Data Cleaning, Modeling and DAX in Power BI

Project : SBI Banking Insights DAX Analytics

Data Importing and Initial Examination

- Imported both datasets into Power BI.
- Conducted a preliminary examination to identify any data quality issues or inconsistencies.

Merging and Relating Datasets

- Merged the datasets using the "Account Number" column.
- Ensured the merge was accurate and retained all essential information.

Cleaning: Handling Missing and Irrelevant Data

- Identified and addressed any missing data in both datasets, removing entries where necessary, though missing data was minimal.
- Eliminated duplicate entries and removed irrelevant data points to enhance overall data quality.

Data Type Conversion

- Transformed date columns to the appropriate data format.
- Split account holder details into separate columns, including years at current residence and city of residence.

Calculated Columns Using DAX

Categorizing transactions as "Credit" for Payments and Deposits, "Debit" for Withdrawals and Transfers, or returning BLANK for other types.

```
TransactionCategory =
IF(
    OR([TransactionType] = "Payment", [TransactionType] = "Deposit"),
    "Credit",
    IF(
        OR([TransactionType] = "Withdrawal", [TransactionType] = "Transfer"),
        "Debit",
        BLANK()
    )
)
```

nsactionType	Amount	TransactionDate	BranchCode	Currency	TransactionTime(in Hours)	TransactionCategory
ısfer	3676.6907470515	01 January 2023	9	GBP	11	Debit
ment	9402.57933438732	05 January 2023	384	GBP	23	Credit
›osit	3434.0443253437	02 January 2023	407	JPY	4	Credit
hdrawal	8367.14781395114	03 January 2023	123	USD	23	Debit
ısfer	7043.06762499775	04 January 2023	458	USD	6	Debit
ısfer	2888.70896203098	06 January 2023	153	EUR	22	Debit
hdrawal	9986.8297353486	07 January 2023	418	USD	5	Debit
ment	9295.62355676316	08 January 2023	265	JPY	2	Credit
hdrawal	7647.59072654907	09 January 2023	183	USD	12	Debit
ment	5105.34037878721	10 January 2023	134	USD	18	Credit
ısfer	2239.7691993294	11 January 2023	152	USD	5	Debit
hdrawal	3613.14646768208	12 January 2023	459	JPY	1	Debit
hdrawal	872.021160696059	13 January 2023	181	GBP	10	Debit
ment	5098.61311330898	14 January 2023	97	USD	22	Credit
ısfer	3384.27925842022	15 January 2023	90	JPY	18	Debit
ısfer	4459.13605751757	16 January 2023	180	USD	10	Debit
hdrawal	620.608874721773	17 January 2023	420	USD	3	Debit
ment	9930.37894342699	18 January 2023	384	GBP	15	Credit
ısfer	235.820340229885	19 January 2023	320	USD	4	Debit
ısfer	1314.07317783956	20 January 2023	75	EUR	4	Debit
ment	1640.85646835357	21 January 2023	215	USD	16	Credit
›osit	1456.81339196224	22 January 2023	278	USD	21	Credit

Calculating the amount in USD by multiplying the original amount by the related currency conversion rate, returning BLANK if the conversion rate is not available.

```
AmountInUSD =
IF(
    RELATED('Currency Rates'[RateToUSD]) <> BLANK(),
    [Amount] * RELATED('Currency Rates'[RateToUSD]),
    BLANK()
)
```

Hours)	TransactionCategory	AmountInUSD	RateToUSD	TransactionValue	F
11	Debit	4779.69797116695	1.3	Low Value	
23	Credit	12223.3531347035	1.3	High Value	
4	Credit	30.9063989280933	0.009	Low Value	
23	Debit	8367.14781395114	1	High Value	
6	Debit	7043.06762499775	1	Low Value	
22	Debit	3177.57985823408	1.1	Low Value	
5	Debit	9986.8297353486	1	High Value	
2	Credit	83.6606120108684	0.009	Low Value	
12	Debit	7647.59072654907	1	Low Value	
18	Credit	5105.34037878721	1	Low Value	
5	Debit	2239.7691993294	1	Low Value	
1	Debit	32.5183182091387	0.009	Low Value	
10	Debit	1133.62750890488	1.3	Low Value	
22	Credit	5098.61311330898	1	Low Value	
18	Debit	30.458513325782	0.009	Low Value	
10	Debit	4459.13605751757	1	Low Value	
3	Debit	620.608874721773	1	Low Value	
15	Credit	12909.4926264551	1.3	High Value	
4	Debit	235.820340229885	1	Low Value	
4	Debit	1445.48049562351	1.1	Low Value	
16	Credit	1640.85646835357	1	Low Value	
21	Credit	1456.81339196224	1	Low Value	

Classifying transaction values as "High Value" if the amount exceeds 8000, and "Low Value" otherwise.

```
1  TransactionValue = IF('Transaction Info'[AmountInUSD]>=8000,"High Value","Low Value")
```

Hours)	TransactionCategory	AmountInUSD	RateToUSD	TransactionValue	FinalSummativeAmoun
11	Debit	4779.69797116695	1.3	Low Value	-4779.6979711
23	Credit	12223.3531347035	1.3	High Value	12223.353134
4	Credit	30.9063989280933	0.009	Low Value	30.906398928
23	Debit	8367.14781395114	1	High Value	-8367.1478139
6	Debit	7043.06762499775	1	Low Value	-7043.0676249
22	Debit	3177.57985823408	1.1	Low Value	-3177.5798582
5	Debit	9986.8297353486	1	High Value	-9986.829735
2	Credit	83.6606120108684	0.009	Low Value	83.660612010
12	Debit	7647.59072654907	1	Low Value	-7647.5907265
18	Credit	5105.34037878721	1	Low Value	5105.3403787
5	Debit	2239.7691993294	1	Low Value	-2239.769199
1	Debit	32.5183182091387	0.009	Low Value	-32.51 -2239.7
10	Debit	1133.62750890488	1.3	Low Value	-1133.6275089
22	Credit	5098.61311330898	1	Low Value	5098.6131133
18	Debit	30.458513325782	0.009	Low Value	-30.45851332
10	Debit	4459.13605751757	1	Low Value	-4459.1360575
3	Debit	620.608874721773	1	Low Value	-620.60887472
15	Credit	12909.4926264551	1.3	High Value	12909.492626
4	Debit	235.820340229885	1	Low Value	-235.82034022
4	Debit	1445.48049562351	1.1	Low Value	-1445.4804956
16	Credit	1640.85646835357	1	Low Value	1640.8564683
21	Credit	1456.81339196224	1	Low Value	1456.8133919
11	Debit	6636.94249242337	1.3	Low Value	-6636.9424924
19	Credit	9194.51203877382	1.1	High Value	9194.5120387
19	Credit	7691.92215090605	1	Low Value	7691.9221509
21	Credit	2003.44861871175	1.3	Low Value	2003.4486187

Identifying transaction types as "Rare" if they occur fewer than 3 times, otherwise classifying them as "Common”.

```
RareTransaction = IF(COUNTROWS(FILTER('Transaction Info', [TransactionType] = EARLIER([TransactionType]))) < 3, "Rare", "Common")
```

	TransactionCategory	AmountInUSD	RateToUSD	TransactionValue	FinalSummativeAmount	ZScore	RareTransaction	Unusu
11	Debit	4779.69797116695	1.3	Low Value	-4779.69797116695	0.151875413393371	Common	Norm
23	Credit	12223.3531347035	1.3	High Value	12223.3531347035	2.16504494683221	Common	Unusu
4	Credit	30.9063989280933	0.009	Low Value	30.9063989280933	-1.13245631663001	Common	Norm
23	Debit	8367.14781395114	1	High Value	-8367.14781395114	1.12211712496693	Common	Norm
6	Debit	7043.06762499775	1	Low Value	-7043.06762499775	0.764013768797425	Common	Norm
22	Debit	3177.57985823408	1.1	Low Value	-3177.57985823408	-0.281424531736623	Common	Norm
5	Debit	9986.8297353486	1	High Value	-9986.8297353486	1.56016727994883	Common	Norm
2	Credit	83.6606120108684	0.009	Low Value	83.6606120108684	-1.11818870586547	Common	Norm
12	Debit	7647.59072654907	1	Low Value	-7647.59072654907	0.927509721197895	Common	Norm
18	Credit	5105.34037878721	1	Low Value	5105.34037878721	0.239946845964316	Common	Norm
5	Debit	2239.7691993294	1	Low Value	-2239.7691993294	-0.535059581340949	Common	Norm
1	Debit	32.5183182091387	0.009	Low Value	-32.5183182091387	-1.13202036591552	Common	Norm
10	Debit	1133.62750890488	1.3	Low Value	-1133.62750890488	-0.834220504557493	Common	Norm
22	Credit	5098.61311330898	1	Low Value	5098.61311330898	0.238127427196708	Common	Norm
18	Debit	30.458513325782	0.009	Low Value	-30.458513325782	-1.13257744927644	Common	Norm
10	Debit	4459.13605751757	1	Low Value	-4459.13605751757	0.065178022936453	Common	Norm
3	Debit	620.608874721773	1	Low Value	-620.608874721773	-0.972968668159034	Common	Norm
15	Credit	12909.4926264551	1.3	High Value	12909.4926264551	2.35061441373202	Common	Unusu
4	Debit	235.820340229885	1	Low Value	-235.820340229885	-1.07703643291126	Common	Norm
4	Debit	1445.48049562351	1.1	Low Value	-1445.48049562351	-0.74987848199185	Common	Norm
16	Credit	1640.85646835357	1	Low Value	1640.85646835357	-0.697038184150963	Common	Norm
21	Credit	1456.81339196224	1	Low Value	1456.81339196224	-0.746813449943278	Common	Norm

Classifying transactions as "Unusual" if the amount exceeds $10,000; otherwise, labeling them as "Normal."

```
UnusualTransaction = IF([AmountInUSD] > 10000, "Unusual", "Normal")
```

	TransactionCategory	AmountInUSD	RateToUSD	TransactionValue	FinalSummativeAmount	ZScore	RareTransaction	UnusualTransaction
11	Debit	4779.69797116695	1.3	Low Value	-4779.69797116695	0.151875413393371	Common	Normal
23	Credit	12223.3531347035	1.3	High Value	12223.3531347035	2.16504494683221	Common	Unusual
4	Credit	30.9063989280933	0.009	Low Value	30.9063989280933	-1.13245631663001	Common	Normal
23	Debit	8367.14781395114	1	High Value	-8367.14781395114	1.12211712496693	Common	Normal
6	Debit	7043.06762499775	1	Low Value	-7043.06762499775	0.764013768797425	Common	Normal
22	Debit	3177.57985823408	1.1	Low Value	-3177.57985823408	-0.281424531736623	Common	Normal
5	Debit	9986.8297353486	1	High Value	-9986.8297353486	1.56016727994883	Common	Normal
2	Credit	83.6606120108684	0.009	Low Value	83.6606120108684	-1.11818870586547	Common	Normal
12	Debit	7647.59072654907	1	Low Value	-7647.59072654907	0.927509721197895	Common	Normal
18	Credit	5105.34037878721	1	Low Value	5105.34037878721	0.239946845964316	Common	Normal
5	Debit	2239.7691993294	1	Low Value	-2239.7691993294	-0.535059581340949	Common	Normal
1	Debit	32.5183182091387	0.009	Low Value	-32.5183182091387	-1.13202036591552	Common	Normal
10	Debit	1133.62750890488	1.3	Low Value	-1133.62750890488	-0.834220504557493	Common	Normal
22	Credit	5098.61311330898	1	Low Value	5098.61311330898	0.238127427196708	Common	Normal
18	Debit	30.458513325782	0.009	Low Value	-30.458513325782	-1.13257744927644	Common	Normal
10	Debit	4459.13605751757	1	Low Value	-4459.13605751757	0.065178022936453	Common	Normal
3	Debit	620.608874721773	1	Low Value	-620.608874721773	-0.972968668159034	Common	Normal
15	Credit	12909.4926264551	1.3	High Value	12909.4926264551	2.35061441373202	Common	Unusual
4	Debit	235.820340229885	1	Low Value	-235.820340229885	-1.07703643291126	Common	Normal

Calculated Measures Using DAX

Calculating Average Transaction Amount by dividing Total transaction Value from Total transaction Volume.

```
Average Transaction Amount = DIVIDE([TotalTransactionValue], [TotalTransactionVolume])

```

Calculating Mean Interest rate using AVERAGE.

```
MeanInterestRate = AVERAGE('Account Info'[InterestRate])
```

Calculating Total transaction volume by counting total rows using COUNTROWS

```
TotalTransactionVolume = COUNTROWS('Transaction Info')

```

Calculates the total number of transactions for each account by counting the TransactionID, while ignoring any filters on other columns but retaining the filter on AccountNumber.

```
TransactionFrequency =
CALCULATE(
    COUNT('Transaction Info'[TransactionID]),
    ALLEXCEPT('Transaction Info', 'Transaction Info'[AccountNumber])
)

```

Calculating Mean Account Balance using AVERAGE

```
MeanAccountBalance = AVERAGE('Account Info'[Balance])
```

Calculating the performance rating of branches by applying a weight of 70% to the transaction volume and 30% to the transaction value, and then returning the sum of both scores.

```
BranchRating =
VAR VolumeScore = [TotalTransactionVolume] * 0.7  // Weight for transaction volume
VAR ValueScore = [TotalTransactionValue] * 0.3   // Weight for transaction value

RETURN
VolumeScore + ValueScore

```

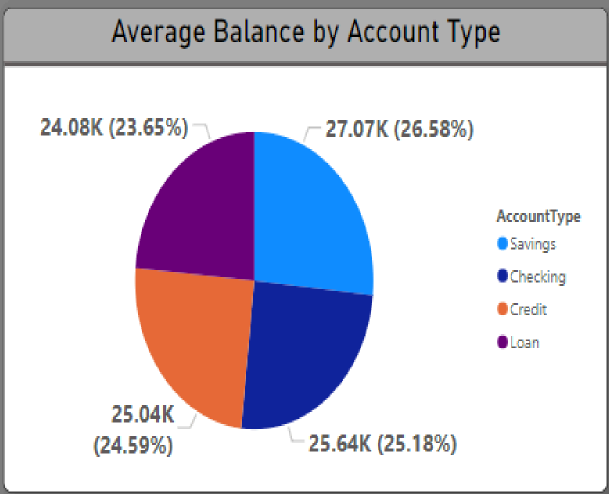
Problems Analysis

1. **Analysis of Account Balances**

    - Calculate the average account balance for each account type. Which account type has the highest average balance?

Problem Statement:

The objective of this analysis is to calculate the average account balance for each account type. By determining the account type with the highest average balance, we aim to provide insights into customer preferences and financial trends across different account types.

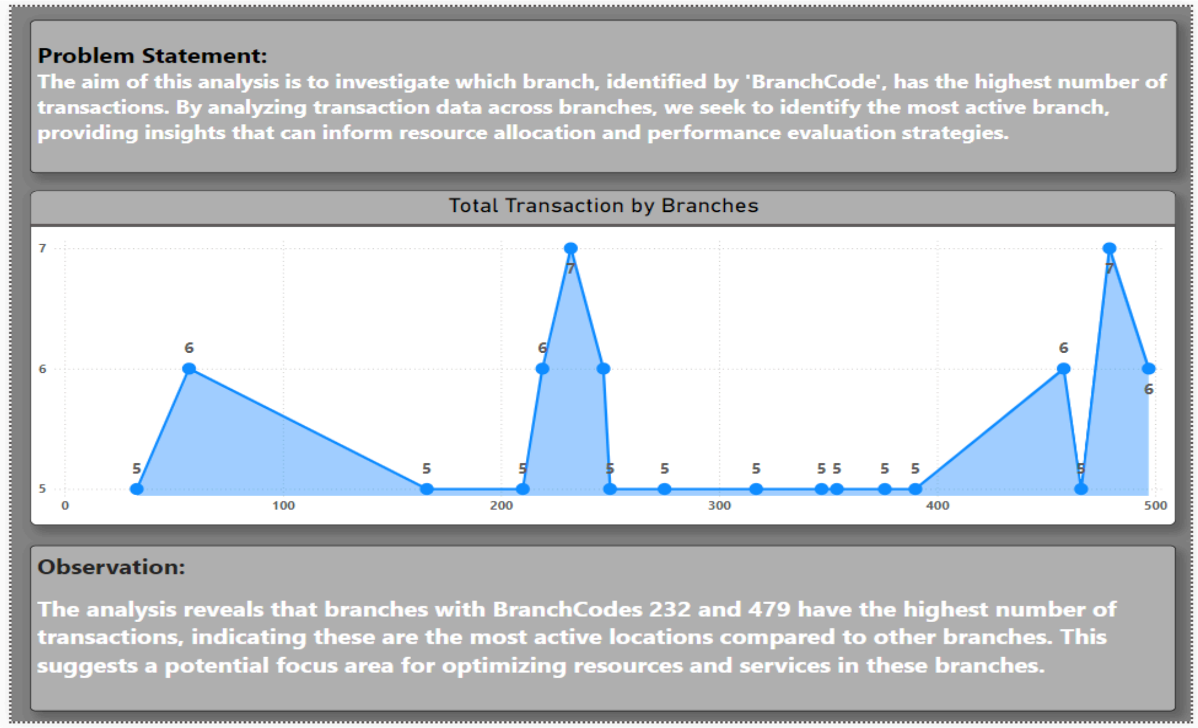


Observation:

The analysis shows that savings accounts have the highest average balance, followed by checking accounts with the second highest average balance. This indicates that customers tend to hold larger balances in their savings accounts compared to other account types.

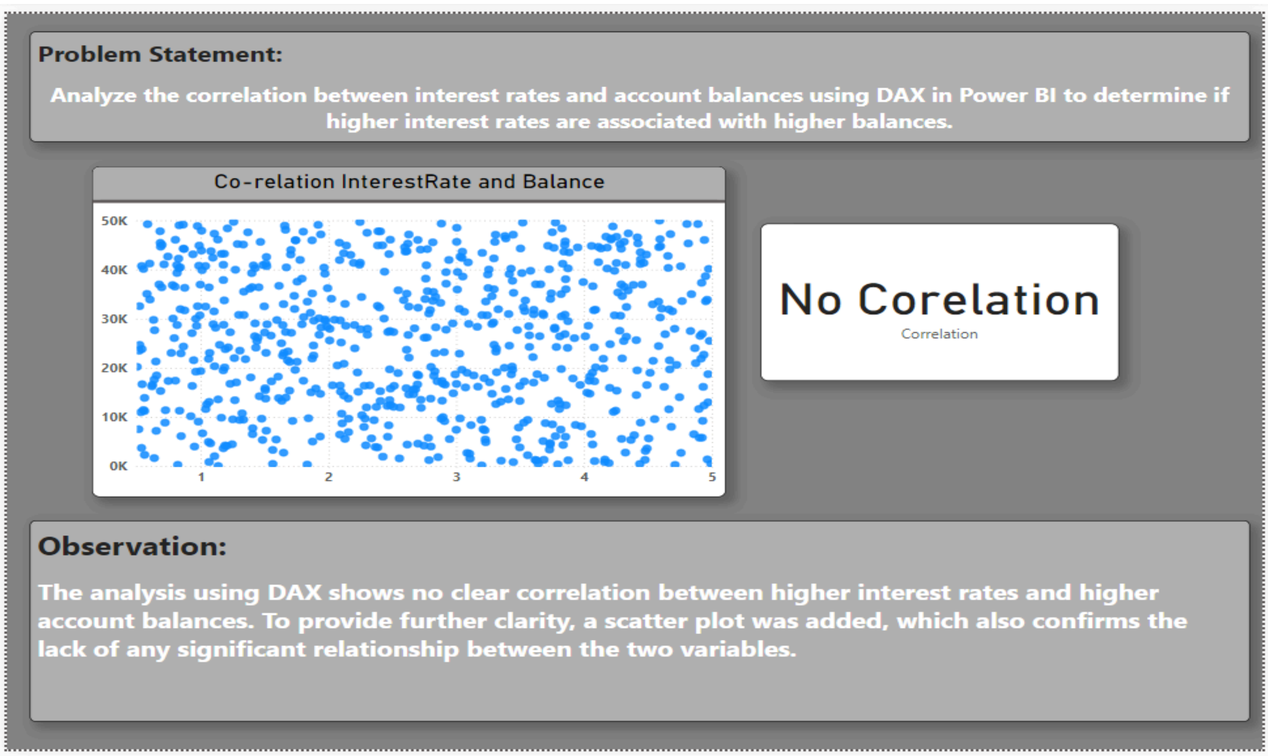
2. **Branch Activity Analysis**

   ○ Investigate which branch (identified by 'BranchCode') has the highest number of transactions.



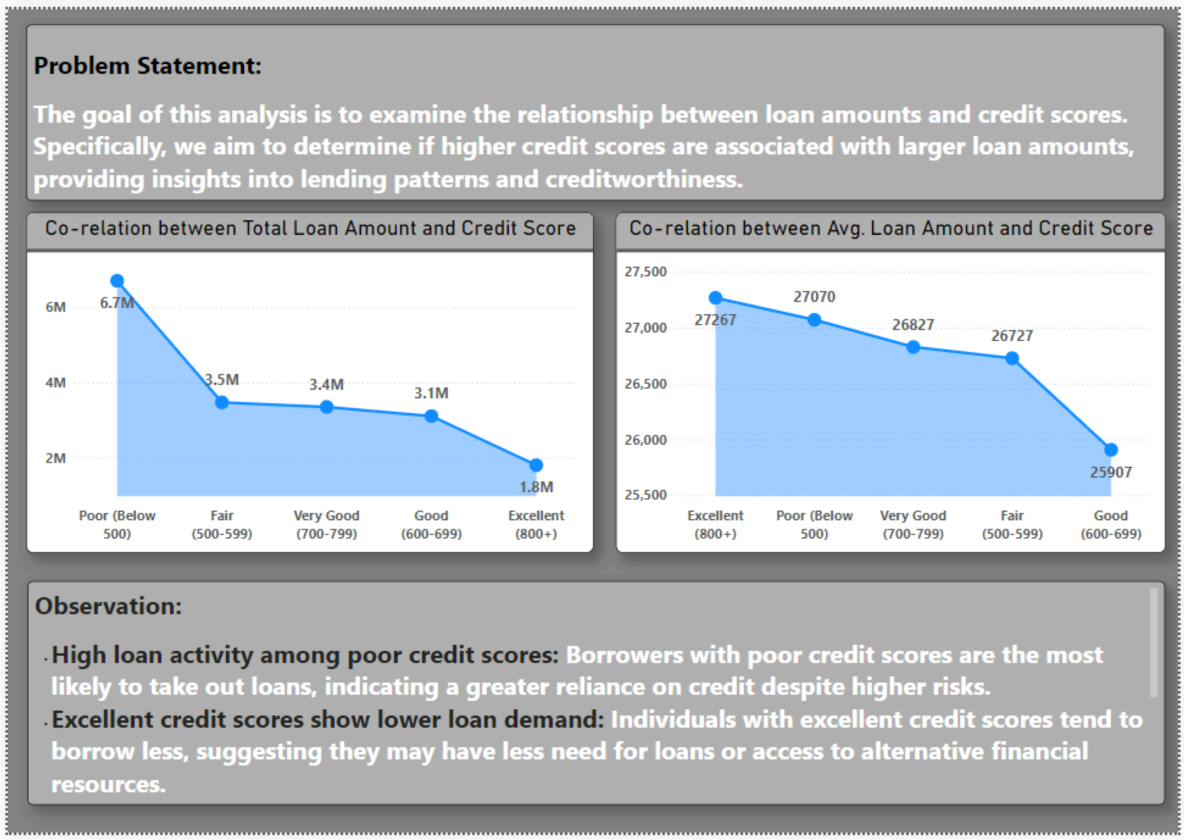
3. **Interest Rate and Balance Correlation**

   ○ Using DAX, analyse the correlation between interest rates and account balances. Does a higher interest rate correlate with higher balances.



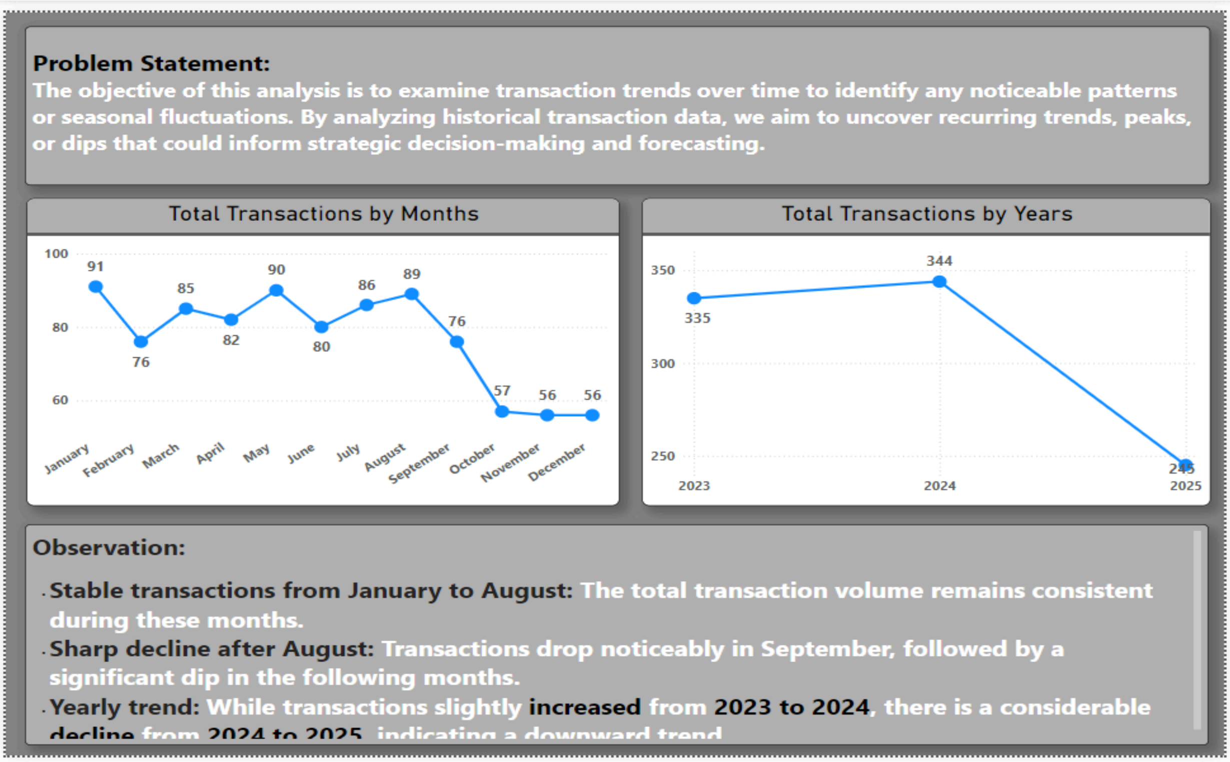
4. **Loan Amount and Credit Score Relation**

- Examine the relationship between loan amount and credit score. Do higher credit scores correlate with larger loan amounts?



5. **Transaction Trends Over Time**

- Analyse transaction trends over time. Are there any noticeable patterns or seasonal fluctuations?



6. **Customer Loyalty Analysis**

- Calculate the duration of each account's relationship with the bank (from 'OpeningDate' to the most recent transaction date). Who are the longest-standing customers?

The DAX formula for Customer_Loyalty **calculates the number of days between a customer's account opening date and their most recent transaction date,** which can help measure customer loyalty.

- DATEDIFF: Computes the difference between two dates in days.
- MIN('Account Info'[OpeningDate]): Retrieves the earliest account opening date.
- MAX('Transaction Info'[TransactionDate]): Retrieves the most recent transaction date.

```
Customer_Loyalty =
DATEDIFF(
    MIN('Account Info'[OpeningDate]),
    MAX('Transaction Info'[TransactionDate]),
    DAY
)
```

Problem Statement:

The aim of this analysis is to calculate the duration of each customer's relationship with the bank, from their account opening date to the most recent transaction. The objective is to identify the longest-standing customers, providing insights into customer loyalty and retention patterns.

AccountNumber	AccountHolder	Customer_Loyalty
100268	Patricia Martinez	1649
100876	Elizabeth Anderson	1649
104716	Karen Taylor	1649
104796	Charles Hernandez	1649
105468	Charles Brown	1649
107687	Jessica Taylor	1649
109038	Michael Jones	1649
109474	Richard Rodriguez	1649
109915	Sarah Miller	1649
118749	Barbara Lopez	1649
119118	Charles Williams	1649
119374	Mary Williams	1649
123228	James Smith	1649
124845	Elizabeth Moore	1649

Observation:

The analysis shows that the majority of customers have been with the bank for over 5 years, indicating a high level of customer loyalty and long-term relationships with the institution.

7. **High-Value Transaction Analysis**

- Identify high-value transactions and analyze their characteristics. What constitutes a high-value transaction in your analysis?

This DAX Formula created a calculated column that will display high, low value transactions based of Transaction Amount.

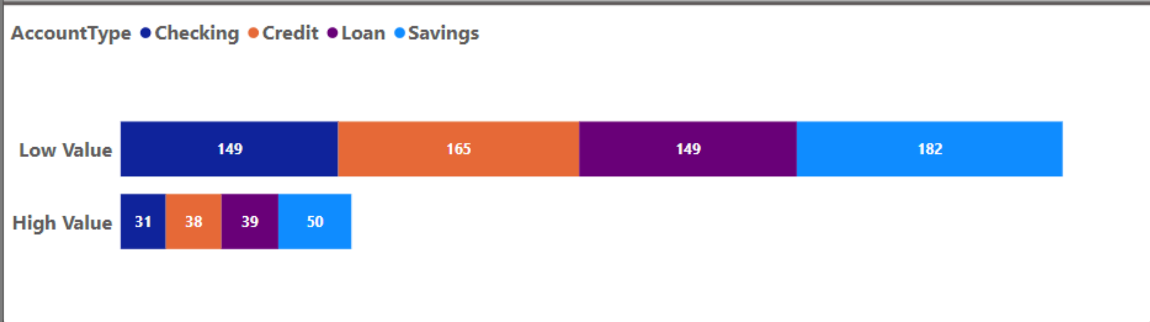
- If the **TransactionAmount** is greater than **8000**, the column will display **"High Value"**.
- Otherwise, it will display **"Low Value"**.

```
1 TransactionValue = IF('Transaction Info'[AmountInUSD]>=8000,"High Value","Low Value")
```

Hours)	TransactionCategory	AmountInUSD	RateToUSD	TransactionValue	FinalSummativeAmoun
11	Debit	4779.69797116695	1.3	Low Value	-4779.6979711
23	Credit	12223.3531347035	1.3	High Value	12223.353134
4	Credit	30.9063989280933	0.009	Low Value	30.906398928
23	Debit	8367.14781395114	1	High Value	-8367.1478139
6	Debit	7043.06762499775	1	Low Value	-7043.0676249
22	Debit	3177.57985823408	1.1	Low Value	-3177.5798582
5	Debit	9986.8297353486	1	High Value	-9986.829735
2	Credit	83.6606120108684	0.009	Low Value	83.660612010
12	Debit	7647.59072654907	1	Low Value	-7647.5907265
18	Credit	5105.34037878721	1	Low Value	5105.3403787
5	Debit	2239.7691993294	1	Low Value	-2239.769199
1	Debit	32.5183182091387	0.009	Low Value	-32.51 -2239.7
10	Debit	1133.62750890488	1.3	Low Value	-1133.6275089
22	Credit	5098.61311330898	1	Low Value	5098.6131133
18	Debit	30.458513325782	0.009	Low Value	-30.45851332
10	Debit	4459.13605751757	1	Low Value	-4459.1360575

Problem Statement:
The purpose of this analysis is to identify and examine high-value transactions, focusing on their defining characteristics. The goal is to determine what constitutes a high-value transaction and analyze factors such as transaction frequency, customer profiles, and transaction types associated with these larger amounts.

TotalTransactions by TransactionValue and TransactionType

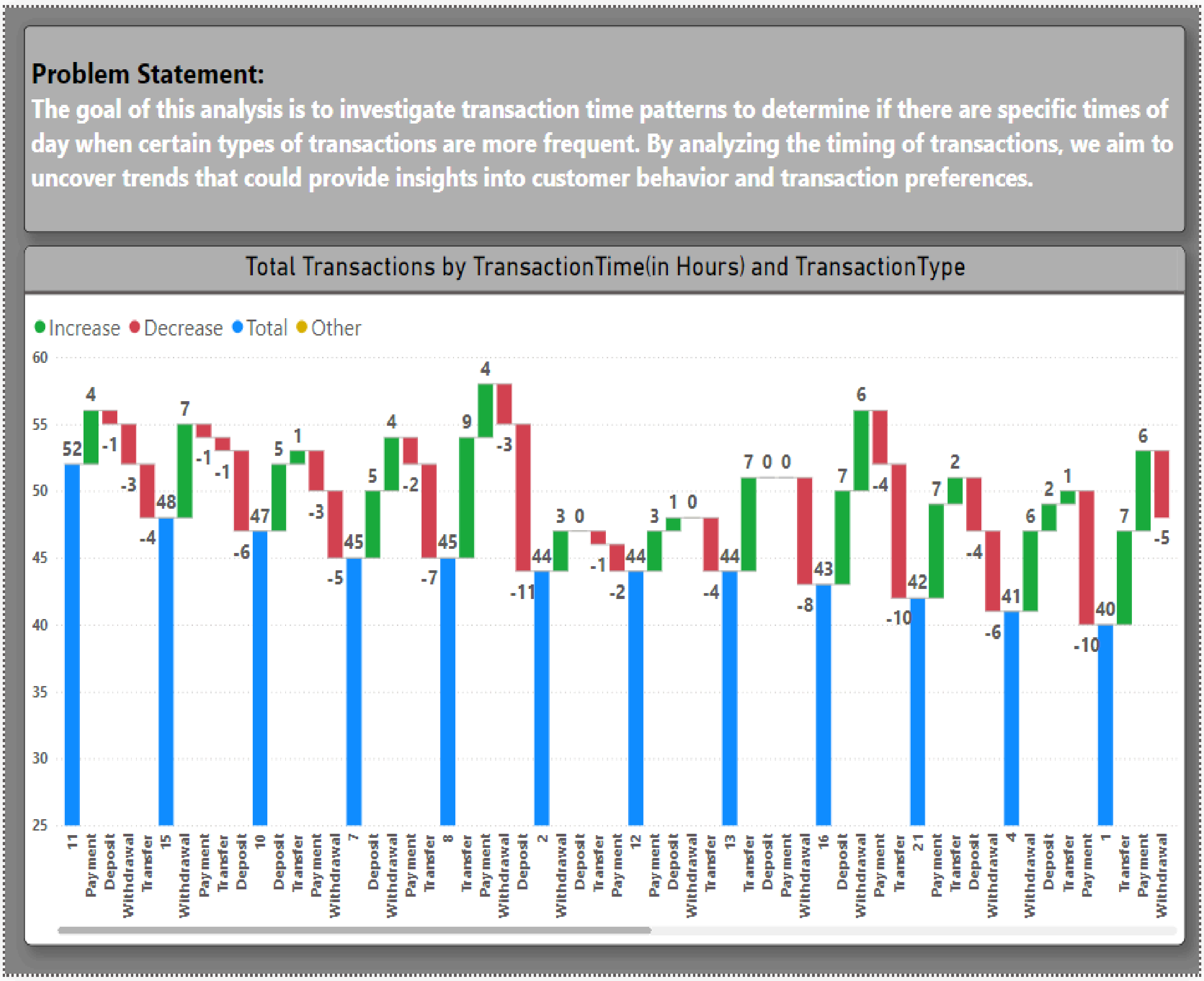


Observation:

The analysis reveals that most high-value transactions occurred in savings accounts, followed by loan and credit accounts. The lowest number of high-value transactions was observed in checking accounts, indicating a preference for larger transactions in savings and credit-related activities.

8. **Analysis of Transaction Time Patterns**

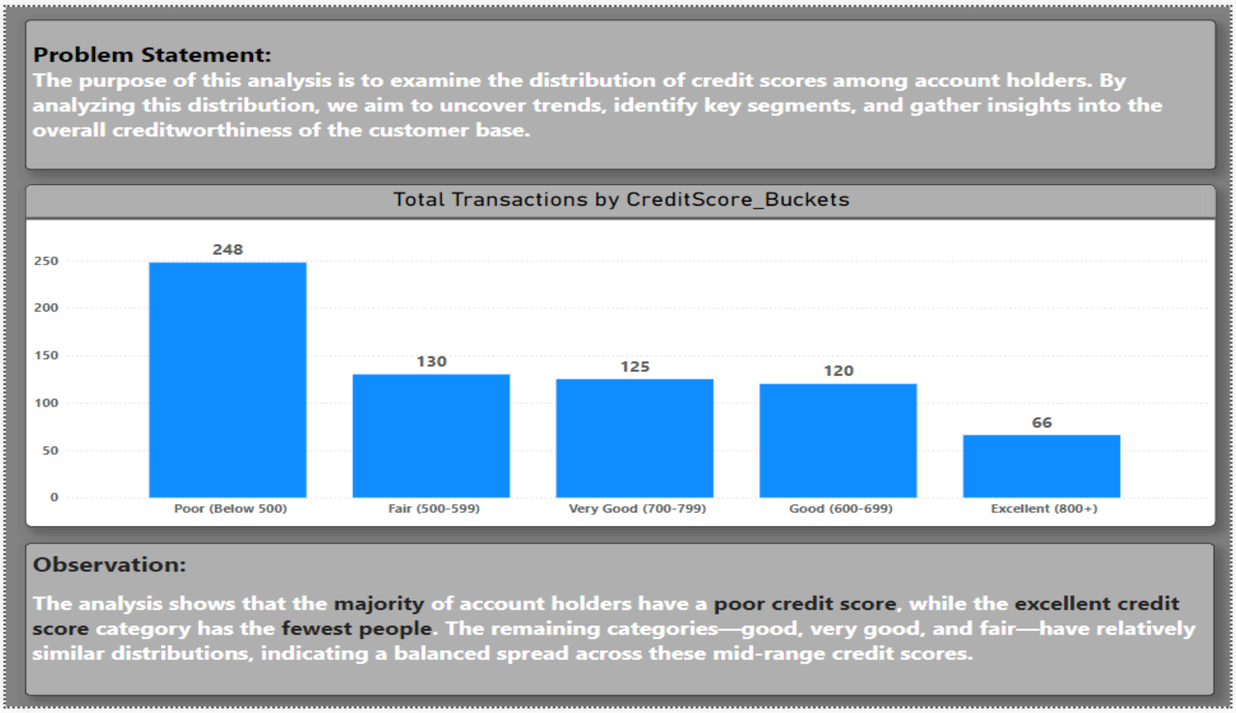
- Investigate if there are patterns in the times of day when different types of transactions are made.



Observation: The waterfall chart highlights that the **highest number of transactions occurred during the 11th hour, followed by the 15th, 10th and 7th.** In contrast, the **lowest number of transactions was recorded during the 9th hour.**

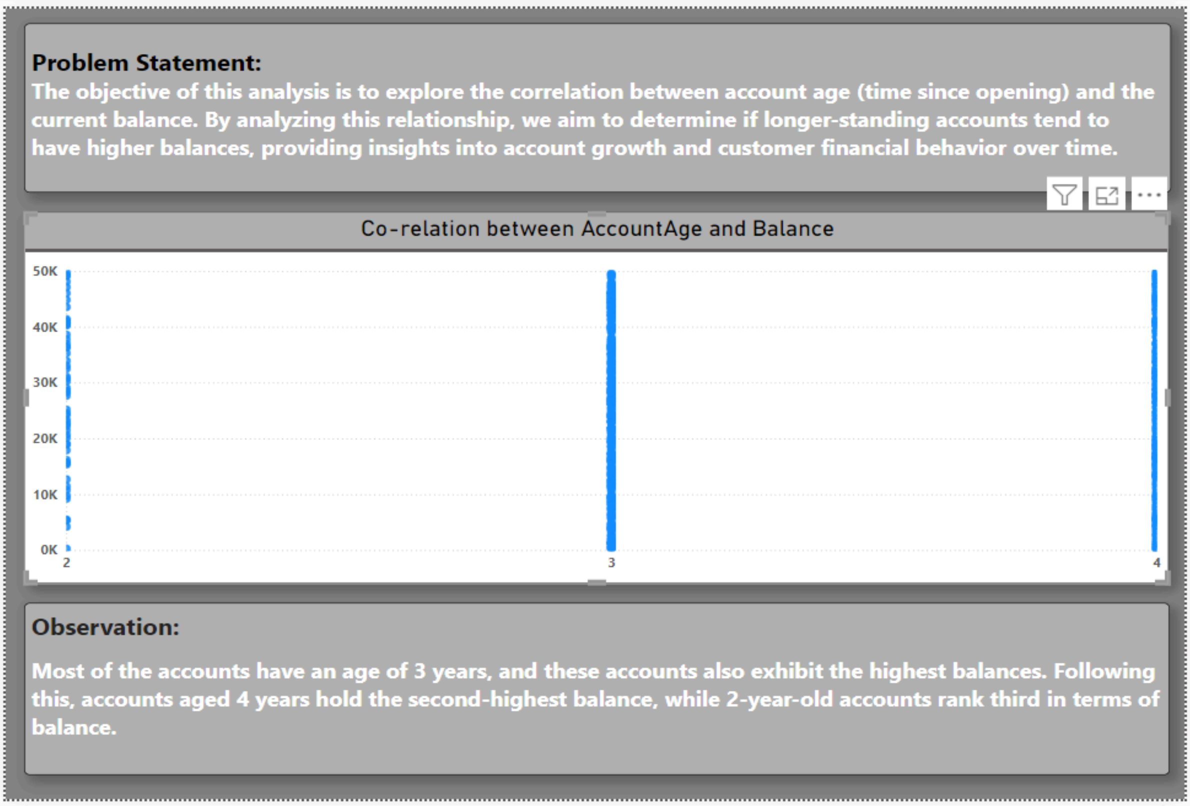
9. **Credit Score Distribution**

- Analyse the distribution of credit scores among account holders. What insights can you gather?



10. **Correlation Between Account Age and Balance**

- Explore if there's a correlation between the age of an account (time since opening) and its current balance.



11. **Risk Assessment**

- Using DAX, develop a risk assessment model based on transaction patterns, account balances, and credit scores.

This **DAX formula** creates a new column, **RiskAssessment**, to classify account holders based on their **Credit Score** and **Balance**. It uses the **IF** function for conditional logic:

- If the **Credit Score** is **less than 600** and the **Balance** is **less than 5000**, the account is labeled as **"High Risk"**.
- If the **Credit Score** is between **600 and 750** and the **Balance** is between **5000 and 10000**, it's labeled as **"Medium Risk"**.
- Otherwise, the account is considered **"Low Risk"**.

```
RiskAssessment =
IF (
    'Account Info'[CreditScore] < 600 && 'Account Info'[Balance] < 5000,
    "High Risk",
    IF (
        'Account Info'[CreditScore] >= 600 && 'Account Info'[CreditScore] < 750 && 'Account Info'[Balance] < 10000 && 'Account Info'
        [Balance] >= 5000,
        "Medium Risk",
        "Low Risk"
    )
)
```

InterestRate	CreditScore	OpeningDate	LoanAmount	Sector	Residence(in Years)	City	CreditScore_Buckets	AccountAge	RiskAssessment
3.07217258857027	376	04 January 2020	25116.0517732897	Finance	16	Tokyo	Poor (Below 500)	4	Low Risk
0.742865871152115	813	10 January 2020	35650.4194290051	Finance	16	Tokyo	Excellent (800+)	4	Low Risk
1.01380125613803	546	11 January 2020	38643.6765591292	Education	2	Tokyo	Fair (500-599)	4	Low Risk
2.77314974571401	358	14 January 2020	6436.5874220971	Education	15	Tokyo	Poor (Below 500)	4	Low Risk
0.769429840128228	530	17 January 2020	32051.7929532114	Technology	11	Tokyo	Fair (500-599)	4	Low Risk
1.59039267327905	583	21 January 2020	33283.8161056025	Finance	11	Tokyo	Fair (500-599)	4	Low Risk
4.92056281362059	630	29 January 2020	20356.17549262	Technology	2	Tokyo	Good (600-699)	4	Low Risk
4.81596553687967	398	02 February 2020	39330.5910418594	Retail	5	Tokyo	Poor (Below 500)	4	Low Risk
3.67988835276771	567	08 February 2020	19391.1084796354	Technology	10	Tokyo	Fair (500-599)	4	Low Risk
1.16142971576234	301	09 February 2020	5776.24931779269	Technology	3	Tokyo	Poor (Below 500)	4	Low Risk
2.4057731011674	497	19 February 2020	49543.3164299473	Education	2	Tokyo	Poor (Below 500)	4	Low Risk
0.783201935809014	330	20 February 2020	48765.8734742784	Finance	7	Tokyo	Poor (Below 500)	4	Low Risk
2.23283072732895	665	26 February 2020	37738.6671000613	Finance	17	Tokyo	Good (600-699)	4	Low Risk

Problem Statement:
The goal of this analysis is to develop a risk assessment model using DAX that evaluates customer risk levels based on transaction patterns, account balances, and credit scores. By leveraging these factors, the model aims to provide insights into potential financial risks associated with customers, helping the bank make informed decisions regarding credit approvals, loan allocations, and account management. This model will

Risk Assesment

Observation:

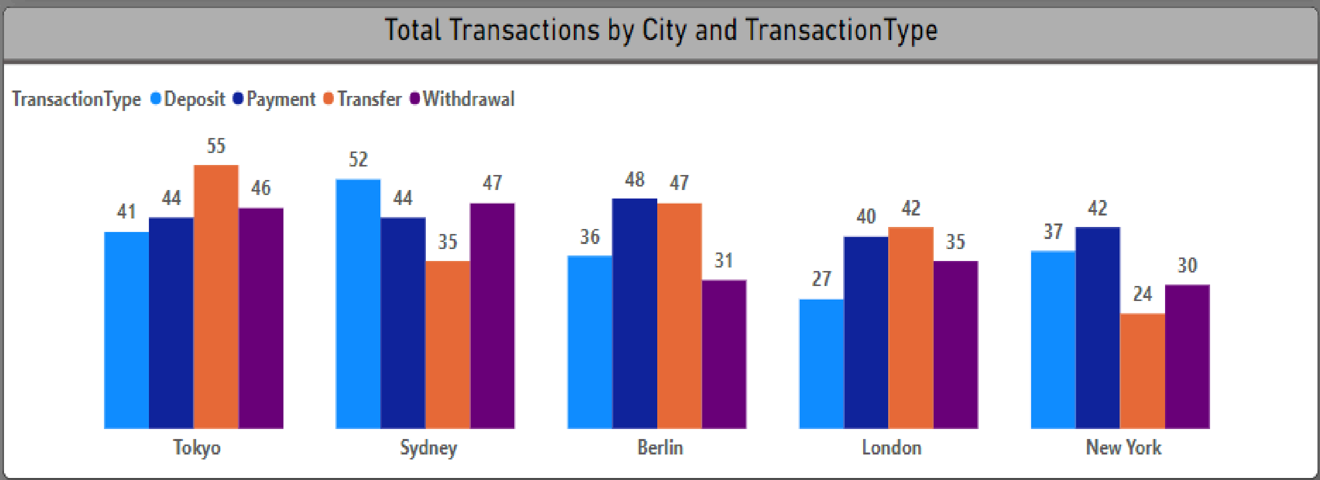
The risk assessment model reveals that 92% of the customers fall under the low-risk category, indicating stable transaction patterns, healthy account balances, and strong credit scores. 6% of the customers are classified as moderate risk, which suggests occasional irregularities in transactions or slightly lower credit scores. The remaining 2% are identified as high-risk customers, showing significant fluctuations in their transaction behavior, low account balances, or poor credit scores. This distribution highlights that the majority of the customer base is financially stable, while a small segment poses a potential risk.

12. **Customer Demographics and Transaction Behaviour**

- Analyse transaction behaviour based on customer demographics inferred from account data.

Problem Statement:

The objective of this analysis is to examine transaction behavior in relation to customer demographics inferred from account data. By analyzing factors such as age, gender, and location, we aim to uncover patterns and trends in transaction activities, providing insights into how demographic characteristics influence financial behavior.



Observation:

- **Sydney** has the **highest number** of total transactions across all transaction types, with the highest being **55 payments**, followed by **52 deposits**.
- **Berlin** shows a more **balanced distribution**, with **48 deposits** and **47 payments** leading in transaction volume.
- **Tokyo**'s transactions are **dominated by payments (44) and deposits (41)**, indicating a preference for these transaction types.

13. **Branch and Account Type Influence on Transactions**

- Investigate if certain branches or account types have a significant influence on the types and values of transactions.

Problem Statement:

The objective of this analysis is to investigate whether certain branches or account types have a significant influence on the types and values of transactions. By analyzing transactional data across various branches and account categories, this study aims to identify patterns that may indicate which branches or account types are driving higher transaction volumes and values. Understanding these influences can help in

BranchCode	AccountType	TotalTransactionValue
219	Checking	22602
289	Savings	22358
33	Savings	22043
442	Credit	20286
332	Credit	19611
12	Credit	18742
109	Savings	17844
71		17664
396	Loan	17440
23	Loan	17272
38	Loan	17096
136	Loan	17032
225	Loan	16948
Total		**3897563**

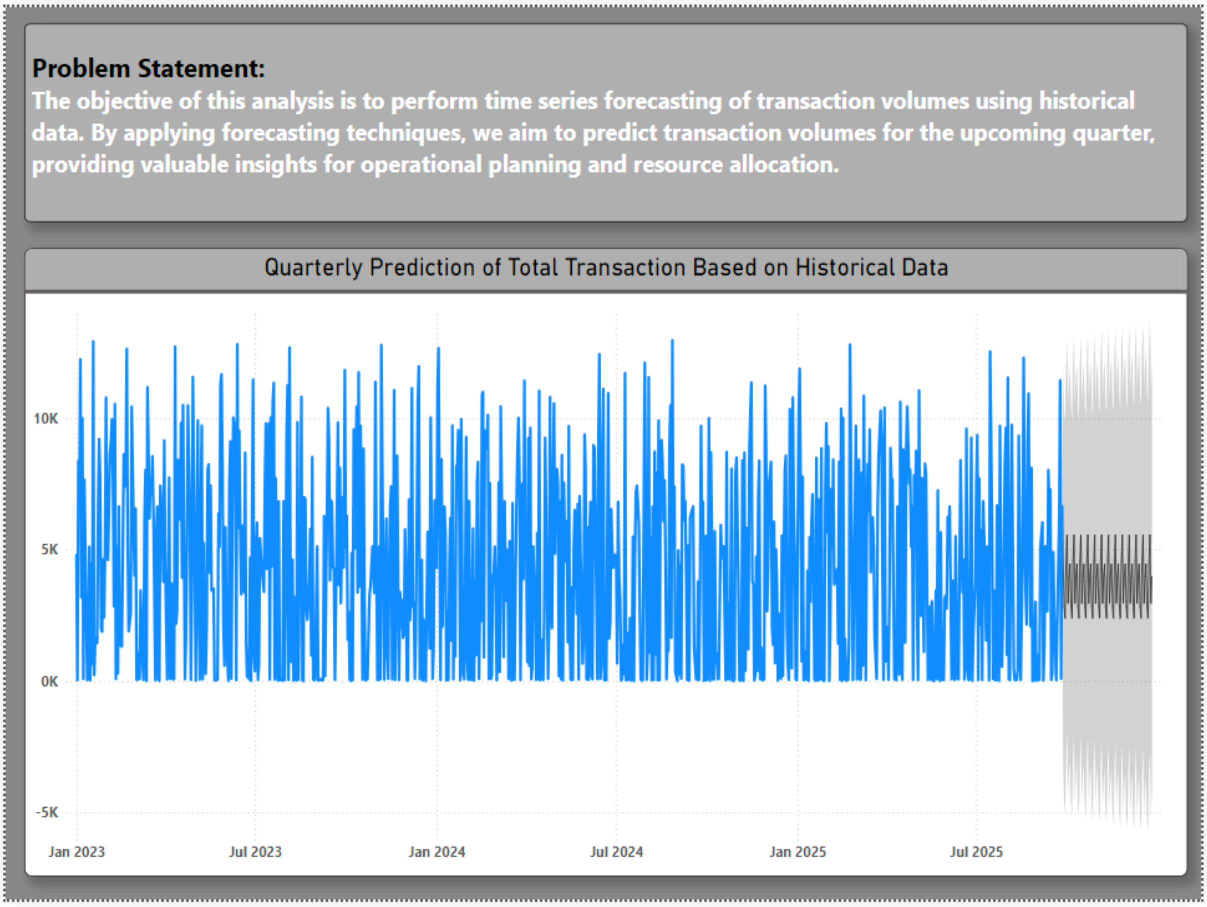
Observation:

From the analysis, it is observed that the **Checking and Savings** account types consistently hold the highest total transaction values across various branches. For instance, **Branch 219** shows the highest transaction value for Checking accounts at 22,602, while Branch 289 and 33 have substantial transaction values for Savings accounts at 22,358 and 22,043, respectively. Credit accounts, such as those in Branch 442, also maintain significant transaction values, indicating that these account types contribute heavily to the overall financial activity within the bank. Loans and other account types have comparatively lower transaction values, suggesting a lesser contribution to high-value transactions.

14. **Data Modeling: Time Series Forecasting of Transactions**

- Perform time series forecasting of transaction volumes using historical data. What are the predicted transaction volumes for the next quarter?

For forecasting, I utilized the built-in **Forecast** feature available under the **Analytics** options (Magnifying glass) in the Visualizations pane, allowing for seamless integration of predictive insights into the existing data visualization.



15. **Data Transformation: Identifying Unusual Transactions**

- Using Power BI's data transformation capabilities, identify any unusual transactions (e.g., unusually high amounts, rare transaction types).

This **DAX formula** creates a calculated column named **RareTransaction** to classify transactions as either **"Rare"** or **"Common"** based on how frequently a specific **TransactionType** occurs:

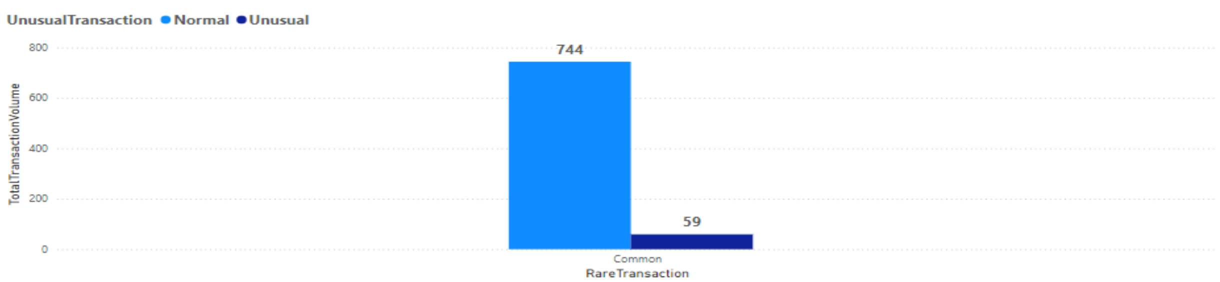
- **COUNTROWS** is used to count the number of rows where the **TransactionType** in the **'Transaction Info'** table matches the current row's **TransactionType** (using the **EARLIER** function to reference the current row).
- If the count is **less than 3**, the transaction is labeled as **"Rare"**.
- Otherwise, the transaction is considered **"Common"**.

```
1 RareTransaction = IF(COUNTROWS(FILTER('Transaction Info', [TransactionType] = EARLIER([TransactionType]))) < 3, "Rare", "Common")
2
```

	TransactionCategory	AmountInUSD	RateToUSD	TransactionValue	FinalSummativeAmount	ZScore	RareTransaction	Unus
11	Debit	4779.69797116695	1.3	Low Value	-4779.69797116695	0.151875413393371	Common	Norm
23	Credit	12223.3531347035	1.3	High Value	12223.3531347035	2.16504494683221	Common	Unusu
4	Credit	30.9063989280933	0.009	Low Value	30.9063989280933	-1.13245631663001	Common	Norm
23	Debit	8367.14781395114	1	High Value	-8367.14781395114	1.12211712496693	Common	Norm
6	Debit	7043.06762499775	1	Low Value	-7043.06762499775	0.764013768797425	Common	Norm
22	Debit	3177.57985823408	1.1	Low Value	-3177.57985823408	-0.281424531736623	Common	Norm
5	Debit	9986.8297353486	1	High Value	-9986.8297353486	1.56016727994883	Common	Norm
2	Credit	83.6606120108684	0.009	Low Value	83.6606120108684	-1.11818870586547	Common	Norm
12	Debit	7647.59072654907	1	Low Value	-7647.59072654907	0.927509721197895	Common	Norm
18	Credit	5105.34037878721	1	Low Value	5105.34037878721	0.239946845964316	Common	Norm
5	Debit	2239.7691993294	1	Low Value	-2239.7691993294	-0.535059581340949	Common	Norm
1	Debit	32.5183182091387	0.009	Low Value	-32.5183182091387	-1.13202036591552	Common	Norm
10	Debit	1133.62750890488	1.3	Low Value	-1133.62750890488	-0.834220504557493	Common	Norm
22	Credit	5098.61311330898	1	Low Value	5098.61311330898	0.238127427196708	Common	Norm
18	Debit	30.458513325782	0.009	Low Value	-30.458513325782	-1.13257744927644	Common	Norm
10	Debit	4459.13605751757	1	Low Value	-4459.13605751757	0.065178022936453	Common	Norm
3	Debit	620.608874721773	1	Low Value	-620.608874721773	-0.972968668159034	Common	Norm
15	Credit	12909.4926264551	1.3	High Value	12909.4926264551	2.35061441373202	Common	Unusu

Problem Statement:
This project aims to leverage Power BI's data transformation capabilities to identify unusual transactions within our financial data, specifically focusing on unusually high amounts and rare transaction types. By analyzing these anomalies, we can enhance risk management and improve fraud detection, ensuring the integrity of our financial operations.

Unusal Transactions



Observation:

The analysis of our transaction data reveals that the majority of transactions are common and follow expected patterns. However, there are a few unusual transactions characterized by significantly high amounts and rare transaction types. This project aims to leverage Power BI's data transformation capabilities to identify these anomalies, enabling improved risk management and fraud detection.