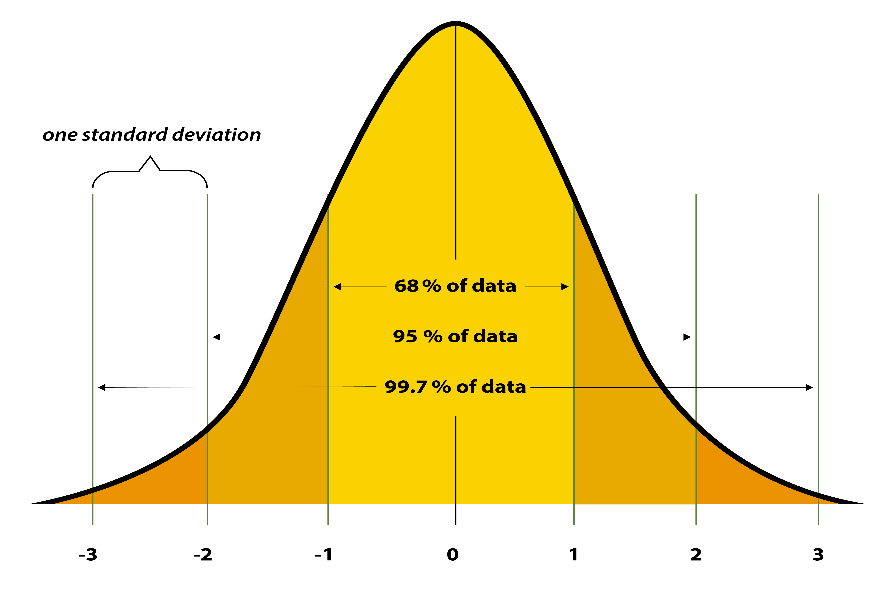
Logic for "Unusual Transaction": An "unusual transaction" in this context is defined statistically as any transaction whose amount falls more than three standard deviations (3 sigma) away from the average transaction amount. This methodology is based on the principles of the Normal Distribution, where data points are typically clustered around the mean.

Standard Deviation: The standard deviation is a measure of the amount of variation or dispersion of a set of values. A low standard deviation indicates that the values tend to be close to the mean, while a high standard deviation indicates that the values are spread out over a wider range.

The Empirical Rule (68-95-99.7 Rule): For data that follows a normal (bell-shaped) distribution:

Approximately 68% of the data falls within 1 standard deviation of the mean.

Approximately 95% of the data falls within 2 standard deviations of the mean.

Approximately 99.7% of the data falls within 3 standard deviations of the mean.

Application for Outlier Detection: By setting the threshold at three standard deviations, we are identifying transactions that are in the extreme tails of the distribution. Statistically, less than 0.3% of data points are expected to fall outside 3 sigma in a truly normal distribution. Therefore, transactions exceeding this threshold are considered highly atypical and are flagged as "unusual," warranting closer inspection for potential fraud or significant anomalies. This approach helps in focusing investigative efforts on transactions that are statistically rare and potentially indicative of illicit activity.

DAX Measure: Unusual\_Transaction\_Count

Unusual\_Transaction\_Count =

VAR avgAmt =

CALCULATE(

AVERAGE('transactions\_data'[amount])

)

VAR sdAmt =

CALCULATE(

STDEV.P('transactions\_data'[amount])

)

RETURN

IF(

ISBLANK(avgAmt) || ISBLANK(sdAmt) || sdAmt = 0,

BLANK(),

CALCULATE(

COUNTROWS('transactions\_data'),

FILTER(

'transactions\_data',

ABS('transactions\_data'[amount] - avgAmt) > 3 \* sdAmt

)

)

)

Explanation of the DAX Measure:

VAR avgAmt: Calculates the AVERAGE of all amount values in the transactions\_data table within the current filter context (e.g., for a specific client, or the entire dataset).

VAR sdAmt: Calculates the STDEV.P (Population Standard Deviation) of all amount values in the transactions\_data table, reflecting the dispersion of amounts around their average.

IF(ISBLANK(avgAmt) || ISBLANK(sdAmt) || sdAmt = 0, BLANK(), ...): This conditional statement serves as an error trap, returning BLANK() if the average or standard deviation cannot be calculated or if the standard deviation is zero, thus preventing errors in the visual.

CALCULATE(COUNTROWS('transactions\_data'), FILTER(...)): This counts the number of rows (transactions) in transactions\_data that meet the specified filtering condition.

FILTER('transactions\_data', ABS('transactions\_data'[amount] - avgAmt) > 3 \* sdAmt): This is the core of the anomaly detection. It filters the transactions\_data table to include only those transactions where the absolute difference between the individual amount and the calculated avgAmt is greater than three times the sdAmt. This precisely identifies the transactions that fall outside the 3 sigma range.

Insight Application: By aggregating and sorting clients based on this Unusual\_Transaction\_Count, the dashboard effectively brings the most statistically atypical transaction behaviors to the forefront. This allows analysts to prioritize investigations into potential fraud or significant anomalies, optimizing resource allocation for fraud detection efforts.

*Error Breakdown Pie chart:* Visual representation of transaction error breakdown for example: Technical glitch

*Potential Fraud Risk by Location (Map Visual)***:**

This powerful map visual provides a crucial geographical perspective on potential fraud risk. It dynamically colors regions based on a calculated Potential Fraud Risk Index, employing a **red-yellow-green color scheme**. **Green** indicates areas with a lower or minimal risk, **yellow** signifies moderate risk, and **red** highlights regions with a higher or significant potential fraud risk. This intuitive color grading allows for a quick visual identification of high-risk geographical areas, drawing immediate attention to specific locations that may warrant increased scrutiny or localized fraud prevention efforts.

* Logic for "Potential Fraud Risk Index": The Potential Fraud Risk Index is a calculated ratio designed to quantify the concentration of unusual transactions within a specific geographic area relative to the total transaction volume in that same area. It's a key performance indicator that helps to normalize the count of unusual transactions, preventing areas with simply high overall transaction volumes from being falsely flagged as high-risk. Instead, it highlights locations where a proportionately higher number of transactions are statistically unusual.
* DAX Measure: Potential Fraud Risk Index

Potential Fraud Risk Index =

VAR TotalTransactions = COUNTROWS('transactions\_data')

VAR UnusualTransactions = [Unusual\_Transaction\_Count] // This refers to the previously defined measure

RETURN

DIVIDE(UnusualTransactions, TotalTransactions, 0)

Explanation of the DAX Measure:

* + VAR TotalTransactions = COUNTROWS('transactions\_data'): This variable calculates the total number of transactions occurring within the current filter context (e.g., for a specific state or country currently highlighted on the map).
  + VAR UnusualTransactions = [Unusual\_Transaction\_Count]: This variable retrieves the value of the Unusual\_Transaction\_Count measure (which we previously defined based on the 3-standard deviation rule for transaction amounts) for the current filter context. This represents the count of transactions deemed statistically unusual in that specific location.
  + RETURN DIVIDE(UnusualTransactions, TotalTransactions, 0): This part defines the final calculation. The DIVIDE function is used for safe division, calculating the ratio of UnusualTransactions to TotalTransactions. If TotalTransactions is zero, it returns 0 instead of an error, ensuring robust calculations across all geographical segments.

*MCC Unusual Transaction Rate (Treemap Visual)*:

This treemap visual provides a powerful way to identify Merchant Category Codes (MCCs) that exhibit a higher proportion of unusual transactions. Each rectangle in the treemap represents an MCC, and its size can indicate the total number of transactions or the total amount. The treemap utilizes a red-yellow-green color scheme to highlight the "Unusual Transaction Rate": Green signifies MCCs with a lower or minimal rate of unusual activity, yellow indicates a moderate rate, and red prominently highlights MCCs with a higher or significant rate of unusual transactions. This intuitive color grading allows analysts to quickly pinpoint specific industries or types of merchants that are more susceptible to suspicious activities, informing targeted fraud prevention strategies or risk assessments for particular merchant categories.

* Logic for "MCC Unusual Transaction Rate": This measure calculates the percentage of unusual transactions within a specific Merchant Category Code (MCC) relative to the total number of transactions for that same MCC. Similar to the "Potential Fraud Risk Index," it normalizes the count of unusual transactions, ensuring that MCCs with high overall transaction volumes aren't mistakenly flagged as high-risk if their proportion of unusual transactions is low. Instead, it focuses on the *rate* of unusual activity within each category.
* DAX Measure: MCC Unusual Transaction Rate

Code snippet

MCC Unusual Transaction Rate =

VAR TotalTransactionsForMCC = COUNTROWS('transactions\_data')

VAR UnusualTransactionsForMCC =

CALCULATE(

[Unusual\_Transaction\_Count],

KEEPFILTERS('transactions\_data'[mcc])

)

RETURN

DIVIDE(UnusualTransactionsForMCC, TotalTransactionsForMCC, 0)

Explanation of the DAX Measure:

* + VAR TotalTransactionsForMCC = COUNTROWS('transactions\_data'): This variable calculates the total number of transactions that fall under the specific MCC currently being evaluated by the treemap (i.e., within its current filter context).
  + VAR UnusualTransactionsForMCC = CALCULATE([Unusual\_Transaction\_Count], KEEPFILTERS('transactions\_data'[mcc])): This variable re-evaluates the Unusual\_Transaction\_Count measure (which flags transactions based on the 3-standard deviation rule for amount outliers) specifically for the current MCC.
    - The CALCULATE function is used to change the filter context.
    - KEEPFILTERS('transactions\_data'[mcc]) is crucial here. It ensures that the filter context applied by the treemap for the current MCC is *preserved* when Unusual\_Transaction\_Count is calculated. This means Unusual\_Transaction\_Count will correctly count unusual transactions *only within that specific MCC*, rather than recalculating the global average and standard deviation.
  + RETURN DIVIDE(UnusualTransactionsForMCC, TotalTransactionsForMCC, 0): This final step computes the ratio by dividing the count of unusual transactions for that MCC by the total transactions for that MCC. The DIVIDE function again provides robustness by returning 0 if the total number of transactions for an MCC is zero, preventing division-by-zero errors.

*Refund Rate by Card Type:* This pie chart breaks down the overall Refund Rate by different card types (Credit, Debit, Prepaid). It visually represents the percentage contribution of each card type to the total refunds, alongside the overall refund rate KPI. While refunds are a standard part of business, understanding their distribution across card types can highlight specific operational or customer behavior patterns. For instance, a disproportionately high refund rate for a particular card type could indicate issues specific to that card's usage, associated merchant categories, or potential avenues for refund fraud schemes.

* Logic for "Refund Rate": The Refund Rate is calculated as the total count of transactions where the amount is negative (signifying a refund or credit) divided by the total number of all transactions. This provides a normalized percentage that can be easily compared across different segments (like card types) or over time.
* DAX Measure: Refund Rate

Code snippet

Refund Rate =

VAR TotalRefunds = CALCULATE(COUNTROWS('transactions\_data'), transactions\_data[amount] < 0)

VAR TotalTransactions = COUNTROWS('transactions\_data')

RETURN DIVIDE(TotalRefunds, TotalTransactions, 0)

Explanation of the DAX Measure:

* + VAR TotalRefunds = CALCULATE(COUNTROWS('transactions\_data'), transactions\_data[amount] < 0): This variable calculates the total number of refund transactions by counting rows in the transactions\_data table where the amount is less than 0 (i.e., negative).
  + VAR TotalTransactions = COUNTROWS('transactions\_data'): This variable counts the total number of all transactions within the current filter context (e.g., all transactions for 'Credit' cards if that segment is selected).
  + RETURN DIVIDE(TotalRefunds, TotalTransactions, 0): This safely calculates the ratio of TotalRefunds to TotalTransactions, returning 0 if TotalTransactions is zero to prevent division-by-zero errors.

1. Demographics Dashboard

*Cards:* Average Age of customer, Total number of customers, Average income of per customer, average number of cards with each individual.

*Slicer*: Time period

*Total spending by age group:* made a calculated column for age groups using dax

Age Group =

SWITCH(

    TRUE(),

    'users\_data'[current\_age] >= 18 && 'users\_data'[current\_age] <= 24, "18-24",

    'users\_data'[current\_age] >= 25 && 'users\_data'[current\_age] <= 34, "25-34",

    'users\_data'[current\_age] >= 35 && 'users\_data'[current\_age] <= 44, "35-44",

    'users\_data'[current\_age] >= 45 && 'users\_data'[current\_age] <= 54, "45-54",

    'users\_data'[current\_age] >= 55 && 'users\_data'[current\_age] <= 64, "55-64",

    'users\_data'[current\_age] >= 65, "65+",

    "Unknown"

)

Customer distribution by credit score range: similar to age group, made a calculated column for credit ratings

Credit Score Range =

SWITCH(

    TRUE(),

    'users\_data'[credit\_score] >= 300 && 'users\_data'[credit\_score] <= 579, "1 - Very Poor (300-579)",

    'users\_data'[credit\_score] >= 580 && 'users\_data'[credit\_score] <= 669, "2 - Fair (580-669)",

    'users\_data'[credit\_score] >= 670 && 'users\_data'[credit\_score] <= 739, "3 - Good (670-739)",

    'users\_data'[credit\_score] >= 740 && 'users\_data'[credit\_score] <= 799, "4 - Very Good (740-799)",

    'users\_data'[credit\_score] >= 800 && 'users\_data'[credit\_score] <= 850, "5 - Excellent (800-850)",

    "6 - Unknown"

)

*Gender demographics by number of customers and total transaction value represented using pie and donut chart.*

*Average transaction value by location:* This map chart shows how much transaction are recorded in a particular location.

*Total debt vs total credit score:* scatter chart

Customer distribution by per capita income range: Similar to previous calculated columns, made a calculated column for income range.

Per Capita Income Range =

SWITCH(

    TRUE(),

    'users\_data'[per\_capita\_income] < 10000, "1 - < $10,000",

    'users\_data'[per\_capita\_income] >= 10000 && 'users\_data'[per\_capita\_income] < 20000, "2 - $10,000 - $19,999",

    'users\_data'[per\_capita\_income] >= 20000 && 'users\_data'[per\_capita\_income] < 30000, "3 - $20,000 - $29,999",

    'users\_data'[per\_capita\_income] >= 30000 && 'users\_data'[per\_capita\_income] < 45000, "4 - $30,000 - $44,999",

    'users\_data'[per\_capita\_income] >= 45000 && 'users\_data'[per\_capita\_income] < 60000, "5 - $45,000 - $59,999",

    'users\_data'[per\_capita\_income] >= 60000, "6 - $60,000+",

    "7 - Unknown"

)

1. **Credit card Portfolio overview Dashboard**

*Cards*: Total number of cards, Total credit limit, Average credit limit, Cards with old pins which were changed before 2010.

*Slicer*: Cards Brand

*Cards type distribution:* This visual shows the number of clients according to different types of cards like, debit, credit and prepaid.

*Average credit by number of cards:* This visual shows the relationship between the number of cards that each customer has and average credit limits

*Top clients by total credit limit:* This table visual shows the client number who have maximum credit limit.

Geographic distribution of clients (map chart) : This visual shows the number of clients in each location.

Average credit limit by brand: this visual shows the average credit limit that each card brand offers.

Cards issued over time: This visual shows the number of cards issued per year.