# POWER-BI SCENARIO BASED QUESTIONS

## QUESTION 13 & 14

- Sort the months as per FY
- Calculate dynamically Top N products

**- MAYURI .D.**

**QUESTION 13** Sort the month column as per financial year April to March

## SAMPLE TABLES

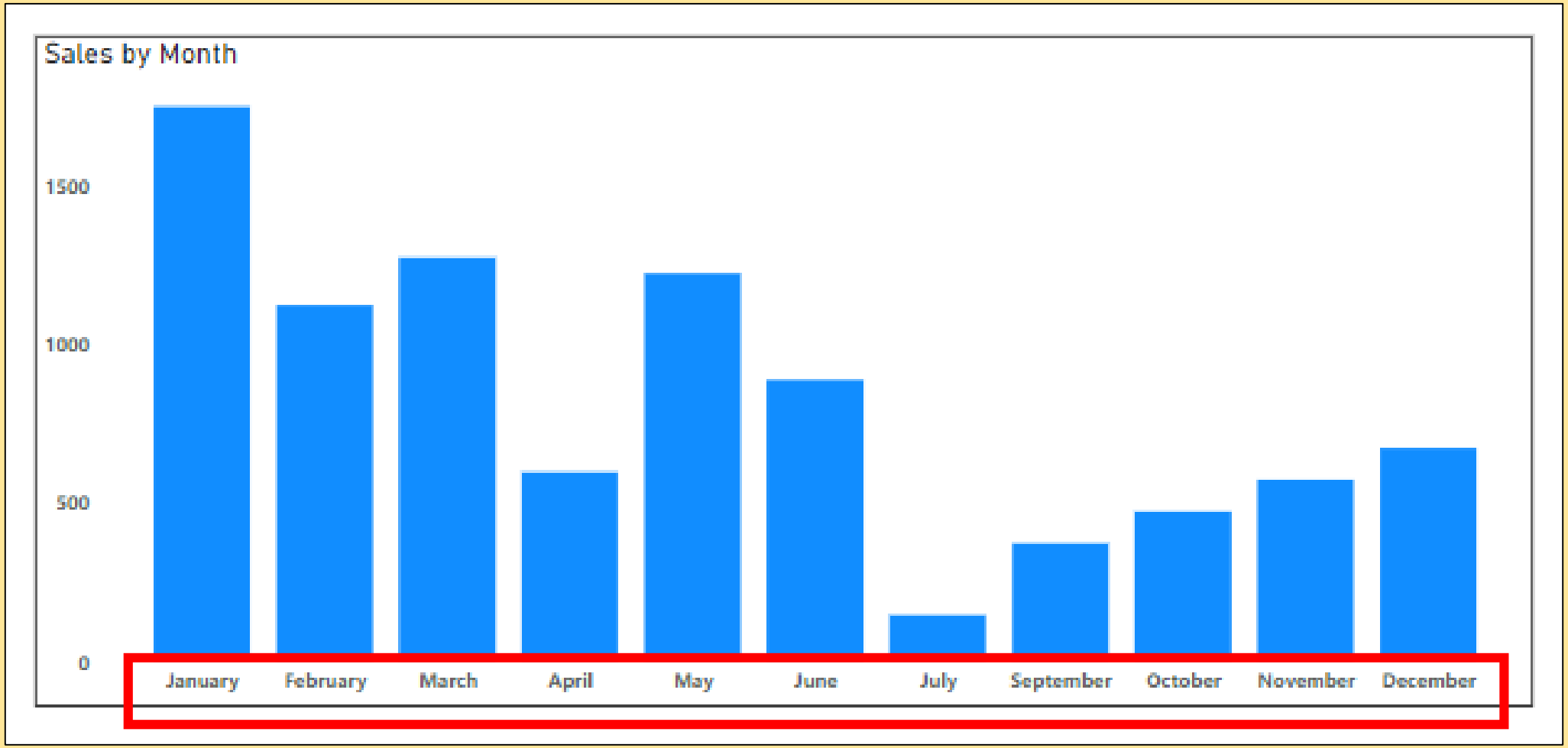
| product | Order Date | sales value |
|---|---|---|
| A | 01 January 2021 | 100 |
| A | 02 February 2021 | 150 |
| B | 03 March 2021 | 200 |
| B | 04 April 2021 | 600 |
| B | 05 May 2021 | 400 |
| C | 05 May 2021 | 550 |
| D | 07 June 2021 | 890 |
| D | 08 July 2021 | 150 |
| E | 09 May 2021 | 275 |
| E | 10 September 2021 | 375 |
| E | 11 October 2021 | 475 |
| E | 12 November 2021 | 575 |
| E | 13 December 2021 | 675 |
| E | 14 January 2022 | 775 |
| E | 15 January 2022 | 875 |
| E | 16 February 2022 | 975 |
| E | 17 March 2022 | 1075 |

Sales4 table

Calendar table

```
1 Calendar1 = ADDCOLUMNS(
2     CALENDAR(DATE(2020,01,01), TODAY()),
3     "Month", format([DATE],"mmmm"),
4     "month no", MONTH([DATE]),
5     "Day", DAY([DATE]),
6     "Year", YEAR([Date])
7 )
```

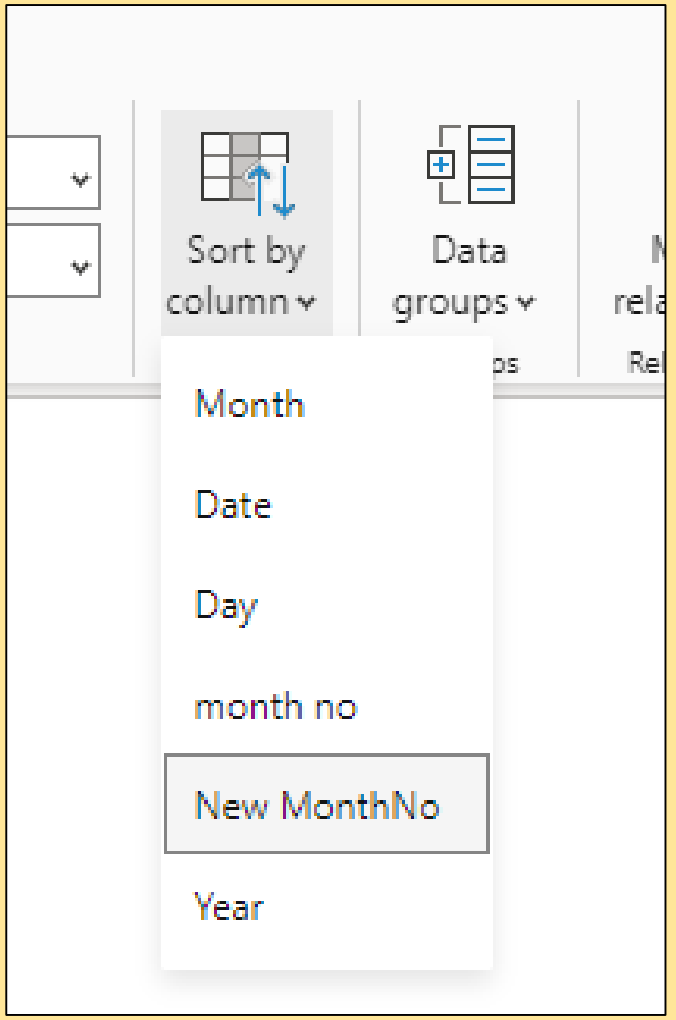
**Sort** x-axis from April to March instead of January to December



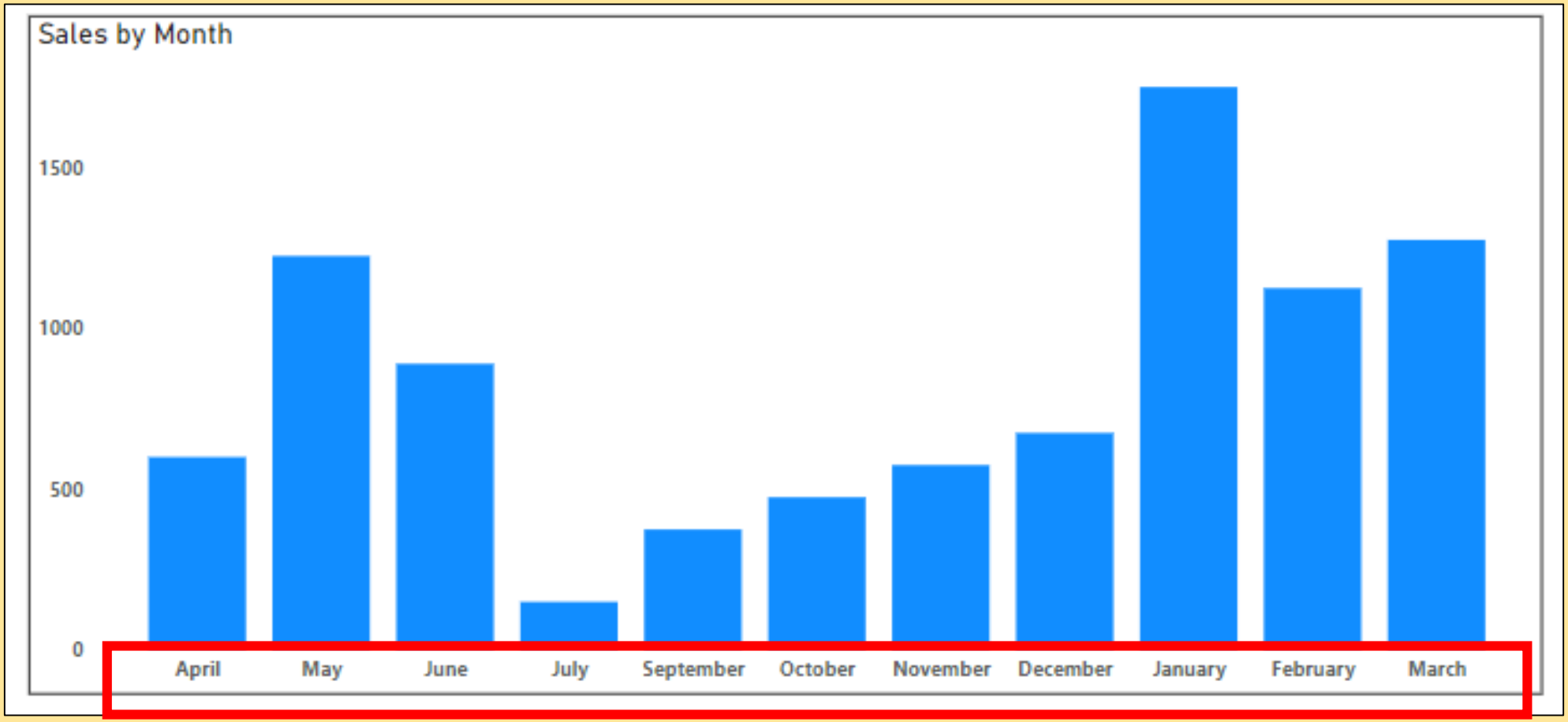
```
1 New monthNo = IF(
2     Calendar1[month no] > 3, Calendar1[month no] - 3,
3     Calendar1[month no] + 9
4 )
```

We need April as 1 and March as 12, so the condition will be
**If** month No is > 3 that is 4, **then** do 4 – 3 =1 i.e 4th month –April =1
**Else**
**If** it is not > 3 that is 2, **then** do 2 + 9 = 11 i.e 2nd month – February = 11

Select month column and click on
SORT BY COLUMN and
select New MonthNo



**Sorted** x-axis from April to March instead of January to December



**QUESTION 14** Calculate dynamically Top N values

## SAMPLE TABLE

Sales5 table

| product | order date | sales value | region | country |
|---|---|---|---|---|
| A | 02 July 2022 | 100 | Asia | India |
| A | 22 June 2022 | 150 | Asia | Srilanka |
| B | 10 May 2022 | 200 | Asia | Bangladesh |
| B | 10 April 2022 | 600 | Europe | Germany |
| B | 10 March 2022 | 400 | Europe | France |
| C | 11 February 2022 | 550 | North America | Mexico |
| D | 12 January 2022 | 890 | North America | Cuba |
| D | 12 January 2022 | 150 | Africa | South Africa |
| E | 10 March 2022 | 275 | Africa | Nigeria |
| G | 10 April 2022 | 340 | Africa | Nigeria |
| H | 30 March 2022 | 500 | Africa | Nigeria |
| I | 10 February 2022 | 290 | Africa | Nigeria |

Top Products table

| value |
|---|
| TOP 2 |
| TOP 3 |
| TOP 5 |

Create ranking measure on sales value

```
1 Total sales value = SUM(sales5[sales value])
```

```
Ranking1 = RANKX(ALL(sales5[product]),[Total sales value],,DESC,Dense)
```

| product | Total sales value | Ranking1 |
|---|---|---|
| A | 250 | 8 |
| B | 1200 | 1 |
| C | 550 | 3 |
| D | 1040 | 2 |
| E | 275 | 7 |
| G | 340 | 5 |
| H | 500 | 4 |
| I | 290 | 6 |

Here you can calculate dynamically TOP Products.
Remove columns sales value and ranking and insert **top N value**.
Filter it with slicer with values from Top Products table

```
1 Top N Values =
2 var selected_top = SELECTEDVALUE('Top Products'[value])
3 var top_products = SWITCH(selected_top,
4 "Top 2", IF([Ranking1] <= 2, [Total sales value]),
5 "TOP 3", IF([Ranking1] <= 3, [Total sales value]),
6 "TOP 5", IF([Ranking1] <= 5, [Total sales value]),
7 [Total sales value])
8 RETURN top_products
```


| product | Top N Values |
|---|---|
| B | 1200 |
| C | 550 |
| D | 1040 |

# RANKX

Returns the ranking of a number in a list of numbers for each row in the *table* argument.

## Syntax

DAX

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

# ALL

Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied. This function is useful for clearing filters and creating calculations on all the rows in a table.

## Syntax

DAX

```
ALL( [<table> | <column>[, <column>[, <column>[,…]]]] )
```

# SELECTEDVALUE

Returns the value when the context for columnName has been filtered down to one distinct value only. Otherwise returns alternateResult.

## Syntax

DAX

```
SELECTEDVALUE(<columnName>[, <alternateResult>])
```

# SWITCH

Evaluates an expression against a list of values and returns one of multiple possible result expressions. This function can be used to avoid having multiple nested IF statements.

## Syntax

DAX

```
SWITCH(<expression>, <value>, <result>[, <value>, <result>]…[, <else>])
```

# THANK YOU

- MAYURI .D.