

Financial Analysis

By applying DAX functions



INTRODUCTION

Objective:

- To calculate and analyze financial metrics such as running totals, moving averages, and growth rates for credit card usage.
- To assess key performance indicators (KPIs) related to customer behavior, credit utilization, and delinquency risk.
- To generate actionable insights for improving customer retention and enhancing financial performance.

Scope:

- Analyzing credit card transaction data and financial metrics.
- Creating dynamic reports and dashboards in Power BI.
- Calculating KPIs related to customer behavior, utilization rates, and delinquency risk.
- Providing insights for strategic decision-making to improve financial outcomes.

Tools:

- Power BI: For data visualization and dashboard creation.
- DAX (Data Analysis Expressions): For advanced calculations and metrics.
- Excel/SQL: For data preprocessing and initial analysis if needed.



Write DAX formulas for the following:



1. Running total of credit card transactions.
2. Calculate the 4-week moving average of the creditLimit for each client.
3. Calculate the MOM% growth and WOW% growth on transaction amount.
4. Calculate customer acquisition cost (CAC) as a ratio of transaction amount.
5. Calculate the yearly average of avg_utilization_ratio for all clients.
6. Calculate the percentage of Interest_Earned compared to Total_Revolving_Bal for each client.
7. Calculate Top 5 Clients by Total Transaction Amount.
8. Identify clients whose Avg_Utilization_Ratio exceeds 80%.
9. Customer Churn Indicator: Create a KPI that flags clients who have not made any transactions (Total_Trans_Amt = 0) in the last 6 months.

Write DAX formulas for the following:



10. Delinquency Rate: Calculate the percentage of clients with `Delinquent_Acc > 0`.
11. Credit Risk Score: Create a score for each client based on their `Avg_Utilization_Ratio`, `delinquent_Acc`, and `Total_Revolving_Bal`.
12. Income vs Credit Limit Correlation: Show the correlation between `Income` and `credit_Limit` for all clients.
13. Average Customer Satisfaction Score by Credit Card Category: Calculate the average `Cust_Satisfaction_Score` by `Card_Category`.
14. Loan Approval vs Credit Limit: Analyze how `Credit_Limit` affects `Personal_loan` approval by calculating the average credit limit for clients with and without loans.
15. High Risk Clients Flag: Create a flag for clients whose `Total_Revolving_Bal` exceeds 90% of their `Credit_Limit` and who have a high `Avg_Utilization_Ratio`.

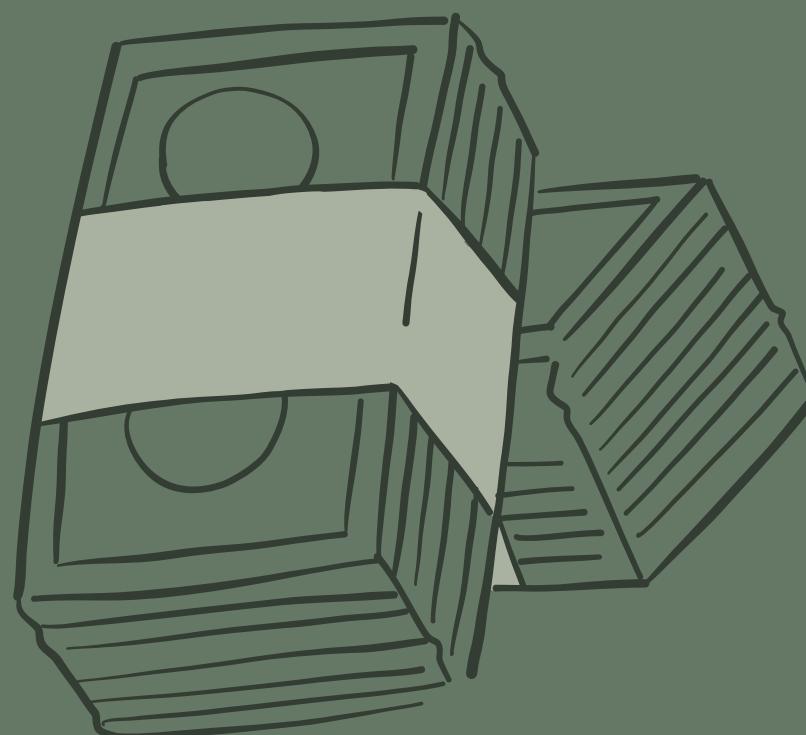


Running total of credit card transactions.

```
Running Total =  
CALCULATE([Total Transaction Amount], FILTER(ALL  
('Credit Card'), 'Credit Card'[Week_Start_Date]  
<= MAX('Credit Card'[Week_Start_Date])))
```

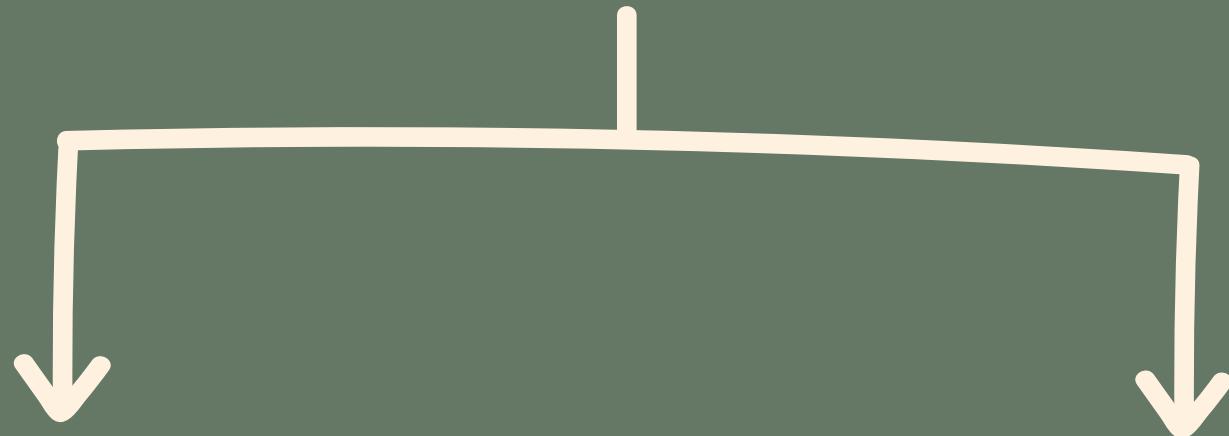


Calculate the 4-week moving average of the CreditLimit for each client.



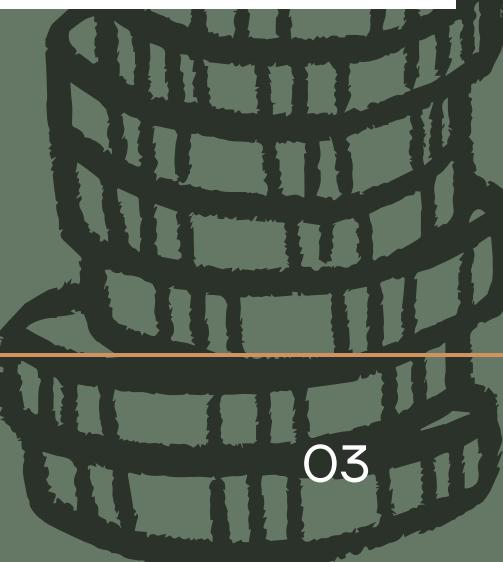
```
Moving_average_4_weeks =  
  
VAR weeks4 = DATESINPERIOD('calander'[Date],MAX('calander'[Date]),-28,  
DAY)  
  
VAR total_amount = CALCULATE([Total Transaction Amount],weeks4)  
  
VAR num_of_weeks = CALCULATE(DISTINCTCOUNT('calander'[week_num]),weeks4)  
  
RETURN DIVIDE(total_amount,num_of_weeks,0)
```

Calculate the MOM% growth and WOW% groth on transaction amount.



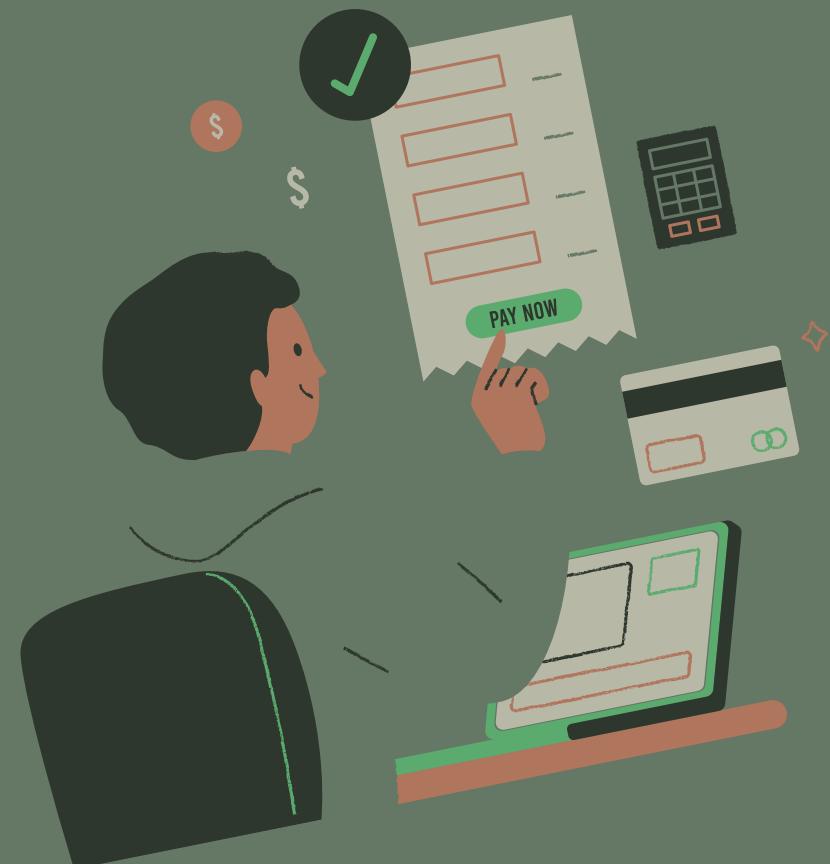
```
MOM%growth =  
VAR perv_month = CALCULATE([Total Transaction Amount],  
DATEADD('calander'[Date],-1,month))  
  
RETURN DIVIDE([Total Transaction Amount]-perv_month,  
perv_month,0)
```

```
WOW%growth =  
VAR prev_week = CALCULATE([Total Transaction  
Amount],DATEADD('calander'[Date],-7,DAY))  
  
RETURN DIVIDE([Total Transaction Amount]-prev_week,  
prev_week,0)
```



Calculate customer acquisition cost (CAC) as a ratio of transaction amount.

```
RATIO_CAC_TRANSACTION_AMOUNT =  
DIVIDE(SUM('Credit Card'[Customer_Acq_Cost]),  
[Total Transaction Amount],0)
```



Calculate the yearly average of avg_utilization_ratio for all clients.

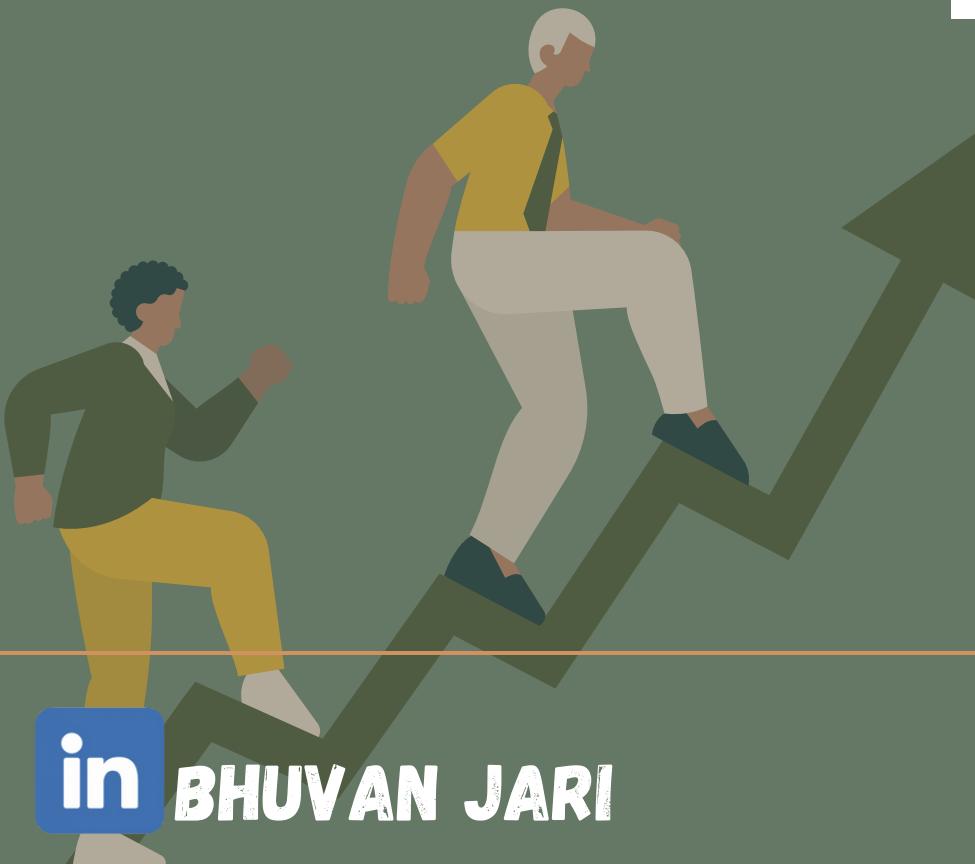
```
AVG_UTILIZATION_RATIO =
```

```
AVERAGE('Credit Card'[Avg_Utilization_Ratio])
```



Calculate the percentage of Interest_Earned compared to Total_Revolving_Bal for each client.

```
INTEREST_EARNED_BY_REVOL_BALANCE =  
DIVIDE(SUM('Credit Card'[Interest_Earned]),  
SUM('Credit Card'[Total_Revolving_Bal]))
```



Calculate Top 5 Clients by Total Transaction Amount.



```
TOP_5_CLIENTS =  
TOPN(5,SUMMARIZE('Credit Card','Credit Card'  
[Client_Num],"tottal amount",[Total Transsaction  
Amount]),[tottal amount],DESC)
```

Identify clients whose Avg_Utilization_Ratio exceeds 80%.

```
CHECK_EXCEEDS_80 =  
IF([AVG_UTILIZATION_RATIO]> 0.80,True,False)
```



Customer Churn Indicator:

**Create a KPI that flags clients who have not made any transactions
(Total_Trans_Amt = 0) in the last 6 months.**

```
CHURN =  
  
VAR balance = CALCULATE([Total Transaction Amount],  
DATESINPERIOD('calander'[Date],MAX('calander'[Date]),  
-6,MONTH))  
  
RETURN IF(ISBLANK(balance),"Churned","Not Churned")
```



Delinquency Rate:

Calculate the percentage of clients with Delinquent_Acc > 0.

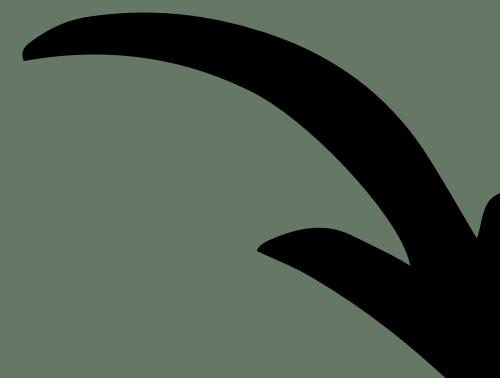
```
DELINQUENCY_RATE =  
  
VAR GREATER_ZERO = CALCULATE(COUNTROWS('Credit Card'),  
'Credit Card'[Delinquent_Acc] >0)  
  
VAR total_rowws = COUNTROWS('Credit Card')  
RETURN DIVIDE(GREATER_ZERO,total_rowws,0)
```



Credit Risk Score:

Create a score for each client based on their
Avg_Utilization_Ratio, deLinquent_Acc, and Total_Revolving_Bal.

```
NORMALIZED_REVOLVING_BALANCE =  
  
DIVIDE('Credit Card'[Total_Revolving_Bal]-min('Credit Card'  
[Total_Revolving_Bal]),  
  
max('Credit Card'[Total_Revolving_Bal])-min('Credit Card'  
[Total_Revolving_Bal]),0)
```



```
CREDITT_RISK_SCORE =  
'Credit Card'[AVG_UTILIZATION_RATIO] * 0.5 +  
'Credit Card'[NORMALIZED_REVOLVING_BALANCE] * 0.3 +  
'Credit Card'[Delinquent_Acc] * 0.2
```

Income vs Credit Limit Correlation: Show the correlation between Income and credit_limit for all clients.

IN THIS ANALYSIS, WE USED QUICK MEASURES TO CALCULATE THE CORRELATION BETWEEN VARIOUS METRICS. SPECIFICALLY

CATEGORY: Client_Num

MEASURE X: INCOME

MEASURE Y: CREDIT_LIMIT

THIS APPROACH HELPS US ASSESS THE RELATIONSHIP BETWEEN A CLIENT'S INCOME AND THEIR CREDIT_LIMIT EFFECTIVELY.

The screenshot shows the 'Quick measure' gallery in Power BI. The 'Calculations' tab is selected. A 'Correlation coefficient' quick measure is chosen, which calculates the correlation coefficient between two values over a category. The 'Category' field is set to 'Client_Num'. The 'Measure X' field is set to 'Income' and the 'Measure Y' field is set to 'Credit_Limit'. An 'Add' button is visible at the bottom right.

Quick measure

Calculations

Suggestions with Copilot

Correlation coefficient

Calculate the correlation coefficient between two values over a category. Originally suggested by Daniil Maslyuk in the quick measures gallery. [Learn more](#)

Category

Client_Num

Measure X

Income

Measure Y

Credit_Limit

Add

» Data

Search

Σ Activation_30_Days

Σ Annual_Fees

Σ Avg_Utilization_Ratio

Card_Category

Client_Num

Σ Credit_Limit

credit_risk_score

credit_risk_value

Σ current_year

Σ Customer_Acq_Cost

Σ Delinquent_Acc

Exp Type

High_Utilization_Clients

Income and Credit_Limit corr...

Σ Interest_Earned

no_trans_in_last_6_months

Average Customer Satisfaction Score by Credit Card Category: Calculate the average Cust_Satisfaction_Score by Card_Category.

```
AVG_SATISFACTION_SCORE =  
SUMMARIZE('Credit Card', 'Credit Card'[Card_Category],  
"AVG_SATISFACTION_SCORE", AVERAGE(Customers  
[Cust_Satisfaction_Score]))
```



Loan Approval vs Credit Limit:

Analyze how Credit_Limit affects Personal_loan approval by calculating the average credit limit for clients with and without loans.

```
LOAN_APPROVAL_NO =  
CALCULATE(AVERAGE('Credit Card'[Credit_Limit]),  
Customers[Personal_loan] = "NO")
```

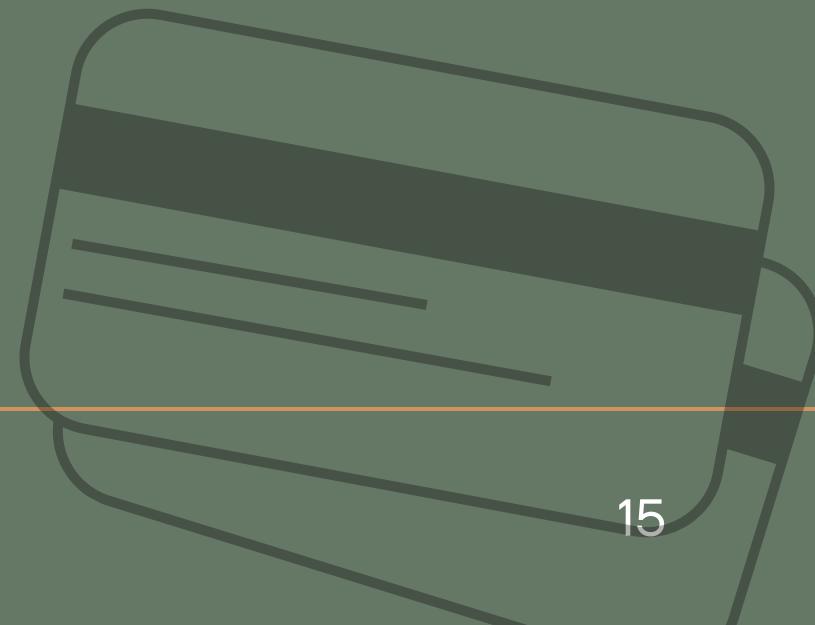


```
LOAN_APPROVAL_YES =  
CALCULATE(AVERAGE('Credit Card'[Credit_Limit]),  
Customers[Personal_loan] = "YES")
```

High Risk Clients Flag:

Create a flag for clients whose Total_Revolving_Bal exceeds 90% of their Credit_Limit and who have a high Avg_Utilization_Ratio.

```
FLAG_CLIENTS =  
IF('Credit Card'[NORMALIZED_REVOLVING_BALANCE] > 0.9 &&  
[Avg_Utilization_Ratio] > 0.8, "FLAGGED", "NOT FLAGGED")
```



**THANK YOU
FOR YOUR TIME AND ATTENTION**