🧩 Lesson 16: Advanced Charting Techniques in Power BI

✅ Prerequisites

1. Load Card_data.csv and sales.csv into Power BI Desktop.
2. Go to:
    - File → Options and Settings → Options → Current File → Data Load → Turn off Auto Date/Time.
3. Create a new DATE table using:

DAX

КопироватьРедактировать

```
DateTable = ADDCOLUMNS(
    CALENDARAUTO(),
    "Year", YEAR([Date]),
    "Month", FORMAT([Date], "MMMM"),
    "Month Number", MONTH([Date])
)
```

4. Mark it as Date Table and create relationships to:
    - account_opened_date in Card_data.csv
    - sales_date in sales.csv

🎯 Visual 1: Total Card Limit by Card Brand and Card Type

- Chart Type: Stacked Column Chart
- X-axis: card_brand (e.g., Visa, Mastercard)
- Y-axis: SUM(card_limit)
- Legend: card_type (Credit, Debit, Prepaid)
- Tooltip: DISTINCTCOUNT(client_id) – shows client diversity

🔍 Visual 2: Drill Down into Monthly Trends

- Chart Type: Stacked Column Chart
- Hierarchy: Year → Month from account_opened_date
- Values: COUNT(card_number)
- Enable drill up/down arrows for interactivity

💼 Visual 3: Top 10 Clients by Total Card Limit

- Chart Type: Bar Chart
- Axis: client_id
- Values: SUM(card_limit)
- Filter: Top 10 only, sorted descending

🕵️ Client Drill-through Page

1. Create a new drill-through page.
2. Add drill-through field: client_id
3. Add visuals:
    - Table with card_type, card_brand, card_limit, expire_date, etc.
    - Filters: Add slicers for card_type, expire_date (Year)

🌡️ Heatmap Matrix: Expiry Trends

- Rows: card_brand
- Columns: Year(expire_date)
- Values: COUNT(card_number)

- Apply conditional formatting (e.g., red for high counts)

🔁 Dynamic Top N Chart

1. Create a slicer using:

DAX

КопироватьРедактировать

TopN_Selector = GENERATESERIES(1, 20, 1)

2. Create measure:

DAX

КопироватьРедактировать

TopN_Limit =

VAR SelectedN = SELECTEDVALUE(TopN_Selector[N])

RETURN

IF (

RANKX(ALL(Card_data[card_brand]), CALCULATE(SUM(Card_data[card_limit]))) <= SelectedN,

1,

0

)

3. Add Column/Bar Chart:
    - X-axis: card_brand
    - Y-axis: SUM(card_limit)
    - Visual-level filter: TopN_Limit = 1

📈 Sales.csv Task: Days Between Purchases

Goal: Average days between each sales date and the previous one for each customer.

Step 1: Sort sales data by customer and sales date

Step 2: Create calculated column:

DAX

КопироватьРедактировать

```
Previous_Sales_Date =
CALCULATE(
    MAX(sales[sales_date]),
    FILTER(
        sales,
        sales[customer_id] = EARLIER(sales[customer_id]) &&
        sales[sales_date] < EARLIER(sales[sales_date])
    )
)
```

Step 3: Calculate Day Difference

DAX

КопироватьРедактировать

```
Days_Between_Sales =
DATEDIFF([Previous_Sales_Date], [sales_date], DAY)
```

Step 4: Average the difference

DAX

КопироватьРедактировать

```
Avg_Days_Between =
AVERAGEX(
    FILTER(sales, NOT(ISBLANK([Days_Between_Sales]))),
```

```
    [Days_Between_Sales]
)
```