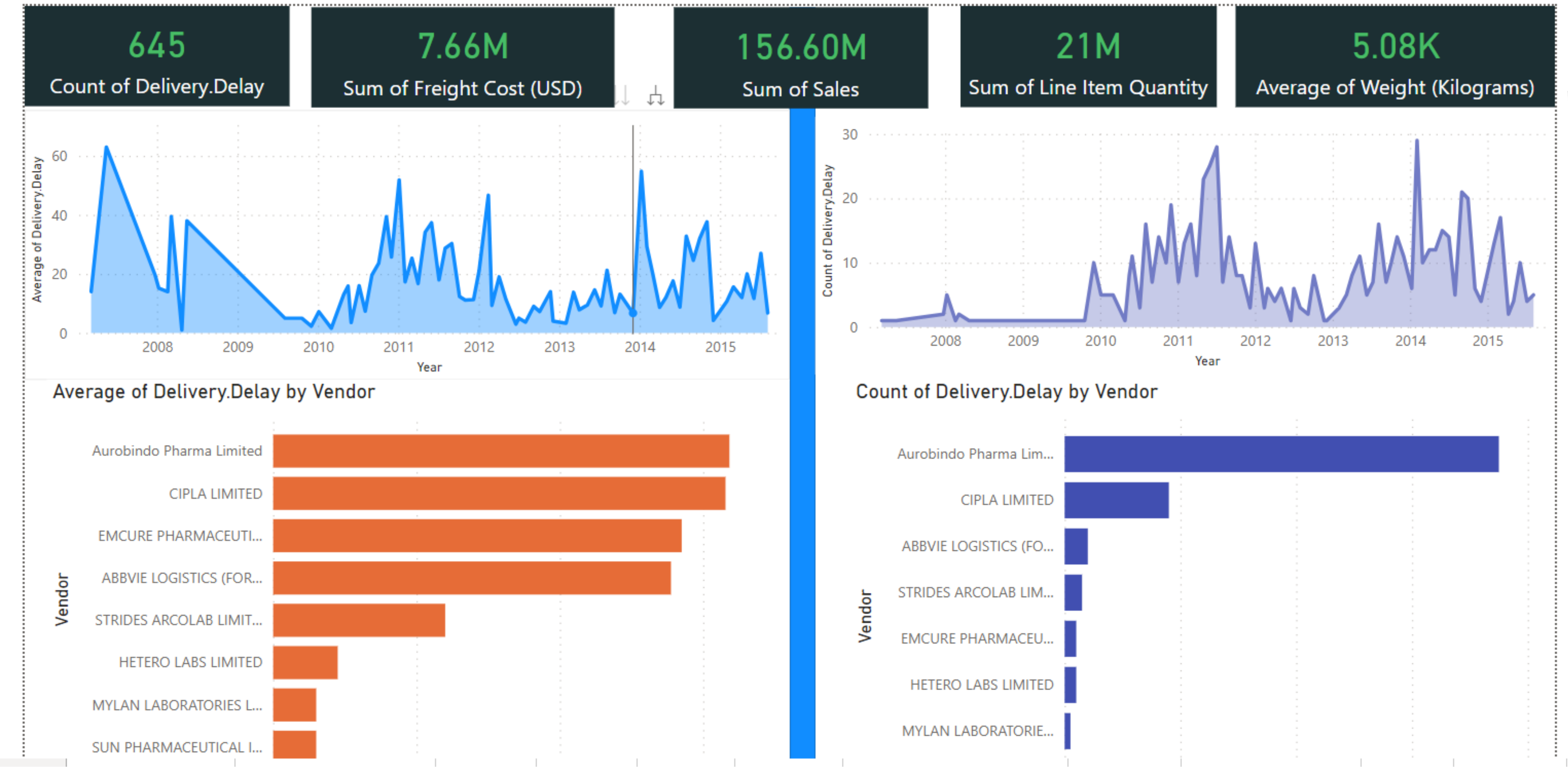
Intro to PowerBI

CS 459 Business Intelligence

Setup a Dashboard

Give a single perspective into the problem

Dashboard



645
Count of Delivery.Delay

7.66M
Sum of Freight Cost (USD)

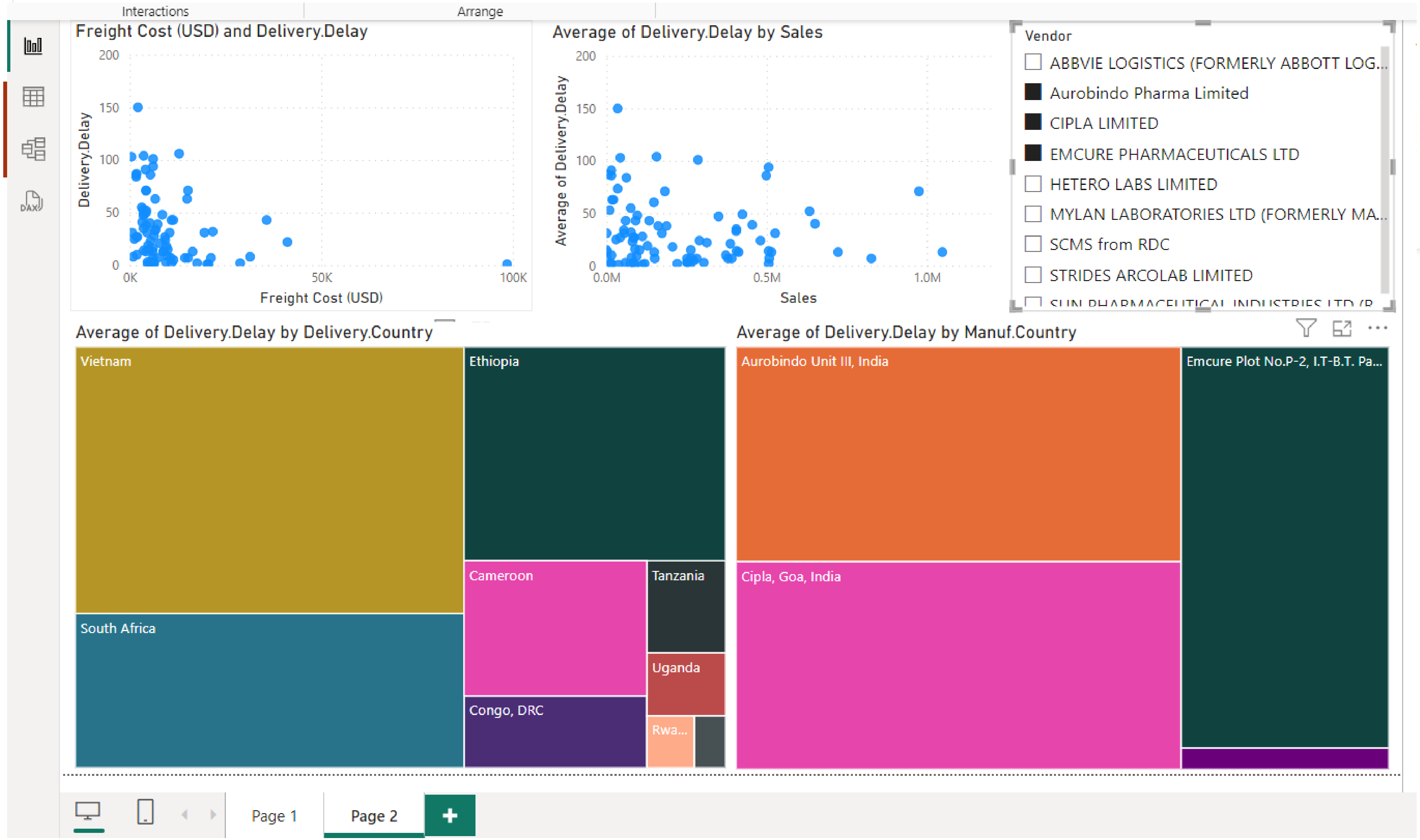
156.60M
Sum of Sales

21M
Sum of Line Item Quantity

5.08K
Average of Weight (Kilograms)

Average of Delivery.Delay by Vendor

Count of Delivery.Delay by Vendor

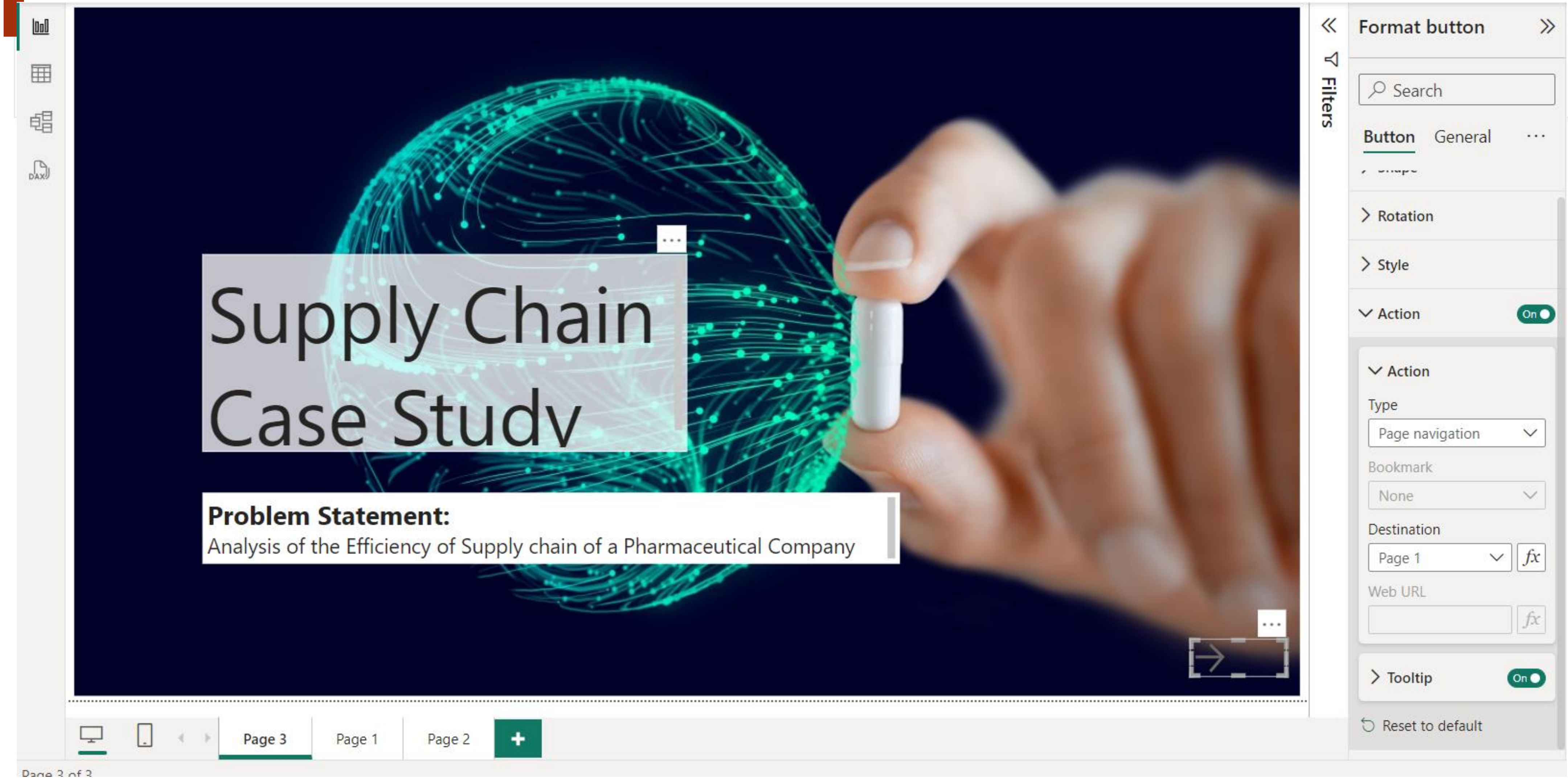

Interactions
Arrange
Freight Cost (USD) and Delivery.Delay
200
150
100
50
0
Delivery.Delay
0K
50K
100K
Freight Cost (USD)
Average of Delivery.Delay by Sales
200
150
100
50
0
Average of Delivery.Delay
0.0M
0.5M
1.0M
Sales
Average of Delivery.Delay by Delivery.Country
Vietnam
Ethiopia
Cameroon
Tanzania
South Africa
Uganda
Congo, DRC
Rwa...
Average of Delivery.Delay by Manuf.Country
Aurobindo Unit III, India
Emcure Plot No.P-2, I.T-B.T. Pa...
Cipla, Goa, India
Page 1
Page 2
+

Vendor

- ☐ ABBVIE LOGISTICS (FORMERLY ABBOTT LOG...
- ■ Aurobindo Pharma Limited
- ■ CIPLA LIMITED
- ■ EMCURE PHARMACEUTICALS LTD
- ☐ HETERO LABS LIMITED
- ☐ MYLAN LABORATORIES LTD (FORMERLY MA...
- ☐ SCMS from RDC
- ☐ STRIDES ARCOLAB LIMITED
- ☐ SUN PHARMACEUTICAL INDUSTRIES LTD (R

Create a Title Page

- Add buttons, text boxes, shapes and images.
- Add actions to form an interactive report



Problem Statement:
Analysis of the Efficiency of Supply chain of a Pharmaceutical Company

Action

Type: Page navigation

Bookmark: None

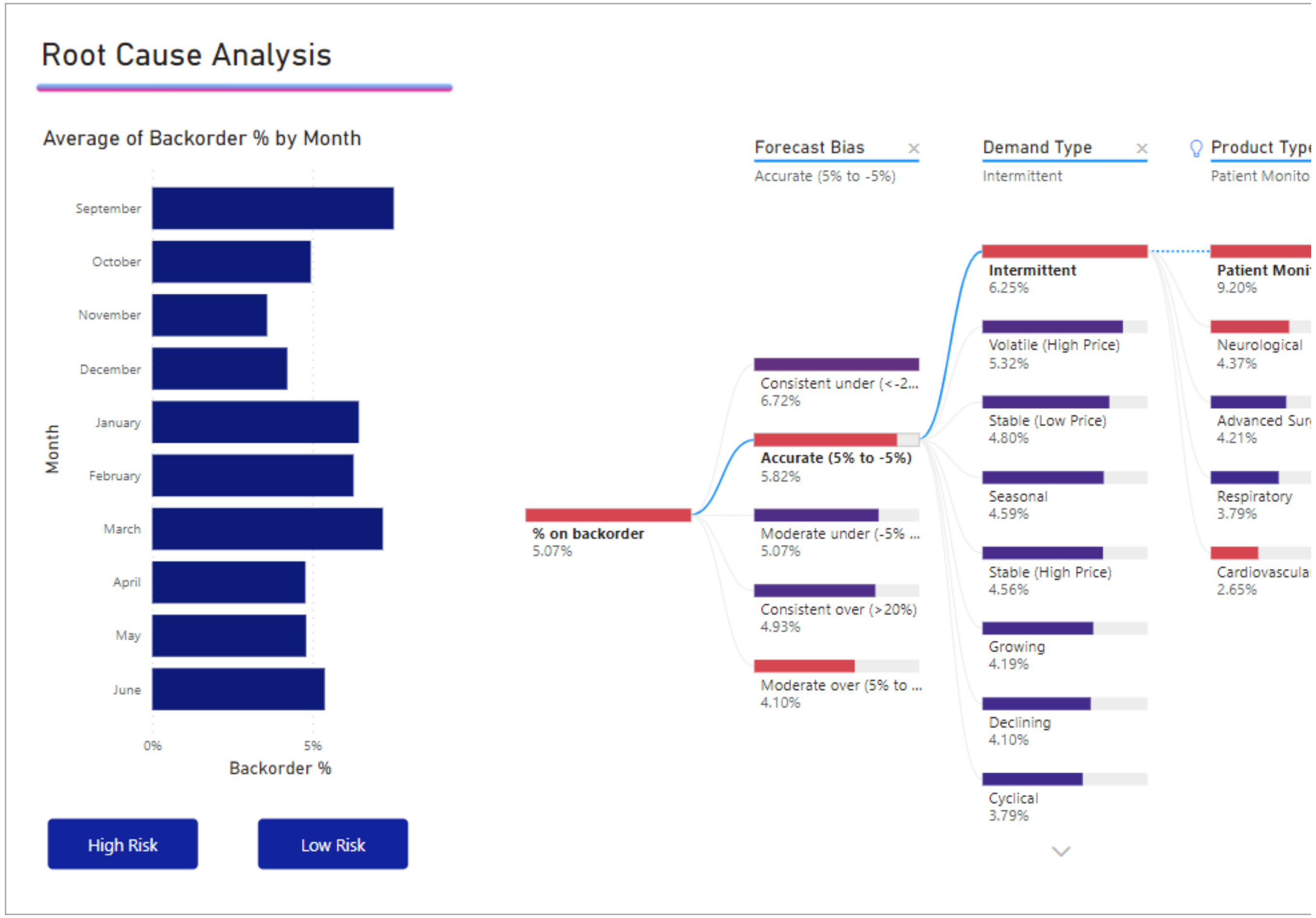
Destination: Page 1

PowerBI Part II

CS 459 Business Intelligence

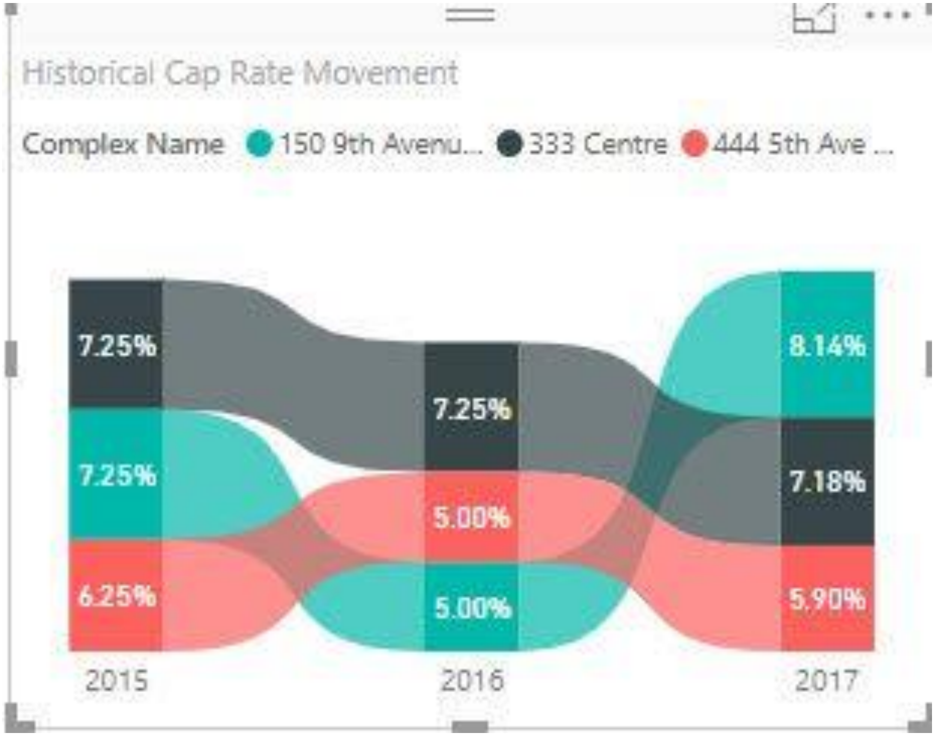
Decomposition tree

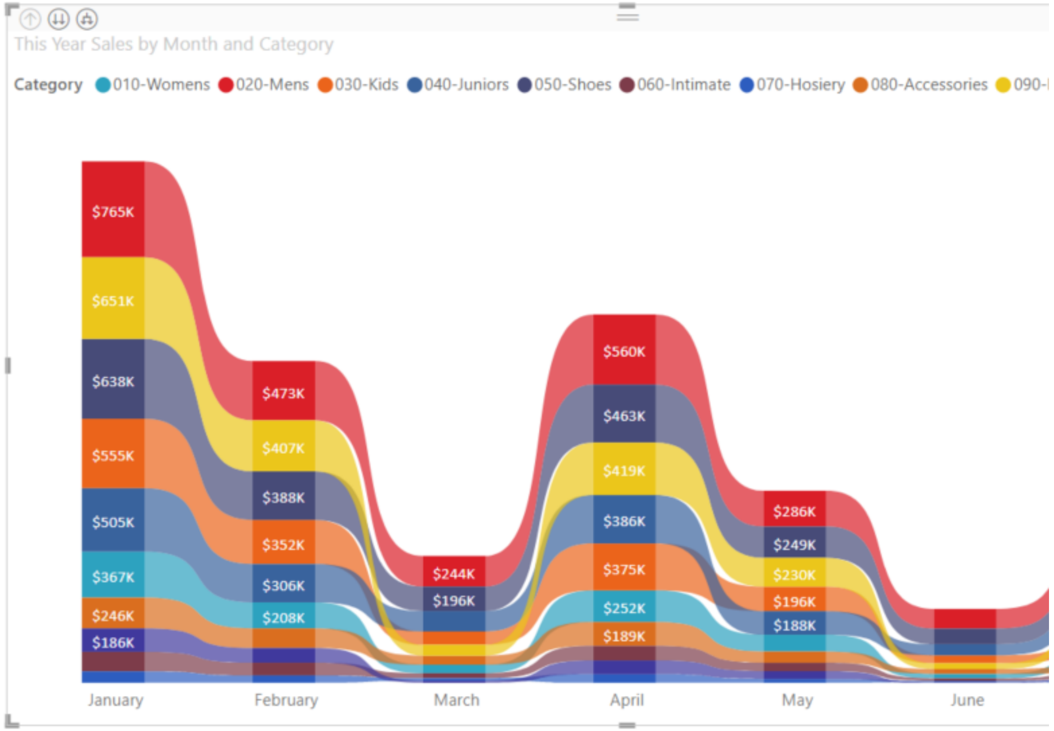
- The decomposition tree visual lets you visualize data across multiple dimensions. It automatically aggregates data and enables drilling down into your dimensions in any order. It is also an artificial intelligence (AI) visualization, so you can ask it to find the next dimension to drill down into based on certain criteria. This makes it a valuable tool for ad hoc exploration and conducting root cause analysis.



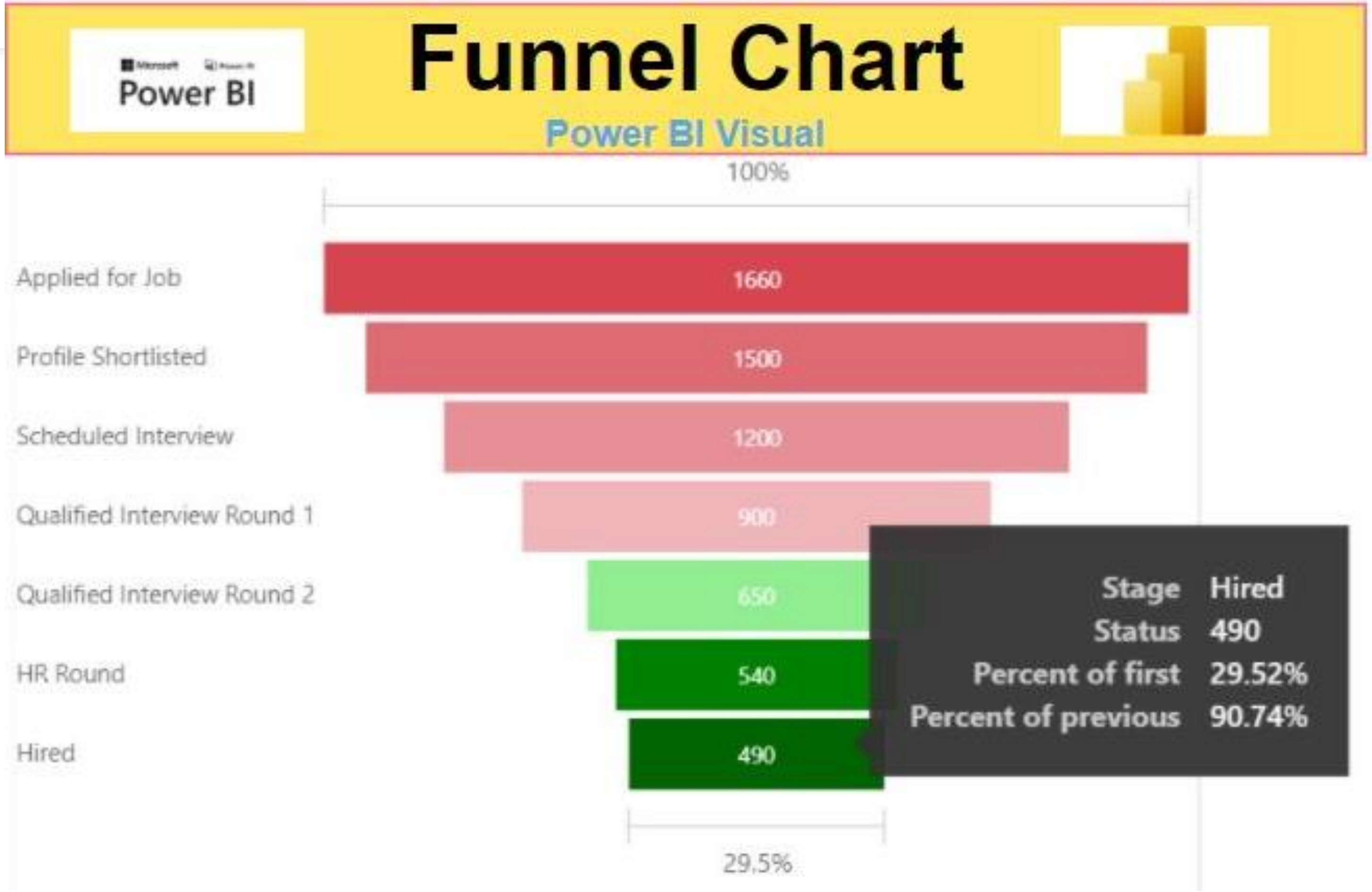
Ribbon Chart

- Power BI Ribbon Chart is useful to quickly identify which categorical data has the highest rank (large values).





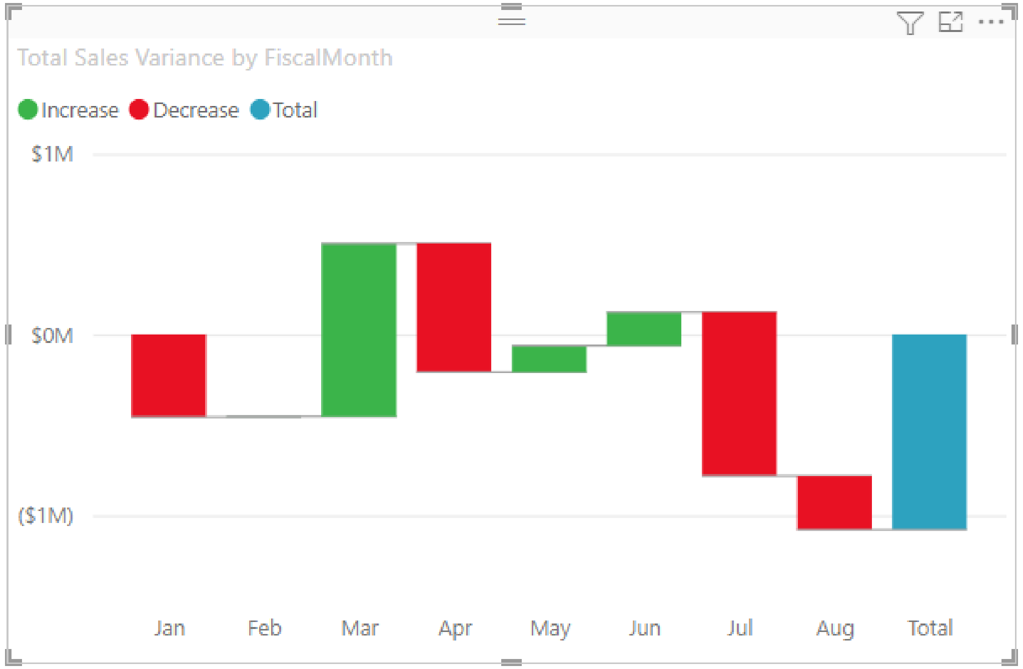
Funnel chart

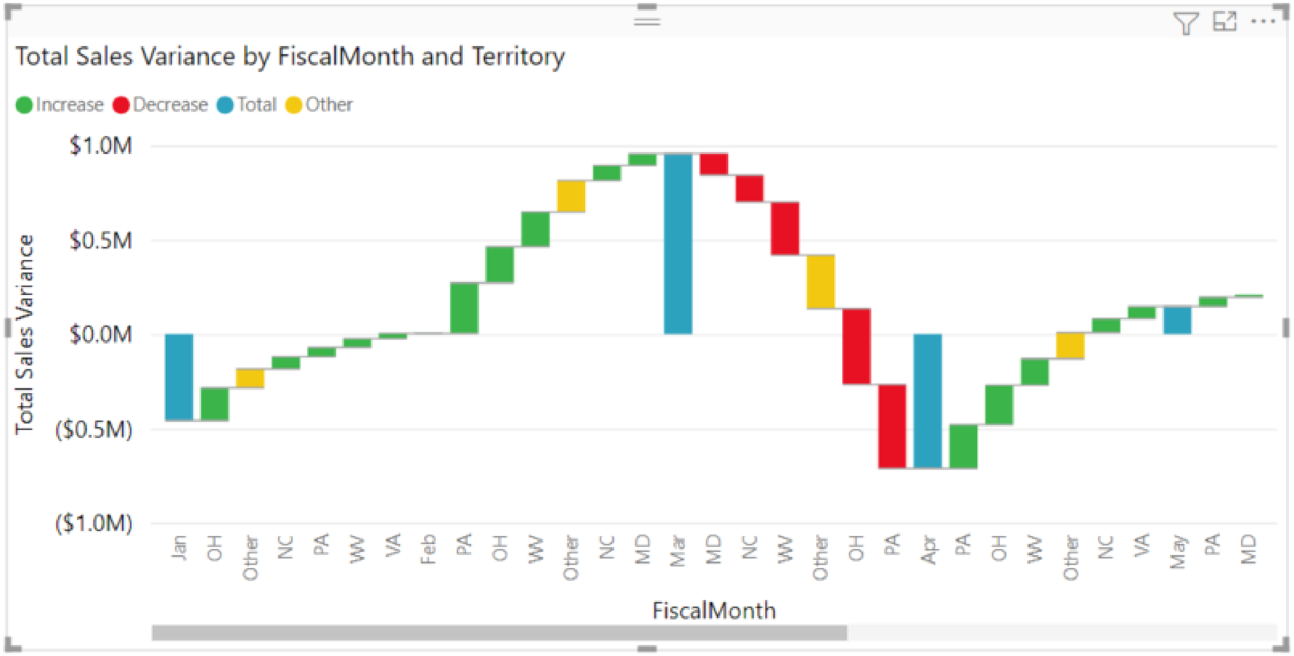


- Funnel charts show values across multiple stages in a process. For example, you could use a funnel chart to show the number of sales prospects at each stage in a sales pipeline. Typically, the values decrease gradually, allowing the bars to resemble a funnel.

Waterfall chart

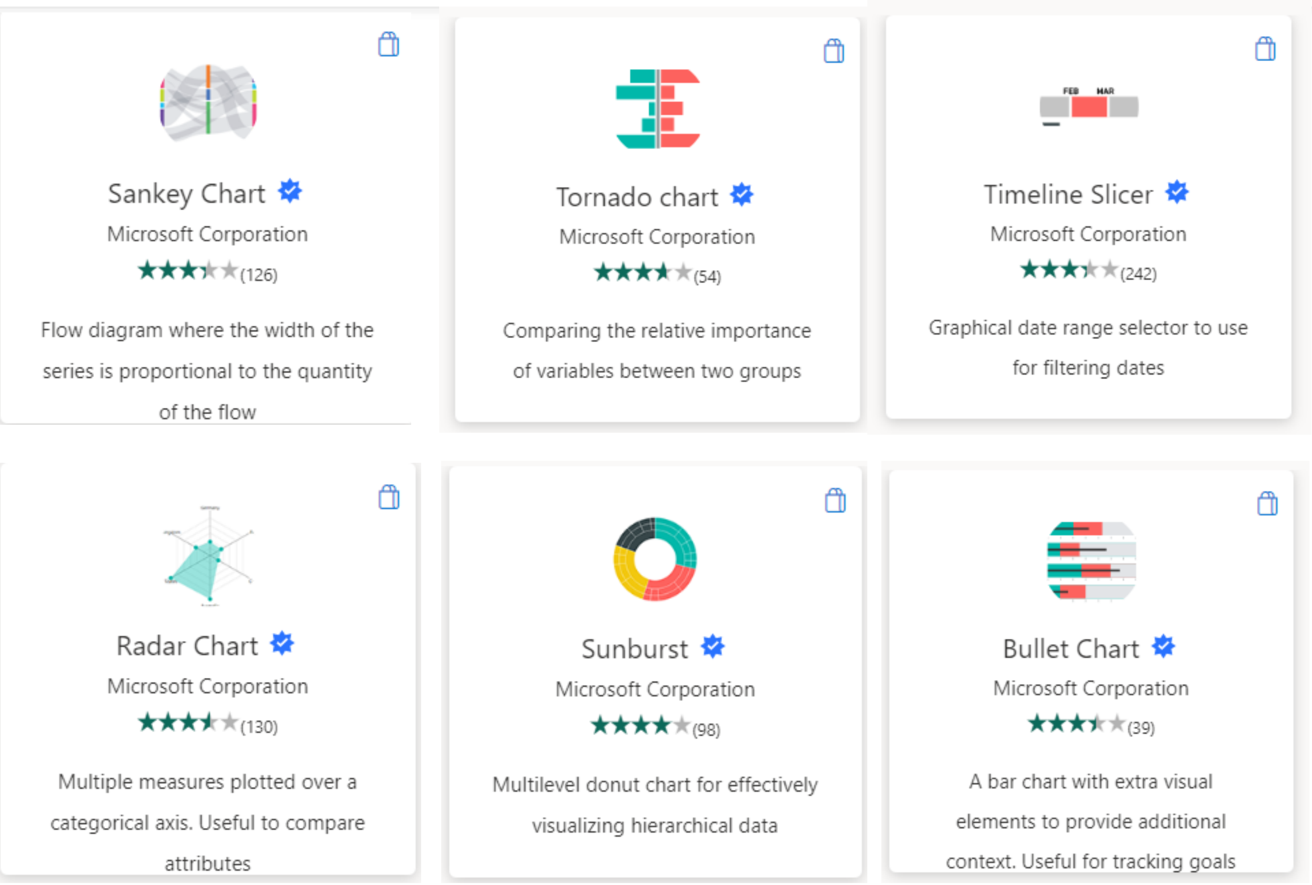
- Waterfall charts show a running total as Power BI adds and subtracts values. These charts are useful for understanding how an initial value (like net income) is affected by a series of positive and negative changes. The columns are color coded so you can quickly notice increases and decreases.





Get more visuals…

- Sankey
- Tornado
- Timeline Slicer
- Radar
- Sunburst
- Bullet



Sankey Chart
Microsoft Corporation
(126)
Flow diagram where the width of the series is proportional to the quantity of the flow

Tornado chart
Microsoft Corporation
(54)
Comparing the relative importance of variables between two groups

Timeline Slicer
Microsoft Corporation
(242)
Graphical date range selector to use for filtering dates

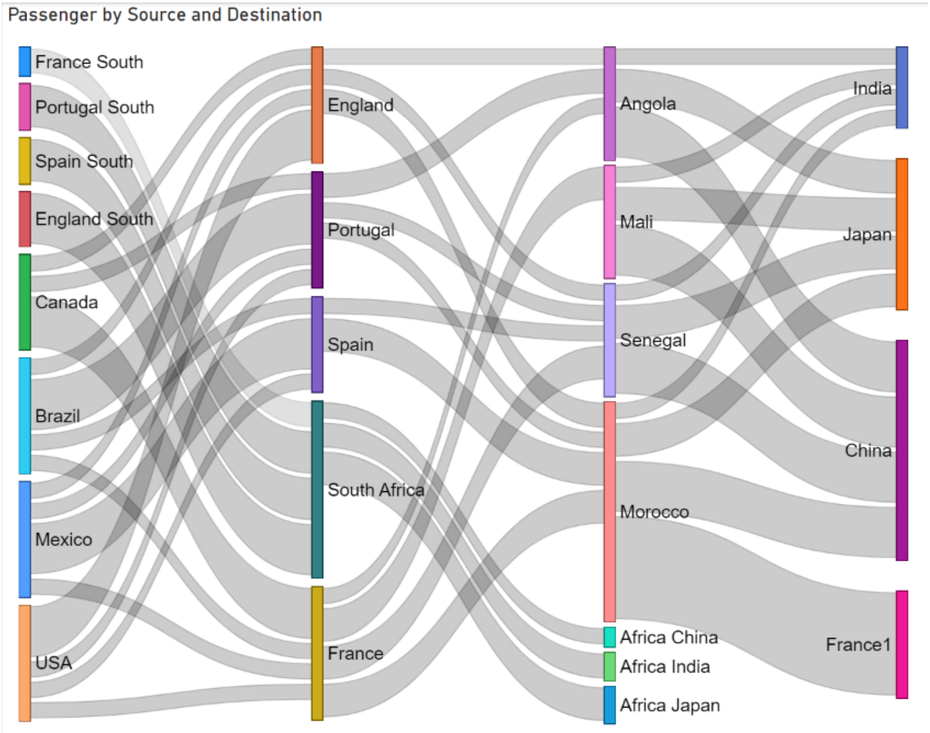
Radar Chart
Microsoft Corporation
(130)
Multiple measures plotted over a categorical axis. Useful to compare attributes

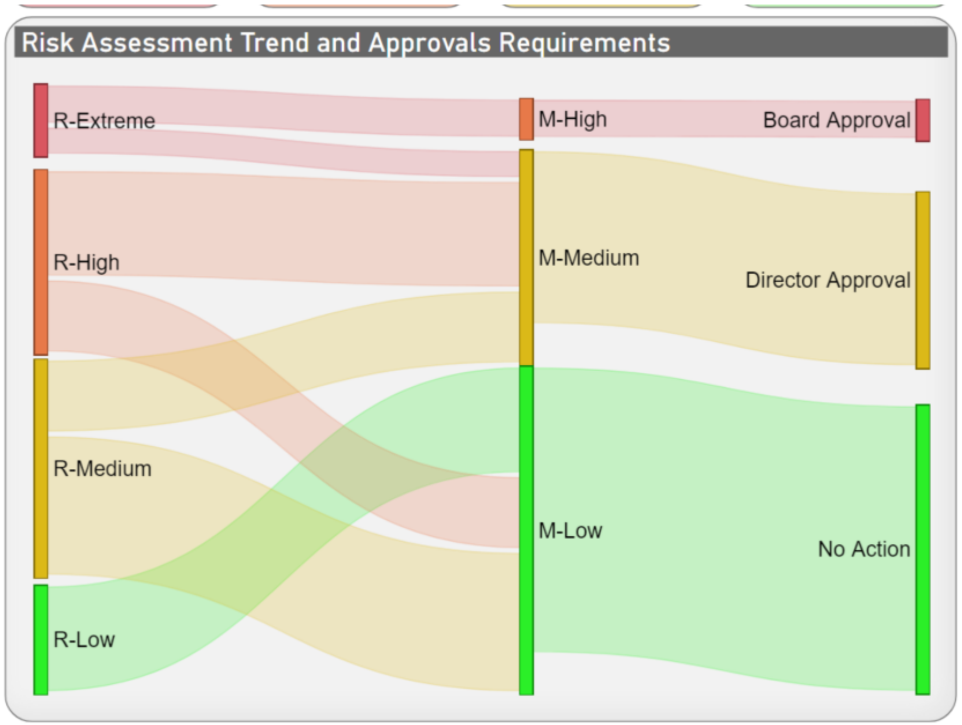
Sunburst
Microsoft Corporation
(98)
Multilevel donut chart for effectively visualizing hierarchical data

Bullet Chart
Microsoft Corporation
(39)
A bar chart with extra visual elements to provide additional context. Useful for tracking goals

Sankey chart

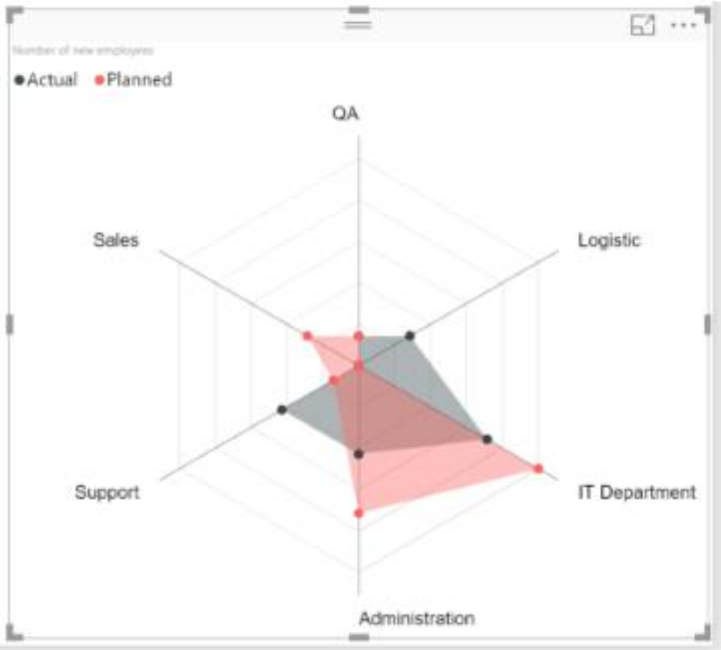
- The Sankey chart is a beautiful visualization used to illustrate a flow of data (source –destination) from multiple levels and to multiple destinations. The things being connected are called nodes and the connections are called links.

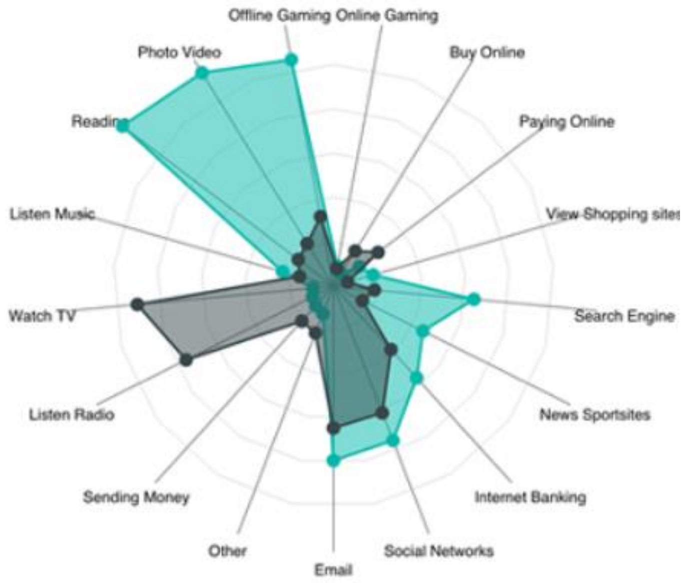


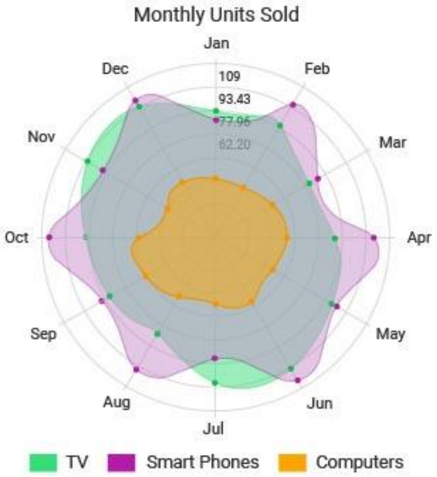


Radar chart

- A radar chart is used to display multiple categories of data on the chart. It is also known as a web chart, spider chart.



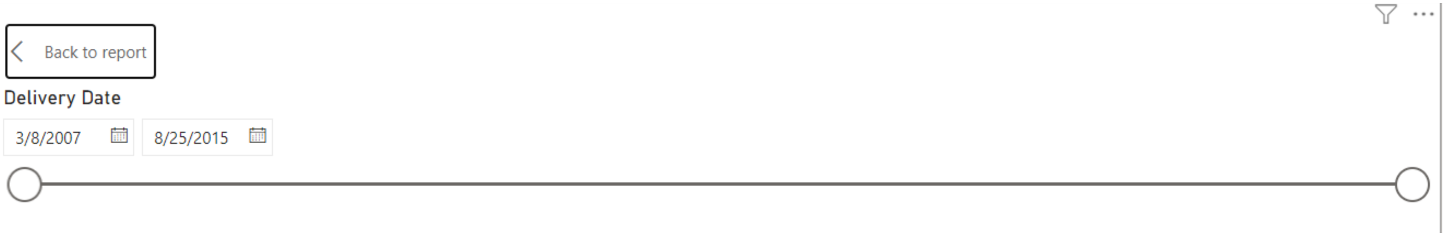




PowerBI Contd.

CS 459 Business Intelligence

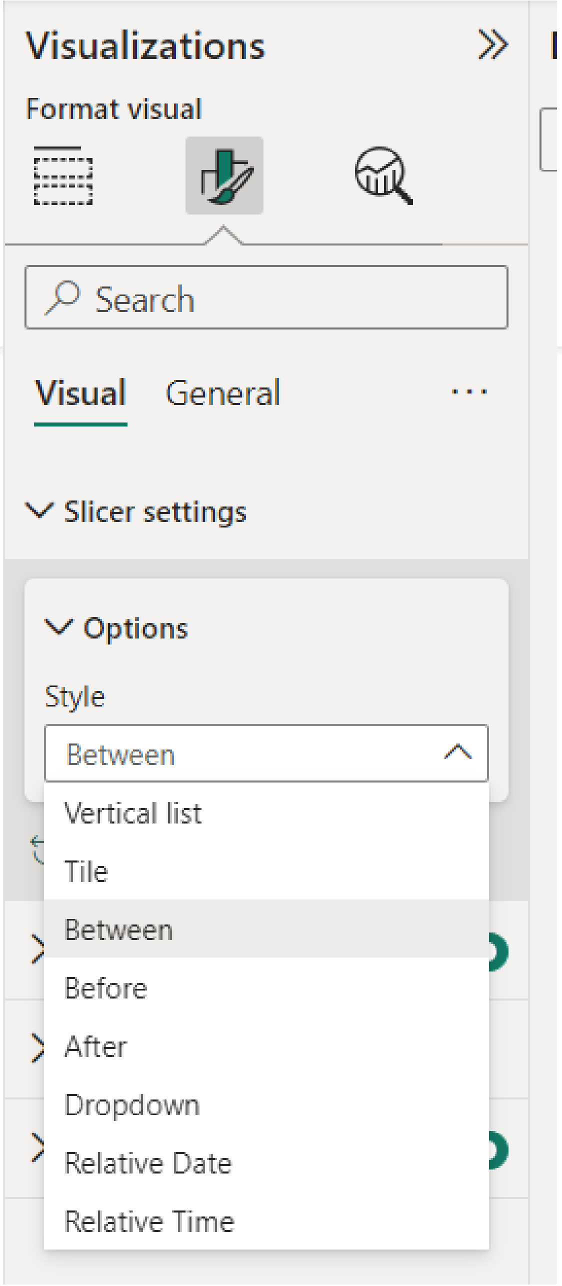
Slicer Types



Delivery Date

3/8/2007 8/25/2015

- Slicer for Delivery Date
- Default – Time Slicer due to Date format
- Switch to Vertical List
- Enable Date hierarchy
- Adjust the Date hierarchy to show years only
- You may try other slicer settings to explore.



Style: Between

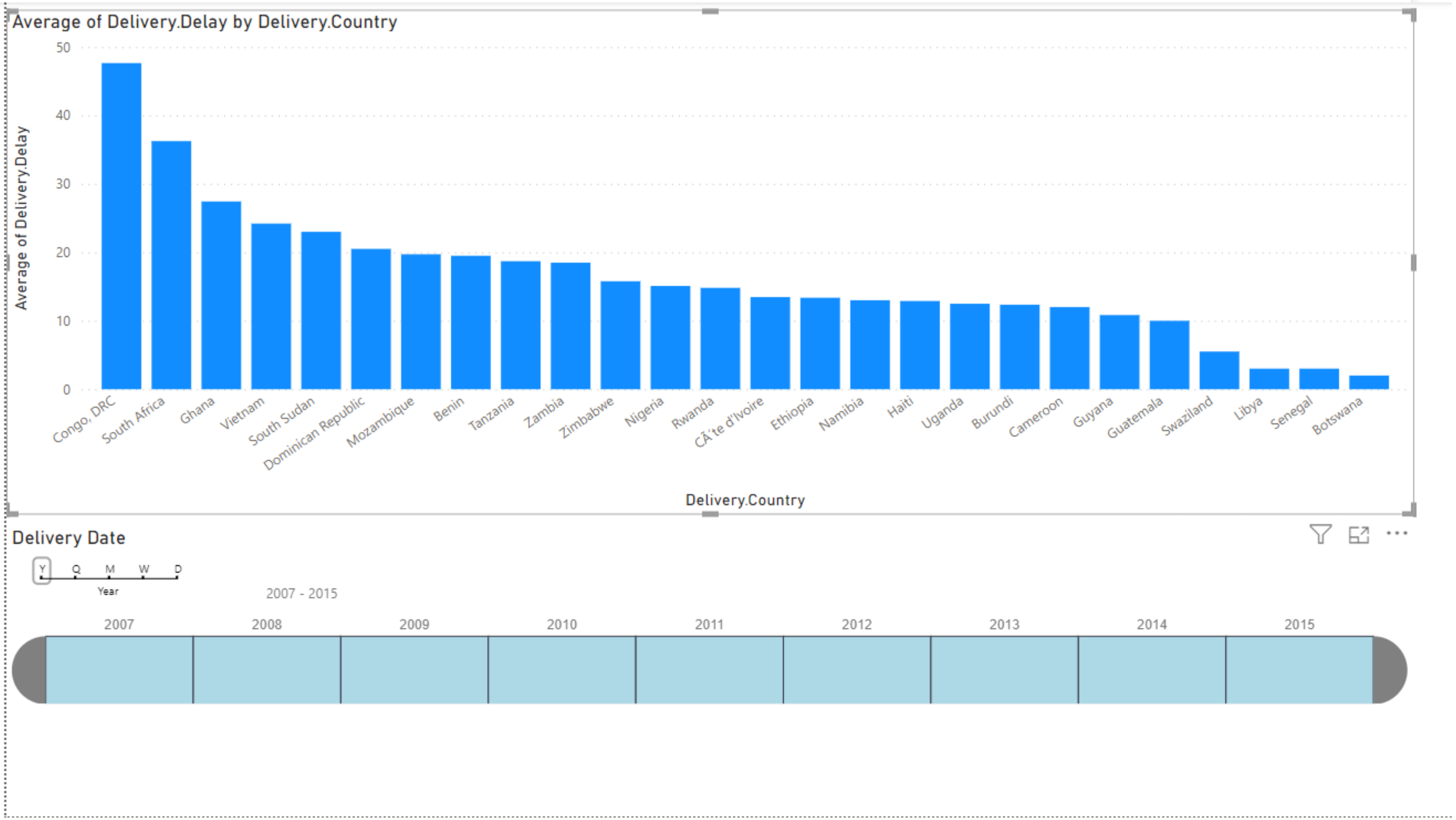
Timeline Slicer

- Graphical date range selector to use for filtering dates

Try it Yourself

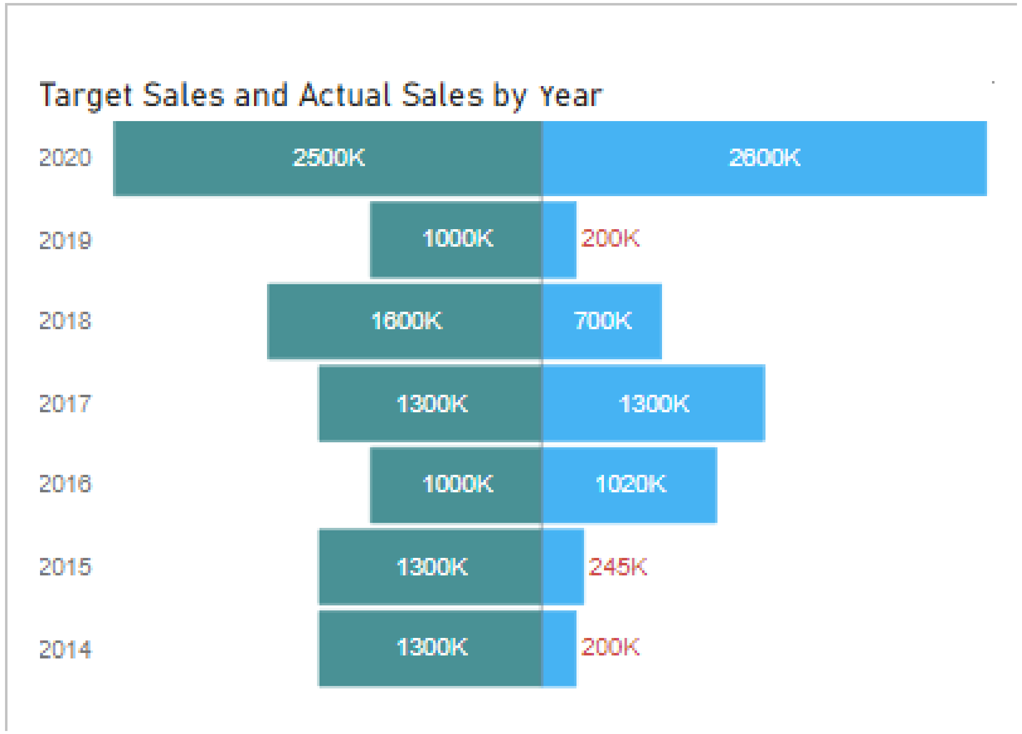
Delivery country with Average of Delivery Delay.

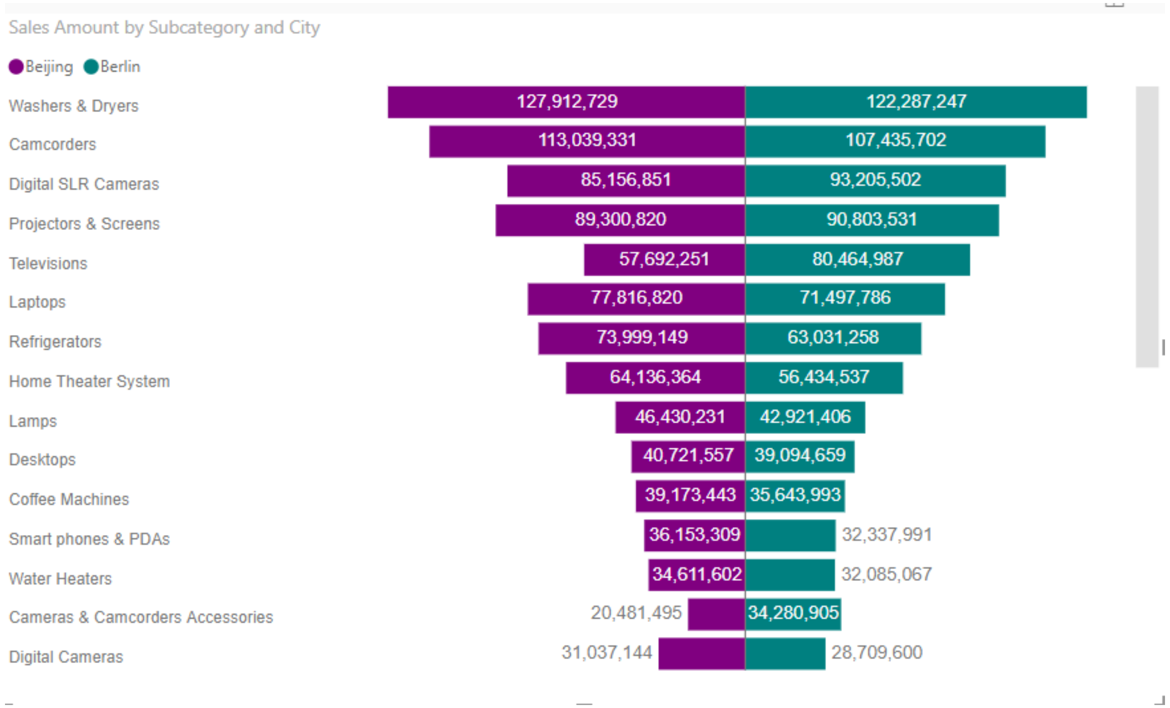
Explore the timeline slicer.



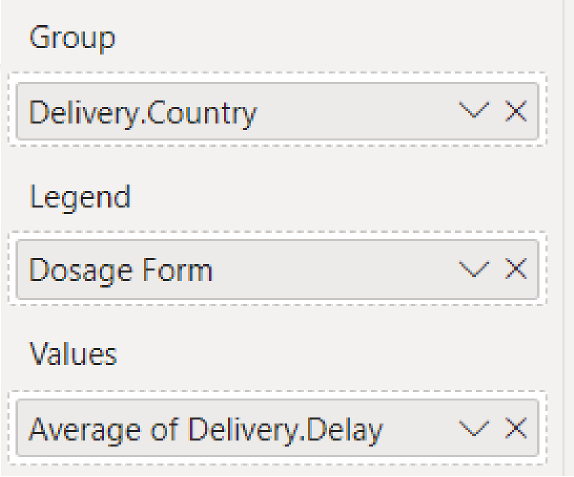
Tornado chart

- Tornado charts, are a special type of Bar chart, where the data categories are listed vertically instead of the standard horizontal presentation, and the categories are ordered so that the largest bar appears at the top of the chart, the second largest appears second from the top, and so on. It allows for a side-by-side comparison for 2 things.





Tornado



Group

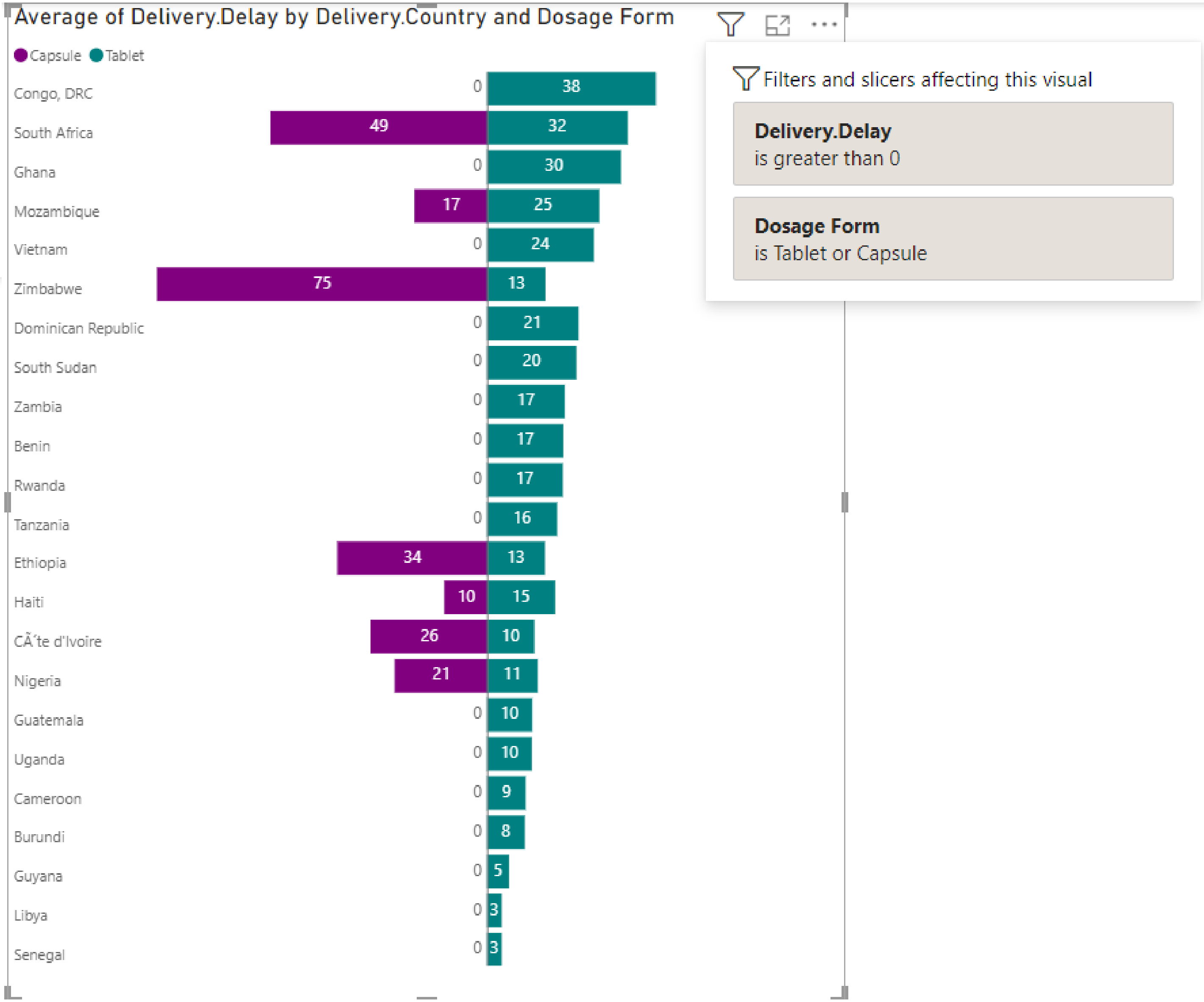
Delivery.Country

Legend

Dosage Form

Values

Average of Delivery.Delay



Average of Delivery.Delay by Delivery.Country and Dosage Form

Filters and slicers affecting this visual

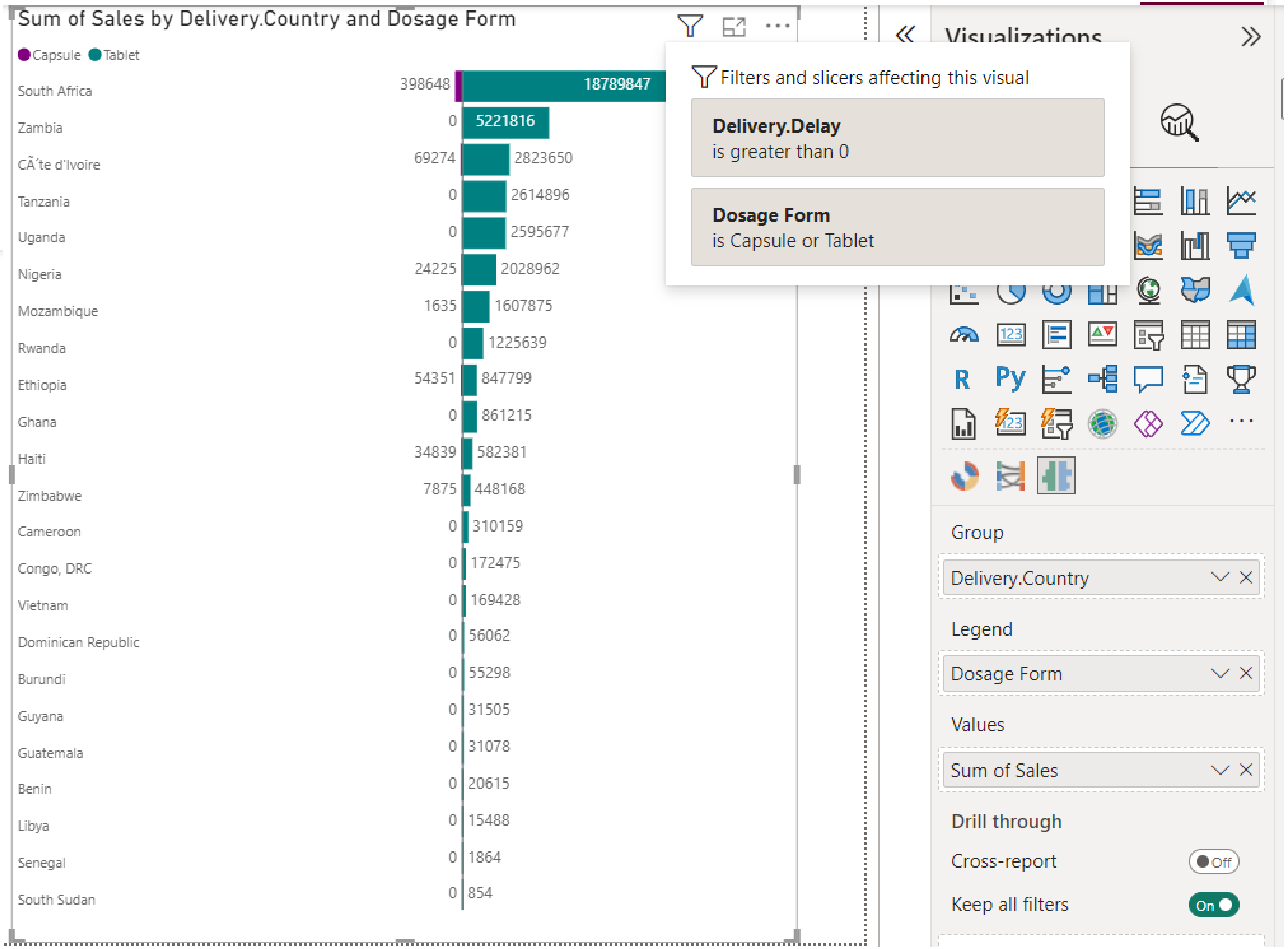
Delivery.Delay
is greater than 0

Dosage Form
is Tablet or Capsule

Comparing Delay of Tablets Vs Capsules

Filter Dosage Form – select Tablet and Capsule

Tornado - Example



Filters and slicers affecting this visual

Delivery.Delay
is greater than 0

Dosage Form
is Capsule or Tablet

Group: Delivery.Country

Legend: Dosage Form

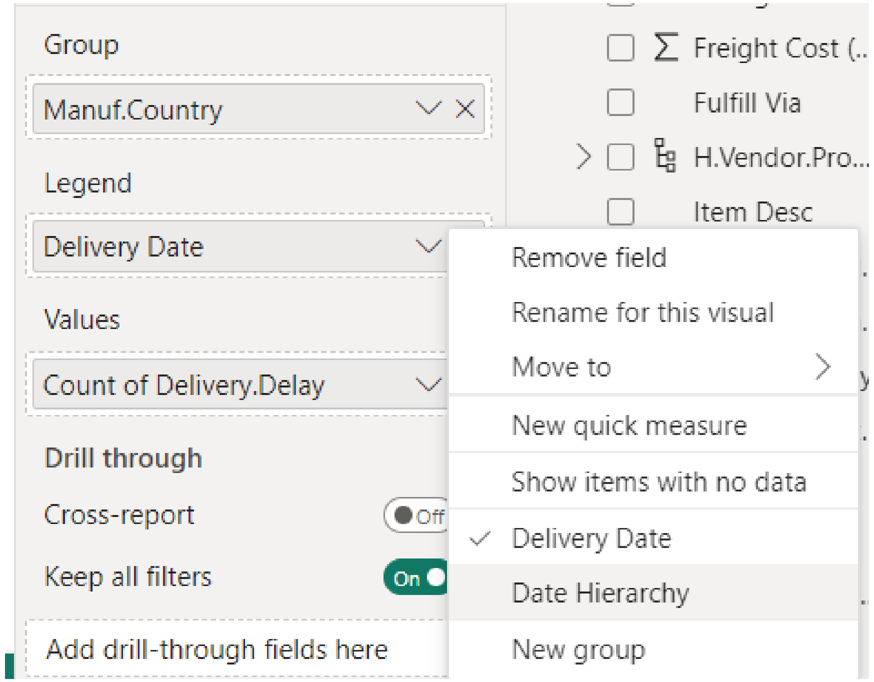
Values: Sum of Sales

Drill through

Cross-report: Off

Keep all filters: On

Tornado – Year Comparison



Group

Manuf.Country

Legend

Delivery Date

Values

Count of Delivery.Delay

Drill through

Cross-report Off

Keep all filters On

Add drill-through fields here

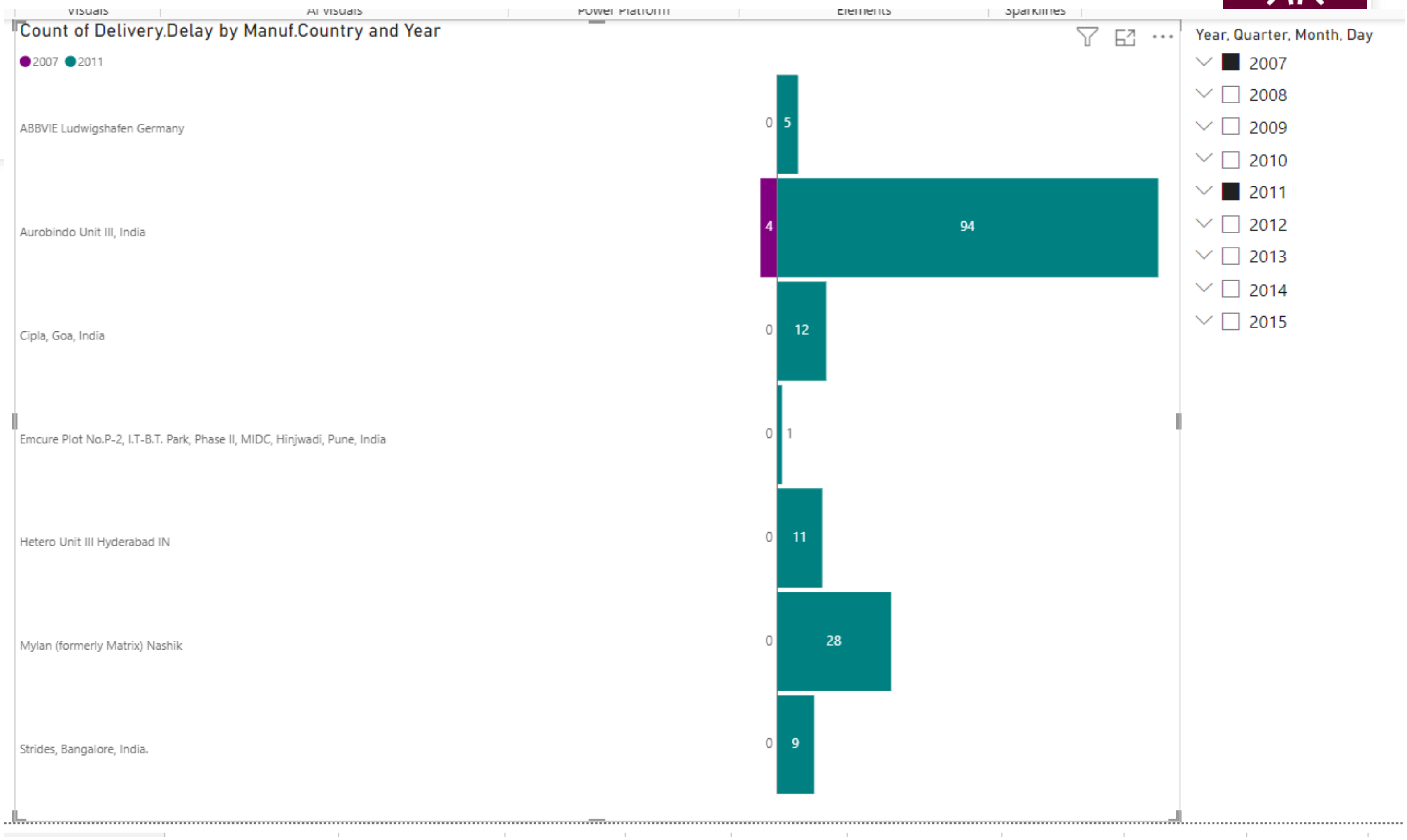
2-year Delay Comparison

Switch to Delivery Date Hierarchy.

Add a slicer to control the 2 years selected for comparison

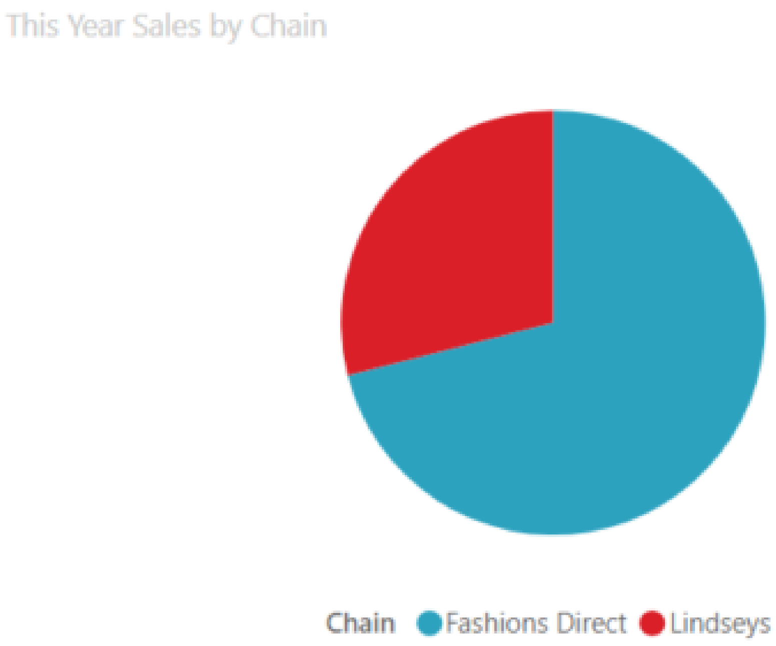
Insert Slicer → Delivery Date

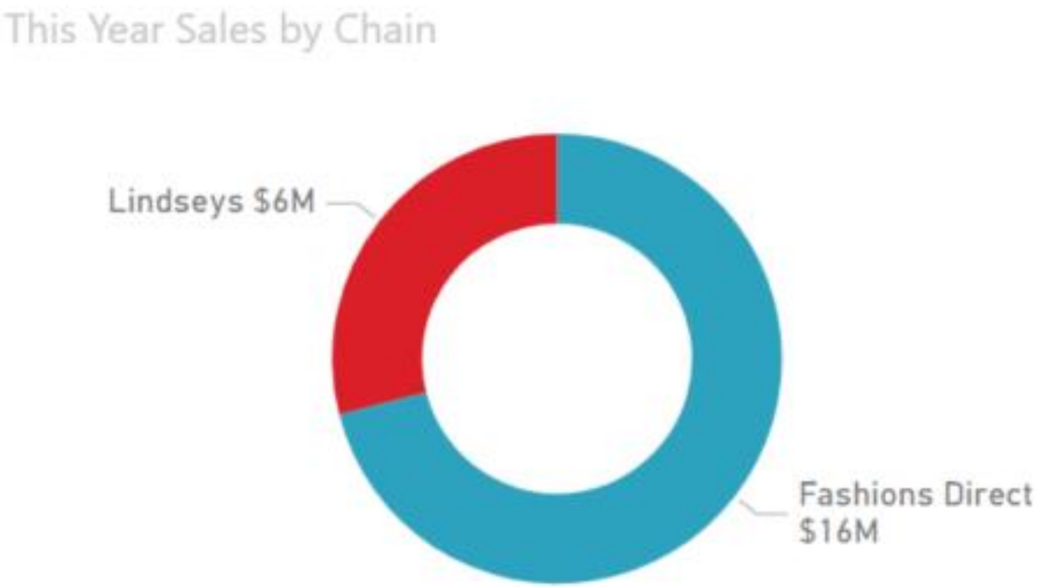
Tornado – 2-year comparison of Count of Delivery Delay for each Manufacturing Country


Count of Delivery.Delay by Manuf.Country and Year
● 2007 ● 2011
ABBVIE Ludwigshafen Germany
0
5
Aurobindo Unit III, India
4
94
Cipla, Goa, India
0
12
Emcure Plot No.P-2, I.T-B.T. Park, Phase II, MIDC, Hinjwadi, Pune, India
0
1
Hetero Unit III Hyderabad IN
0
11
Mylan (formerly Matrix) Nashik
0
28
Strides, Bangalore, India.
0
9
Year, Quarter, Month, Day
■ 2007
□ 2008
□ 2009
□ 2010
■ 2011
□ 2012
□ 2013
□ 2014
□ 2015

Doughnut charts and Pie Charts

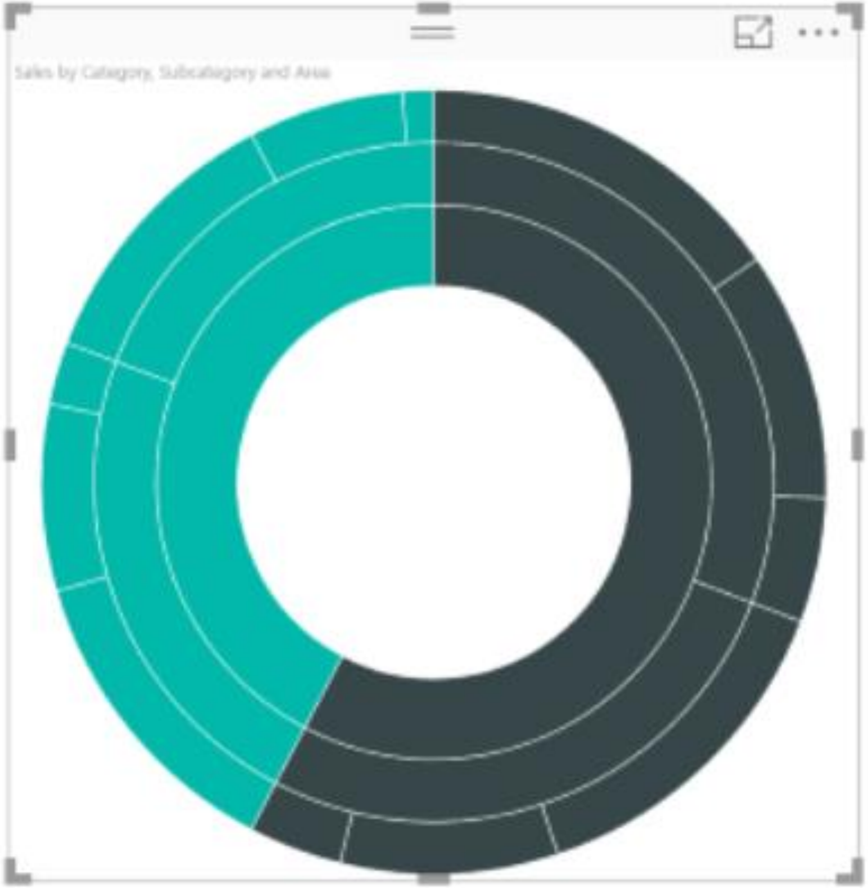
- Doughnut charts are similar to pie charts. They show the relationship of parts to a whole. The only difference is that the center is blank and allows space for a label or icon.



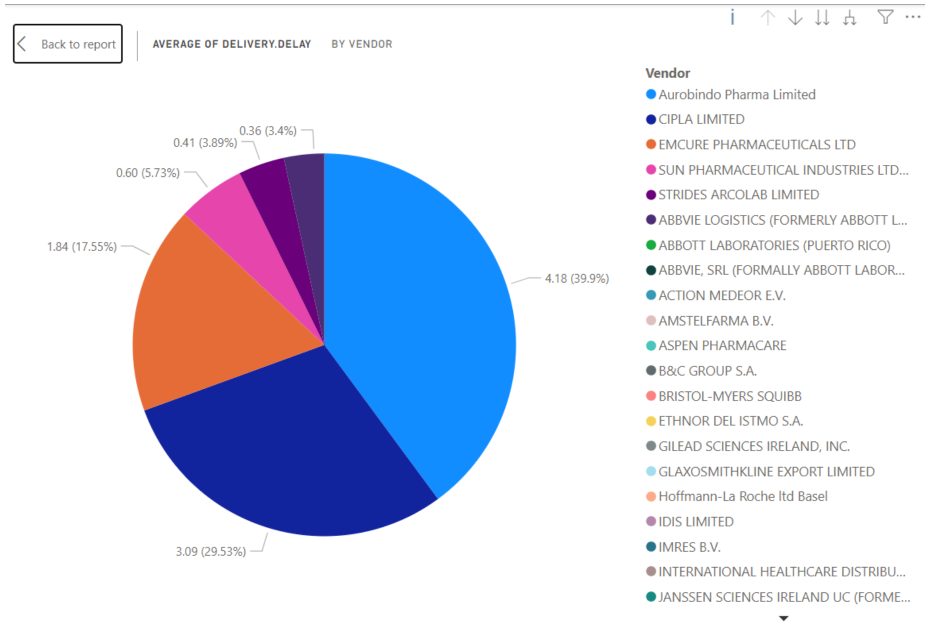


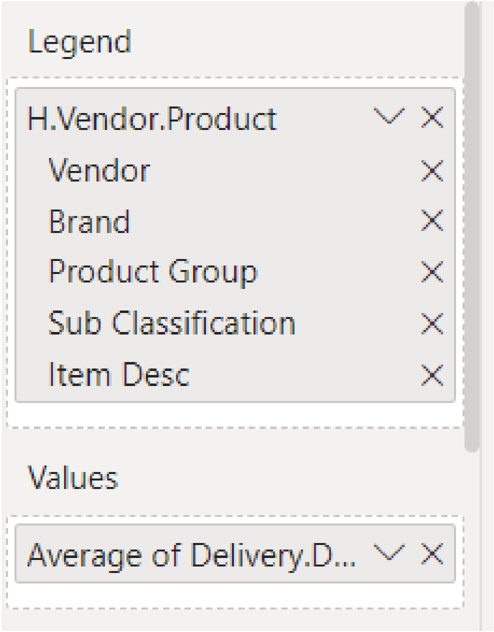
Sunburst

- Multilevel donut chart for effectively visualizing hierarchical data
- Sunburst chart is used to visualize hierarchical data, depicted by concentric circles. The circle in the center represents the root node, with the hierarchy moving outward from the center.
- A segment of the inner circle bears a hierarchical relationship to those segments of the outer circle which lie within the angular sweep of the parent segment.



Pie VS Donut





Legend

H.Vendor.Product ∨ ✕

Vendor ✕

Brand ✕

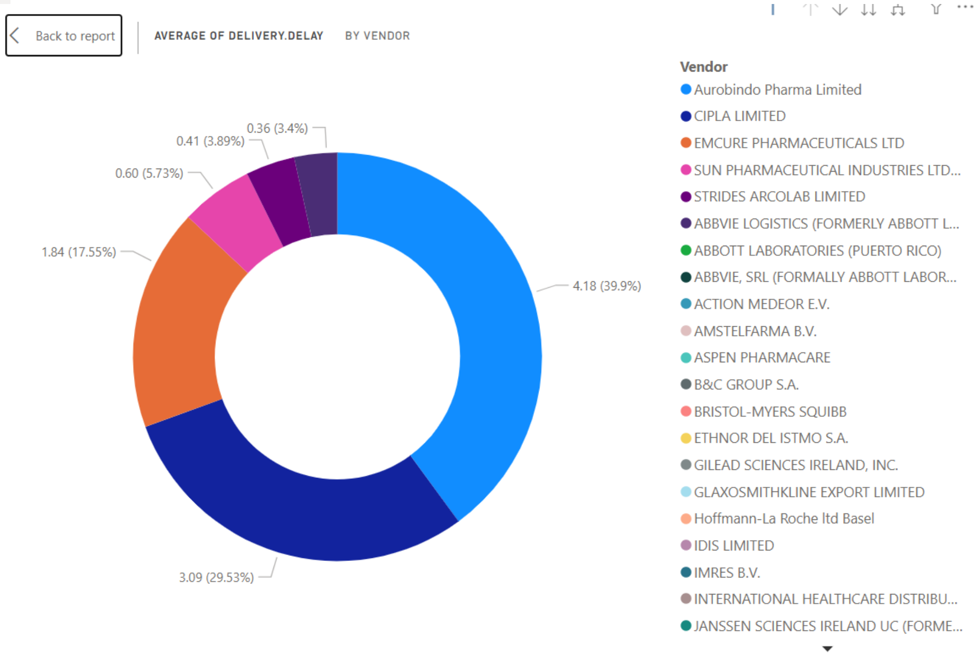
Product Group ✕

Sub Classification ✕

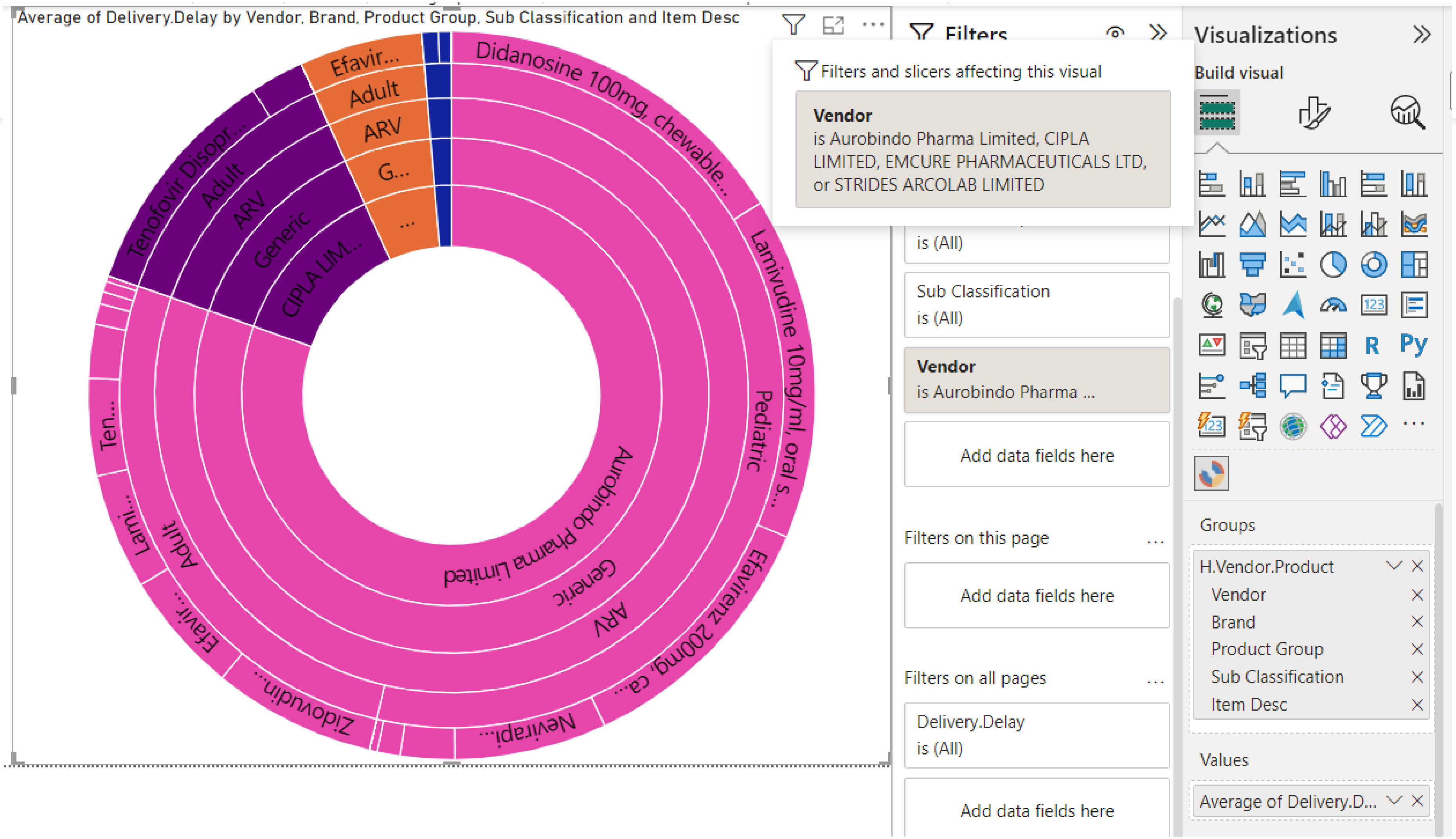
Item Desc ✕

Values

Average of Delivery.D... ∨ ✕



Sunburst



Average of Delivery.Delay by Vendor, Brand, Product Group, Sub Classification and Item Desc

Filters

Filters and slicers affecting this visual

Vendor
is Aurobindo Pharma Limited, CIPLA LIMITED, EMCURE PHARMACEUTICALS LTD, or STRIDES ARCOLAB LIMITED

is (All)

Sub Classification
is (All)

Vendor
is Aurobindo Pharma ...

Add data fields here

Filters on this page

Add data fields here

Filters on all pages

Delivery.Delay
is (All)

Add data fields here

Visualizations

Build visual

Groups

H.Vendor.Product
- Vendor
- Brand
- Product Group
- Sub Classification
- Item Desc

Values

Average of Delivery.D...

The Art of Story Telling

- Tell the data narrative page by page → takes shape of a report
- Text helps to show interpretation – keep in mind your client.

Dashboard Vs Report

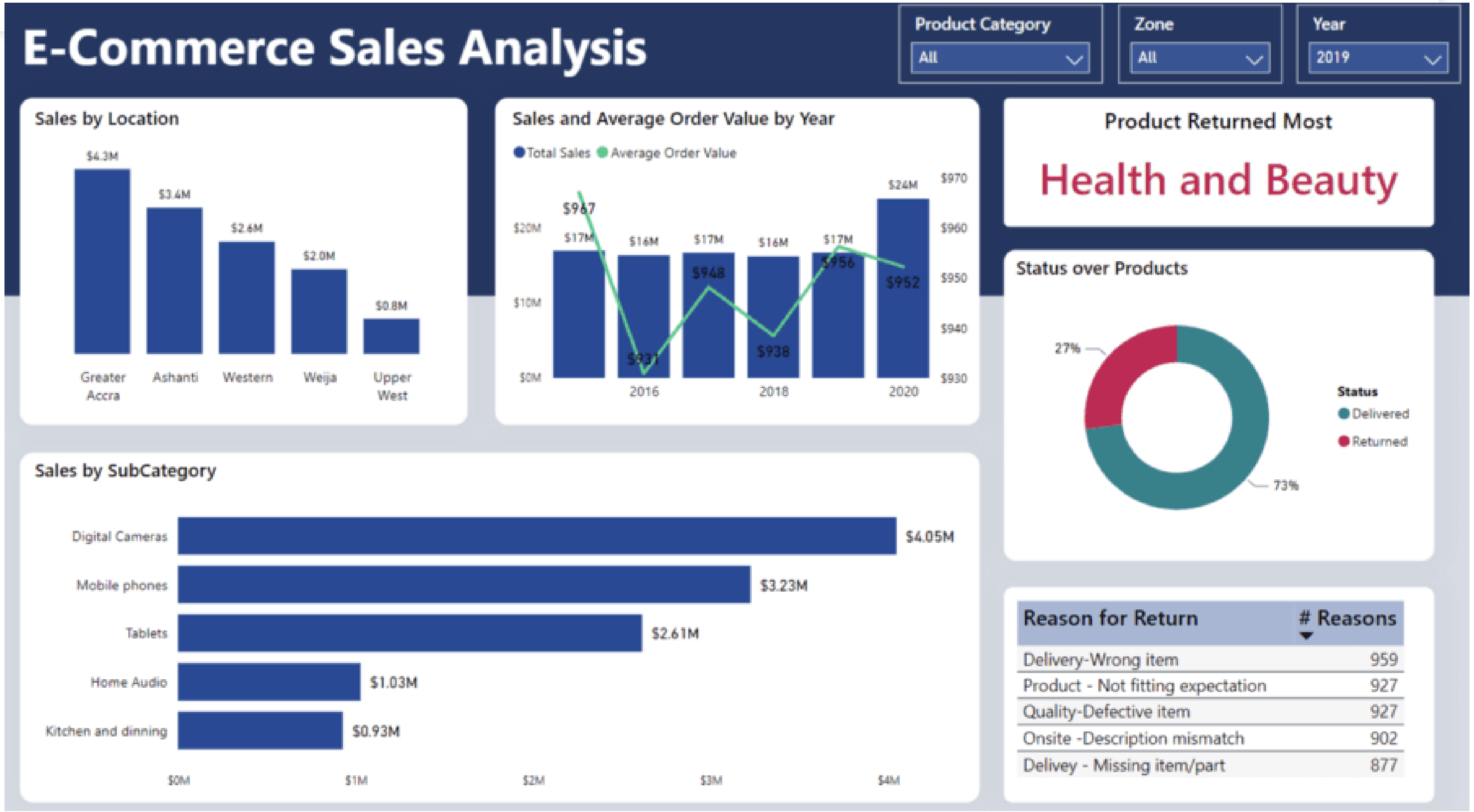
	Dashboard	Report
Purpose	Used for high-level monitoring, often in real-time or near-real-time, providing a consolidated view of business performance	Used for in-depth analysis and exploration of data to answer complex business questions.
Interactivity and Structure	Limited interactivity – single page view at a glance.	Greater interactivity – detailed multi-page analysis.
Updates	Designed for real-time or near real-time updates making monitoring of live data easy.	Focused on historical data and may require periodic updates enabled through PowerBI scheduled refresh capabilities.
Use case	Quick sharing of insights across organization	More suited for detailed analysis allowing for specific aspects of the data.

Dashboard

Monitoring e-Commerce Sales

A quick review of all KPIs.

A well-designed dashboard enable spotting problems quickly.



E-Commerce Sales Analysis

Product Category: All

Zone: All

Year: 2019

Product Returned Most

Health and Beauty

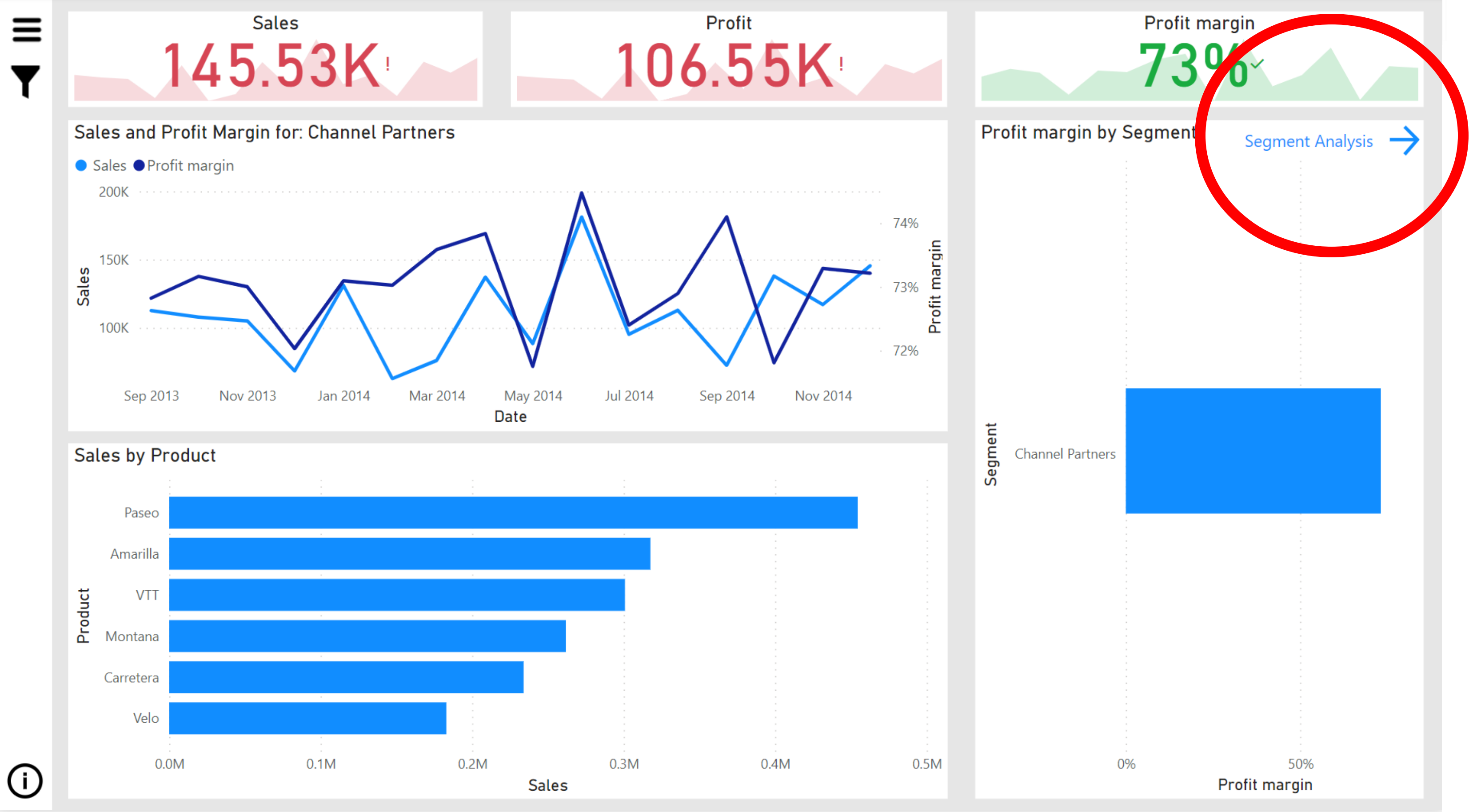
Reason for Return	# Reasons
Delivery-Wrong item	959
Product - Not fitting expectation	927
Quality-Defective item	927
Onsite -Description mismatch	902
Delivey - Missing item/part	877

Source: DataCamp

Report

Analyze impact of a recent promotional campaign on sales.

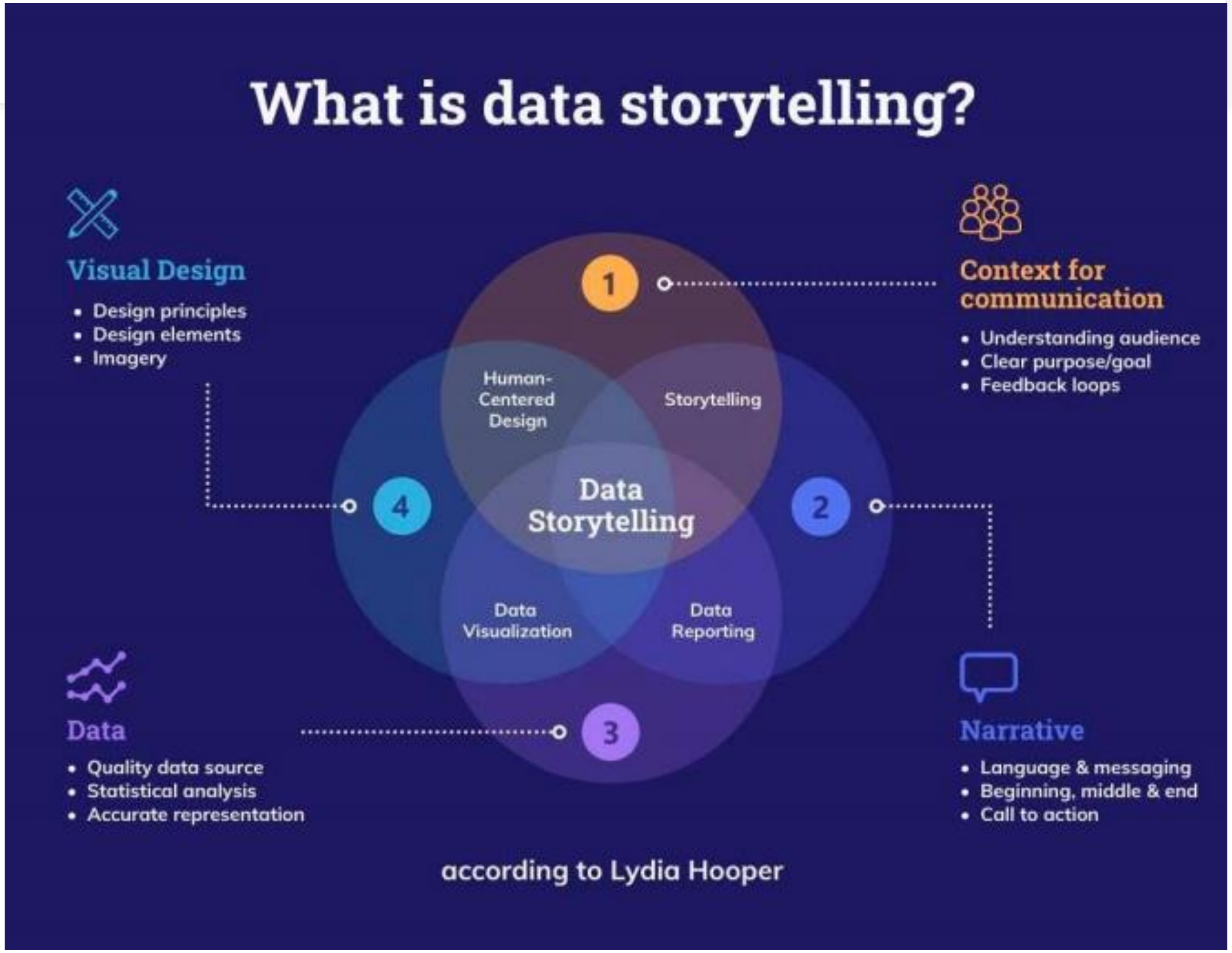
Review the sales by different dimensions.



Source: DataCamp

Data Story Telling

- Data storytelling is the concept of building a compelling narrative based on complex data and analytics that help tell your story and influence and inform a particular audience.



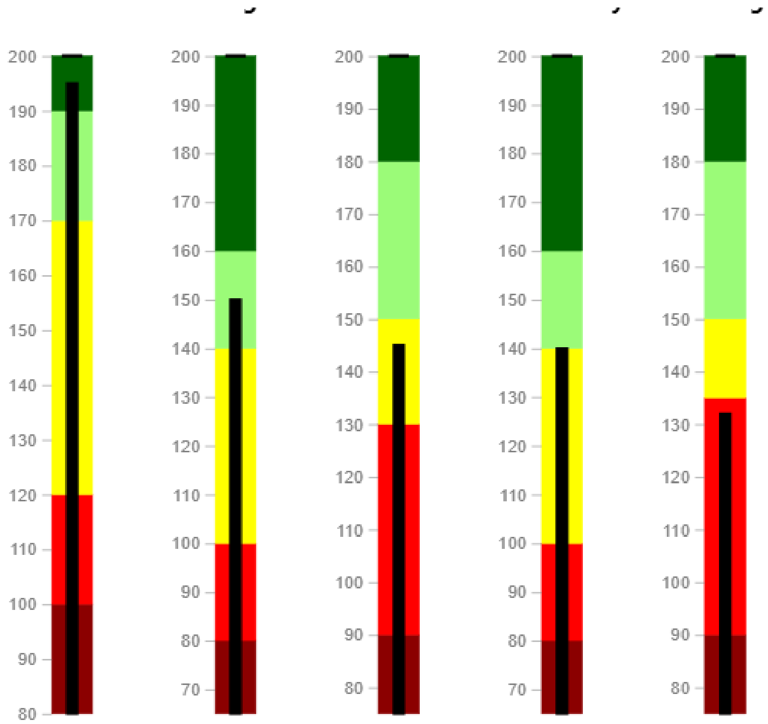
A good story is…

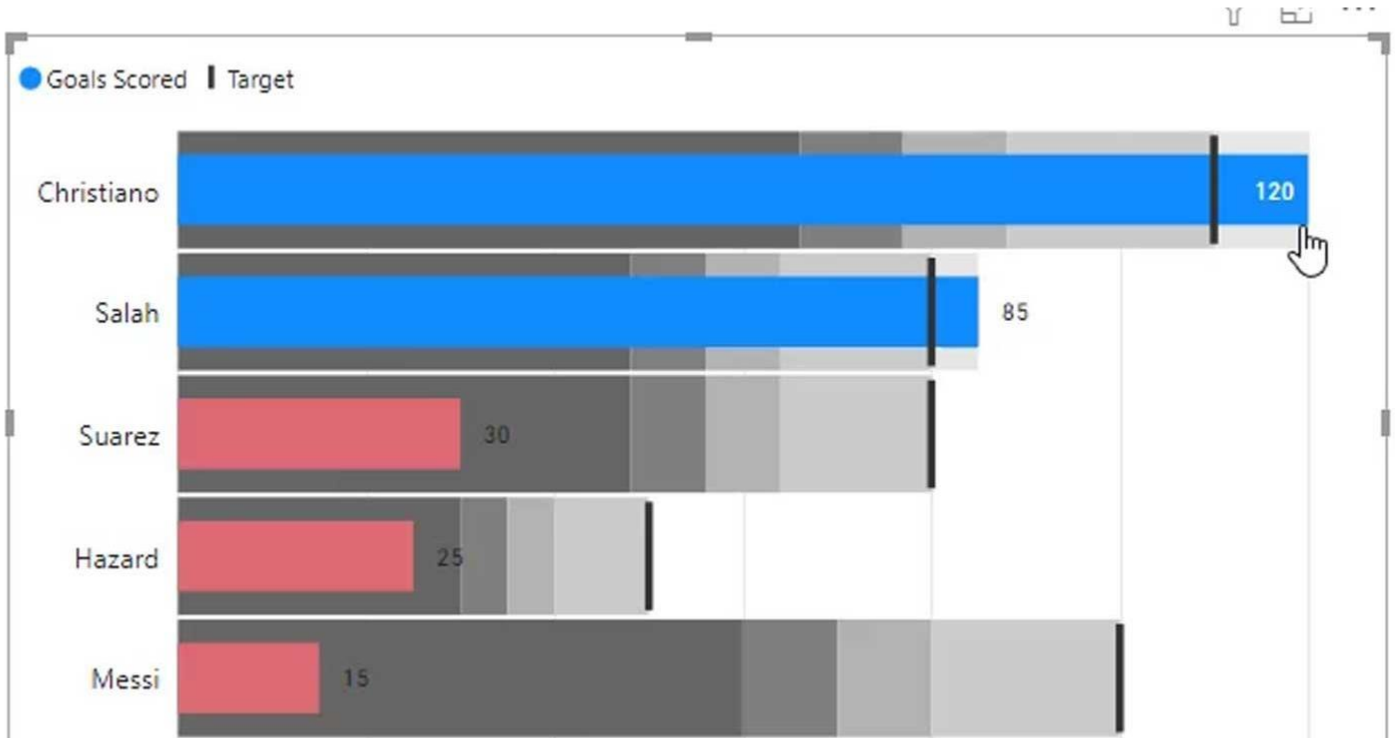
- **Relevant** to the audience , the business domain and specific problem being solved.
- Uses **good data** – collected form reputable sources that convey the true story.
- Forms a **clear narrative** – introduce the topic before diving in.
- Uses **Smart visuals** (charts/images/etc)– convey the intended purpose. Well labelled, legible, not misleading

Bullet Chart

Bullet chart

- A Power BI bullet chart is a more advanced type of bar chart and great for plotting data comparisons. Each bar is shown against multiple qualitative ranges and a target, making it easy to monitor progress. For example, you could use the chart to show how exam results are graded.





Adding some targets

- Let’s first understand what DAX is…

DAX
Data Analysis Expression

What is DAX in Power BI?

- Data Analysis Expressions (DAX) is a programming language that is used throughout Microsoft Power BI for creating calculated columns, measures, and custom tables.
- It is a collection of functions, operators, and constants that can be used in a formula, or expression, to calculate and return one or more values.
- You can use DAX to solve a number of calculations and data analysis problems, which can help you create new information from data that is already in your model.

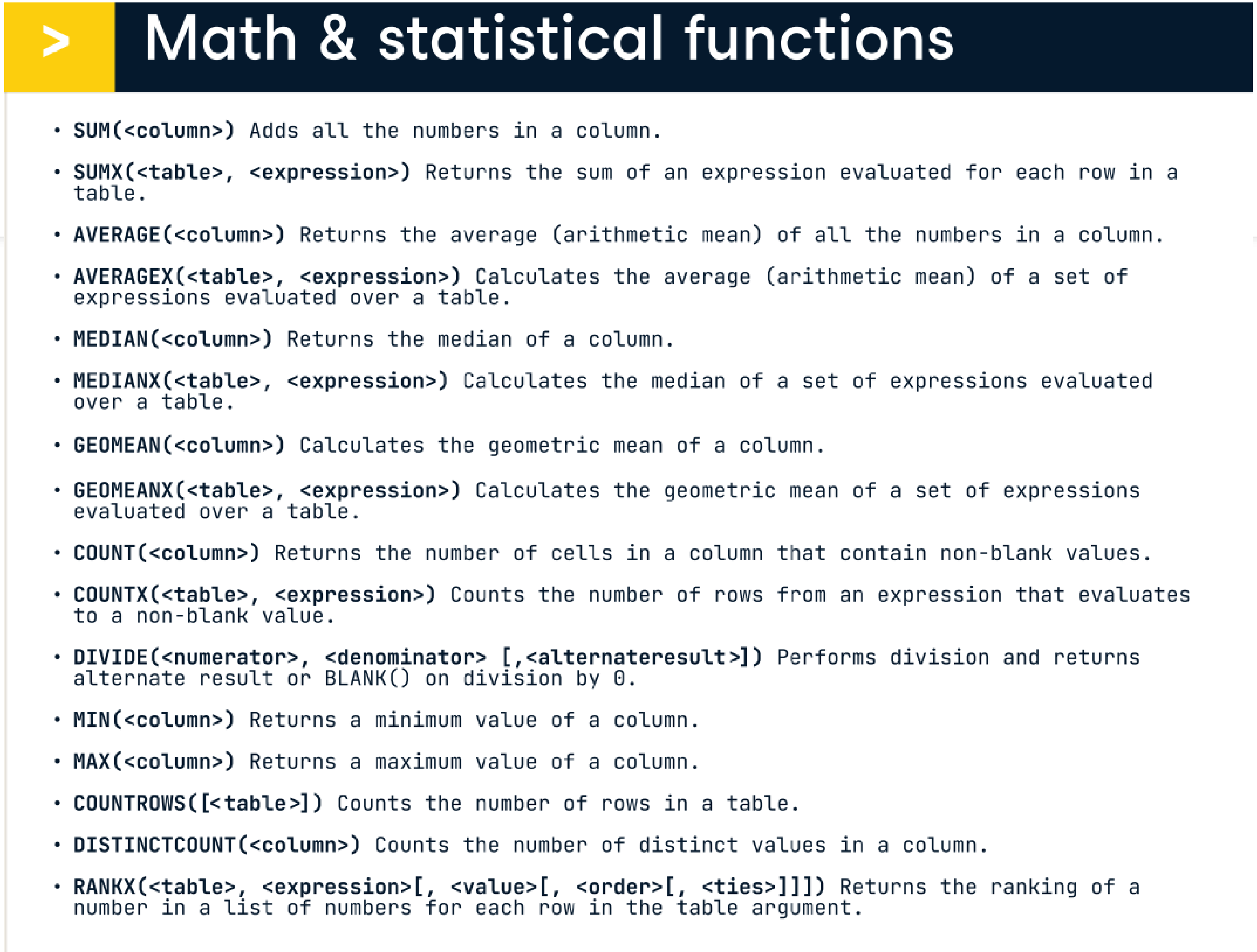
Calculated Columns vs Measures

Calculated Column

- Similar to other regular columns, calculated columns are the ones that come out as a result of **computations within two columns** of different data sets. They are ideal for row-wise calculations;
- In the calculated column type, the calculation takes place at the **row level** within a given table.

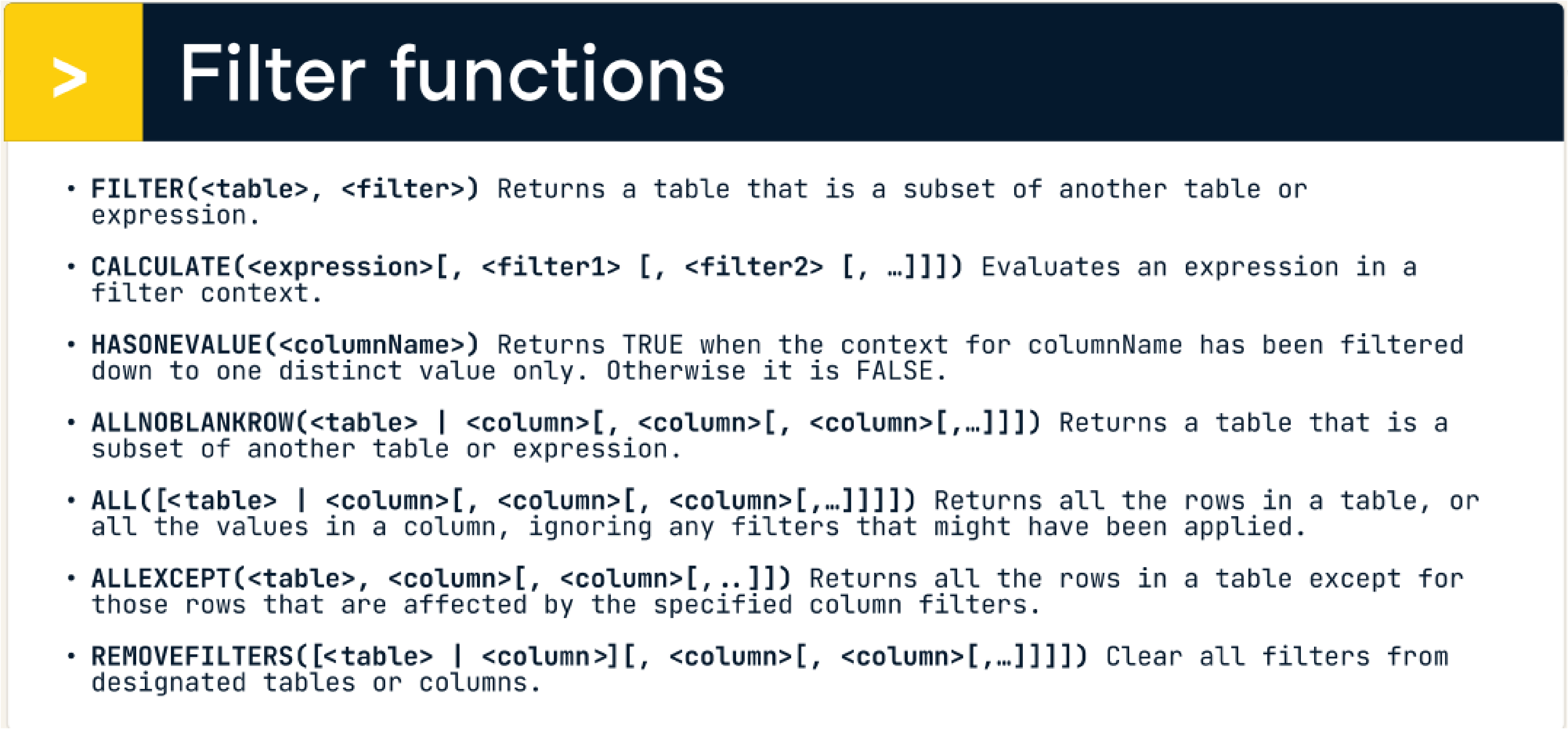
Calculated Measure

- Calculated measure are ideal for dynamic calculations as they do not acquire any physical space. **These are used when data in rows are grouped together for computations.**
- In the calculated measure type, the calculation is done at the **cell level in the entire report or the query.**



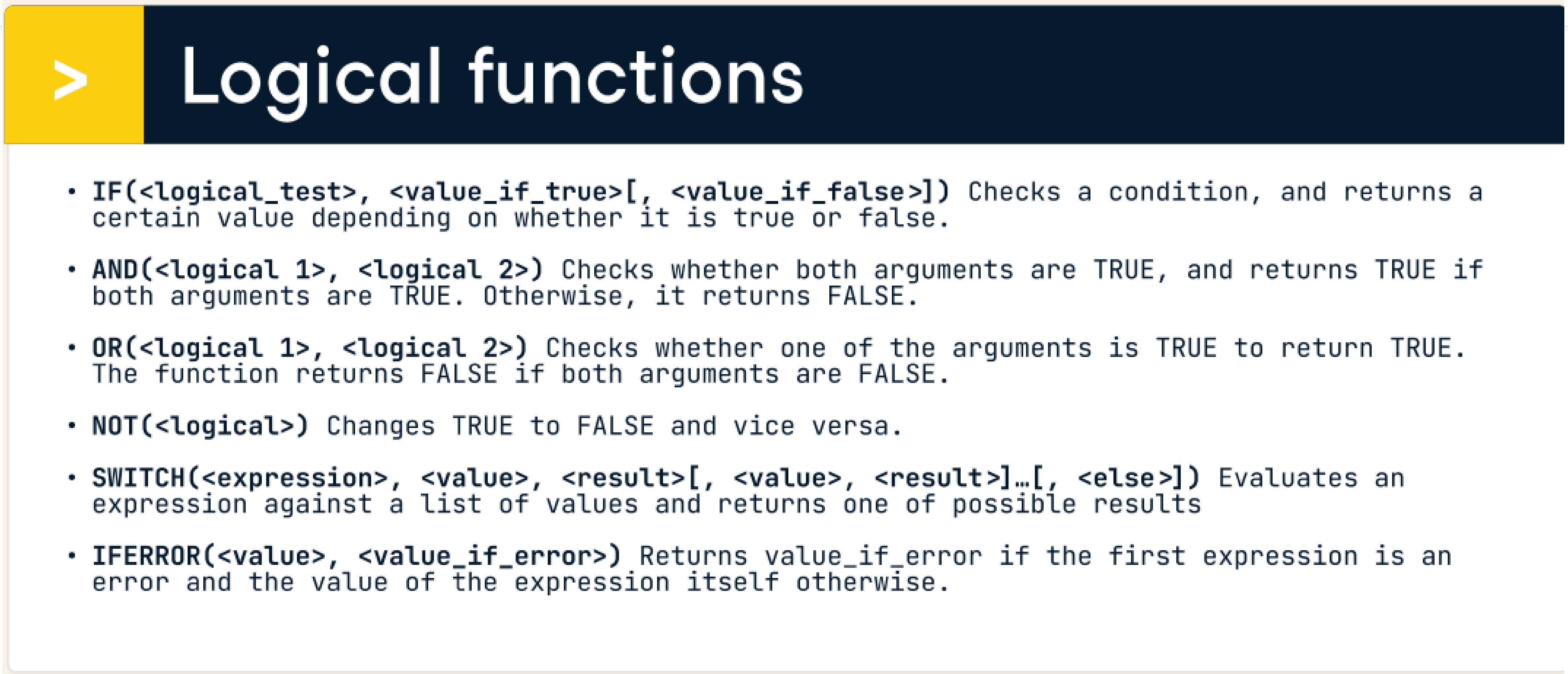
> Math & statistical functions

- **SUM(<column>)** Adds all the numbers in a column.
- **SUMX(<table>, <expression>)** Returns the sum of an expression evaluated for each row in a table.
- **AVERAGE(<column>)** Returns the average (arithmetic mean) of all the numbers in a column.
- **AVERAGEX(<table>, <expression>)** Calculates the average (arithmetic mean) of a set of expressions evaluated over a table.
- **MEDIAN(<column>)** Returns the median of a column.
- **MEDIANX(<table>, <expression>)** Calculates the median of a set of expressions evaluated over a table.
- **GEOMEAN(<column>)** Calculates the geometric mean of a column.
- **GEOMEANX(<table>, <expression>)** Calculates the geometric mean of a set of expressions evaluated over a table.
- **COUNT(<column>)** Returns the number of cells in a column that contain non-blank values.
- **COUNTX(<table>, <expression>)** Counts the number of rows from an expression that evaluates to a non-blank value.
- **DIVIDE(<numerator>, <denominator> [,<alternateresult>])** Performs division and returns alternate result or BLANK() on division by 0.
- **MIN(<column>)** Returns a minimum value of a column.
- **MAX(<column>)** Returns a maximum value of a column.
- **COUNTROWS([<table>])** Counts the number of rows in a table.
- **DISTINCTCOUNT(<column>)** Counts the number of distinct values in a column.
- **RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])** Returns the ranking of a number in a list of numbers for each row in the table argument.



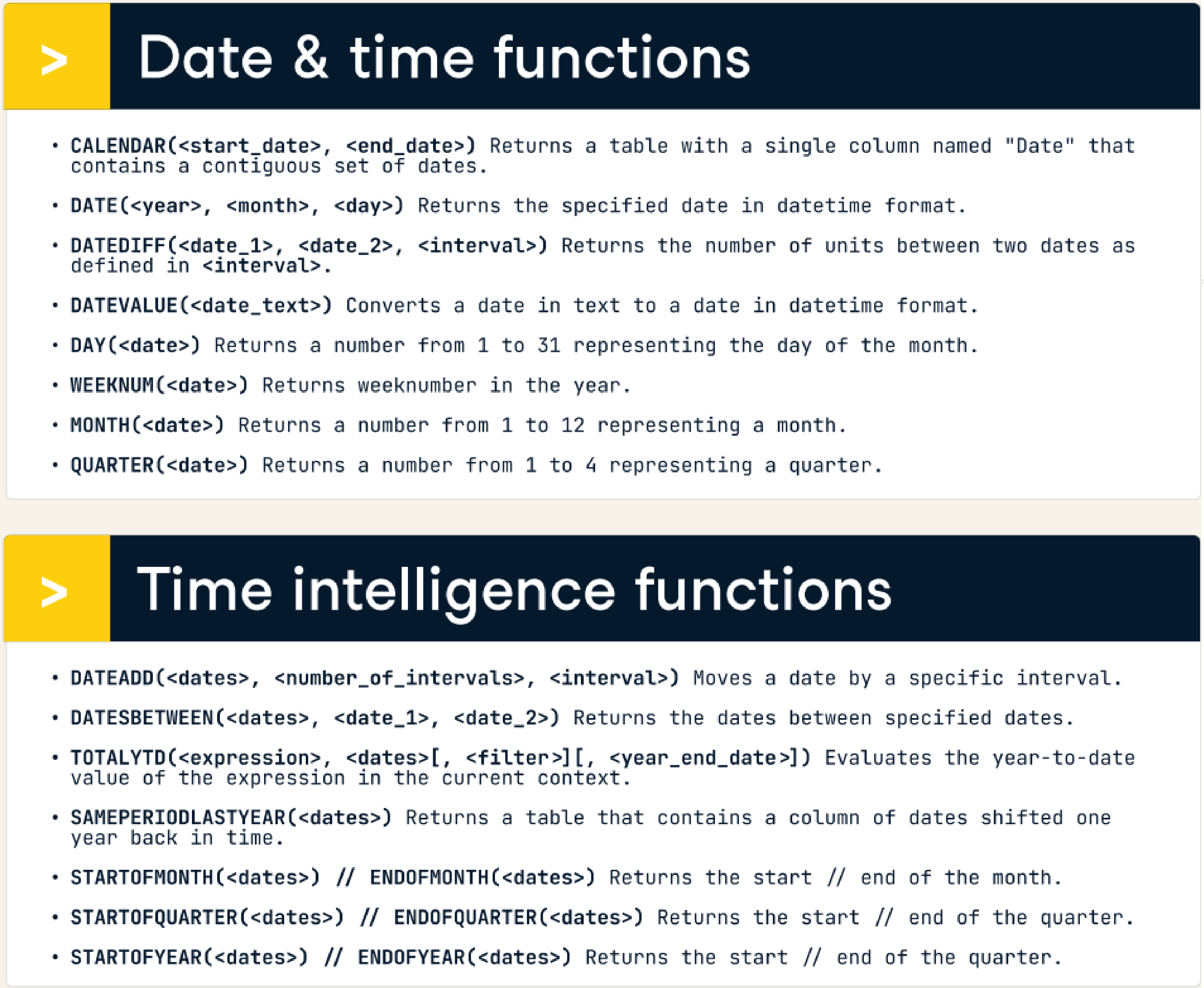
> Filter functions

- **FILTER(<table>, <filter>)** Returns a table that is a subset of another table or expression.
- **CALCULATE(<expression>[, <filter1> [, <filter2> [, …]]])** Evaluates an expression in a filter context.
- **HASONEVALUE(<columnName>)** Returns TRUE when the context for columnName has been filtered down to one distinct value only. Otherwise it is FALSE.
- **ALLNOBLANKROW(<table> | <column>[, <column>[, <column>[,…]]])** Returns a table that is a subset of another table or expression.
- **ALL([<table> | <column>[, <column>[, <column>[,…]]]])** Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied.
- **ALLEXCEPT(<table>, <column>[, <column>[,..]])** Returns all the rows in a table except for those rows that are affected by the specified column filters.
- **REMOVEFILTERS([<table> | <column>][, <column>[, <column>[,…]]]])** Clear all filters from designated tables or columns.



Logical functions

- **IF(<logical_test>, <value_if_true>[, <value_if_false>])** Checks a condition, and returns a certain value depending on whether it is true or false.
- **AND(<logical 1>, <logical 2>)** Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE. Otherwise, it returns FALSE.
- **OR(<logical 1>, <logical 2>)** Checks whether one of the arguments is TRUE to return TRUE. The function returns FALSE if both arguments are FALSE.
- **NOT(<logical>)** Changes TRUE to FALSE and vice versa.
- **SWITCH(<expression>, <value>, <result>[, <value>, <result>]…[, <else>])** Evaluates an expression against a list of values and returns one of possible results
- **IFERROR(<value>, <value_if_error>)** Returns value_if_error if the first expression is an error and the value of the expression itself otherwise.

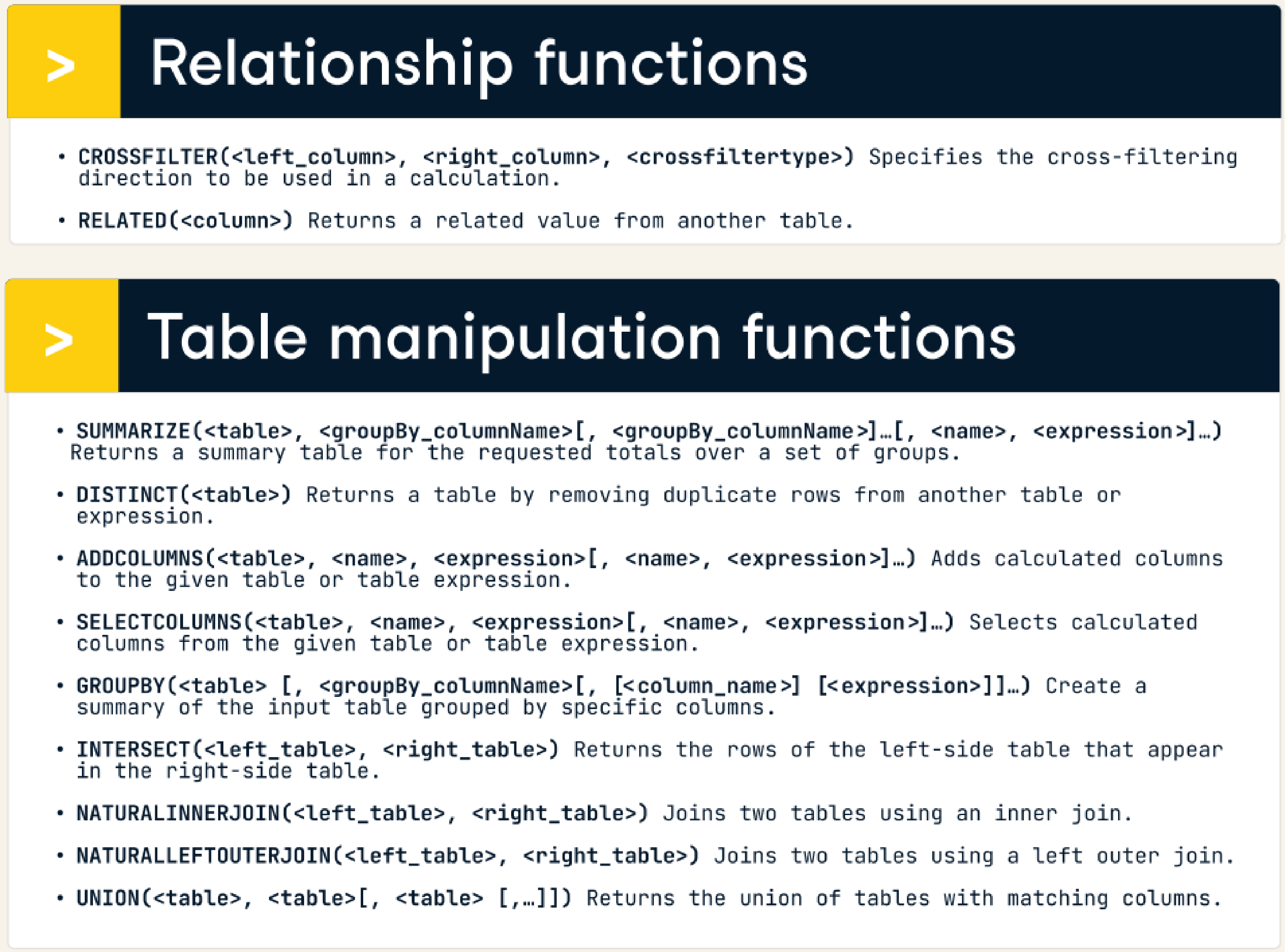


Date & time functions

- `CALENDAR(<start_date>, <end_date>)` Returns a table with a single column named "Date" that contains a contiguous set of dates.
- `DATE(<year>, <month>, <day>)` Returns the specified date in datetime format.
- `DATEDIFF(<date_1>, <date_2>, <interval>)` Returns the number of units between two dates as defined in <interval>.
- `DATEVALUE(<date_text>)` Converts a date in text to a date in datetime format.
- `DAY(<date>)` Returns a number from 1 to 31 representing the day of the month.
- `WEEKNUM(<date>)` Returns weeknumber in the year.
- `MONTH(<date>)` Returns a number from 1 to 12 representing a month.
- `QUARTER(<date>)` Returns a number from 1 to 4 representing a quarter.

Time intelligence functions

- `DATEADD(<dates>, <number_of_intervals>, <interval>)` Moves a date by a specific interval.
- `DATESBETWEEN(<dates>, <date_1>, <date_2>)` Returns the dates between specified dates.
- `TOTALYTD(<expression>, <dates>[, <filter>][, <year_end_date>])` Evaluates the year-to-date value of the expression in the current context.
- `SAMEPERIODLASTYEAR(<dates>)` Returns a table that contains a column of dates shifted one year back in time.
- `STARTOFMONTH(<dates>) // ENDOFMONTH(<dates>)` Returns the start // end of the month.
- `STARTOFQUARTER(<dates>) // ENDOFQUARTER(<dates>)` Returns the start // end of the quarter.
- `STARTOFYEAR(<dates>) // ENDOFYEAR(<dates>)` Returns the start // end of the quarter.

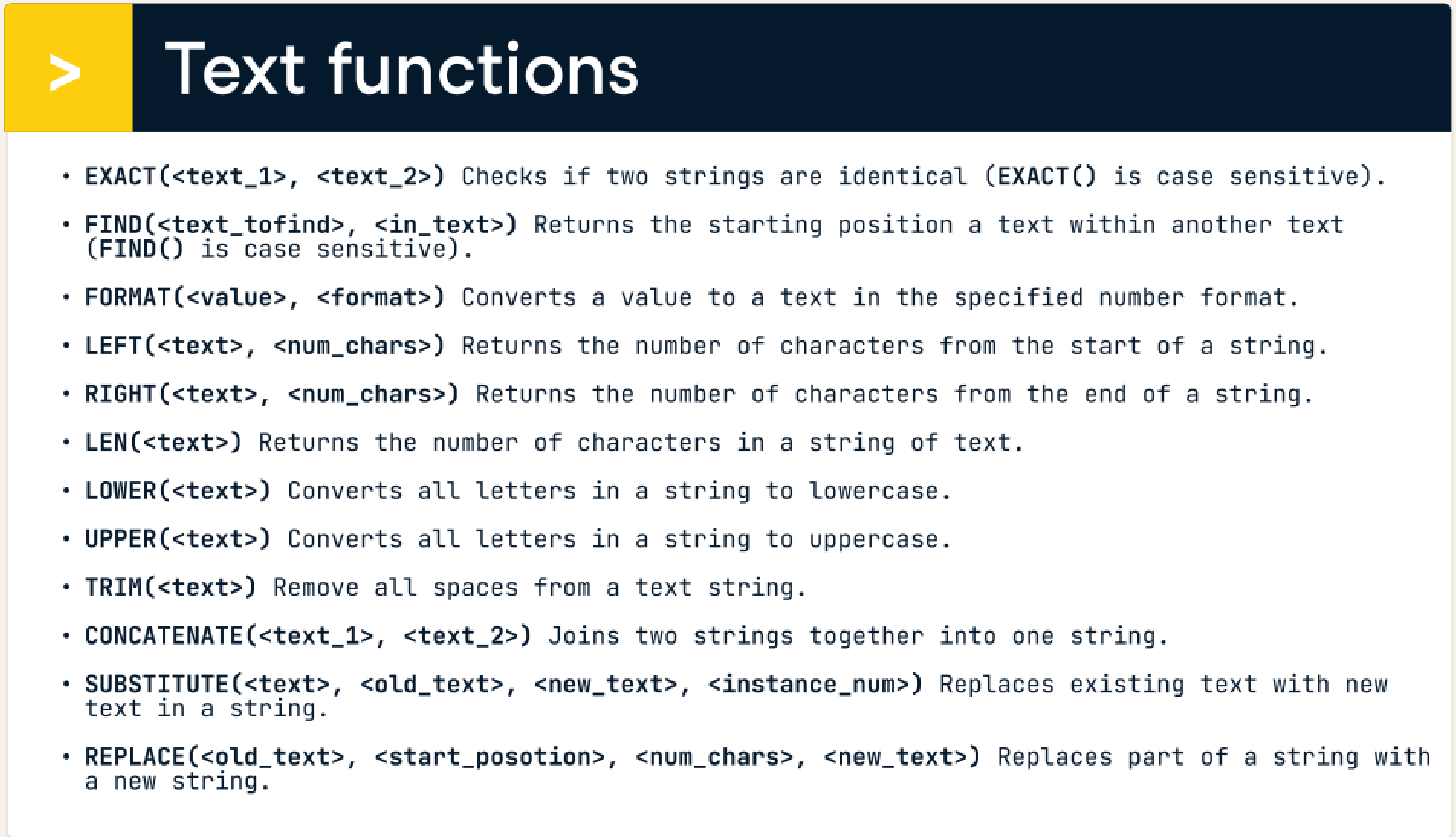


> Relationship functions

- CROSSFILTER(<left_column>, <right_column>, <crossfiltertype>) Specifies the cross-filtering direction to be used in a calculation.
- RELATED(<column>) Returns a related value from another table.

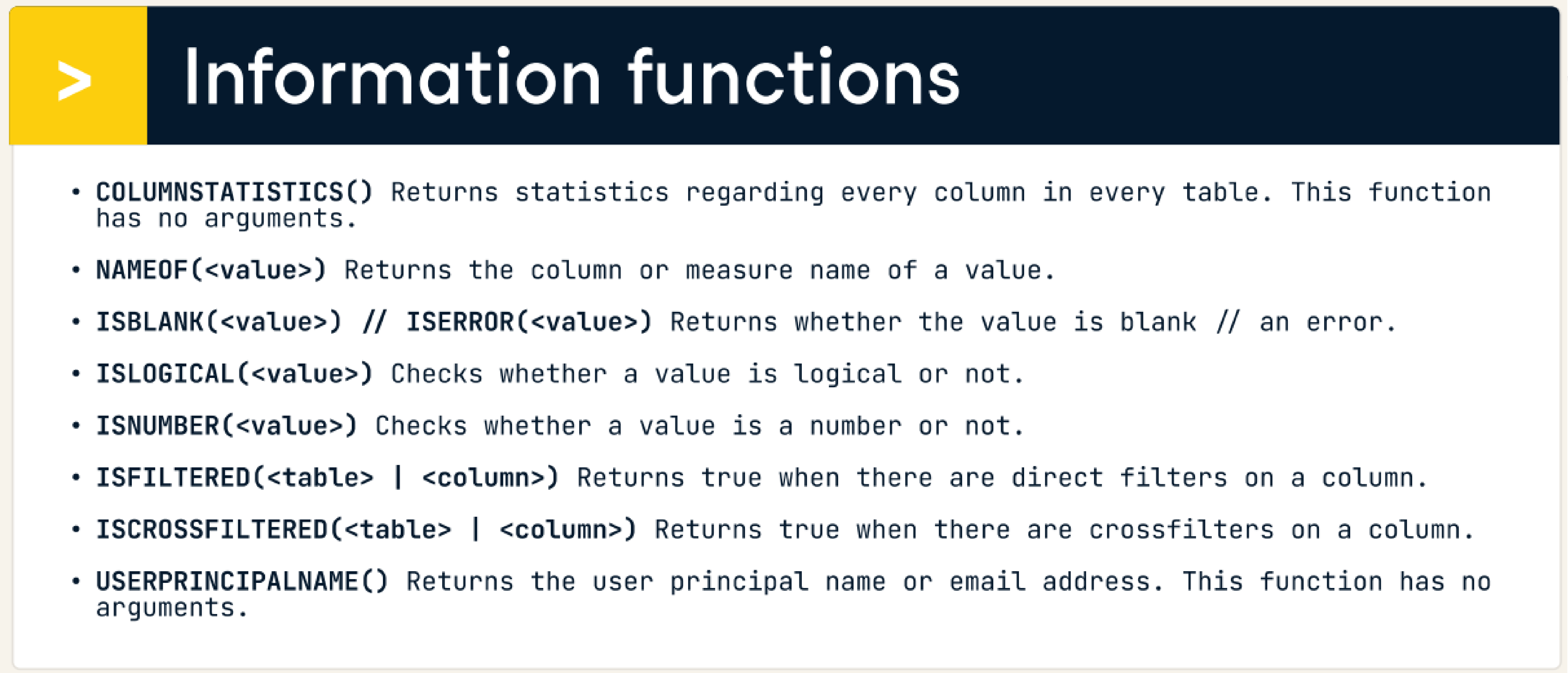
> Table manipulation functions

- SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]…[, <name>, <expression>]…) Returns a summary table for the requested totals over a set of groups.
- DISTINCT(<table>) Returns a table by removing duplicate rows from another table or expression.
- ADDCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]…) Adds calculated columns to the given table or table expression.
- SELECTCOLUMNS(<table>, <name>, <expression>[, <name>, <expression>]…) Selects calculated columns from the given table or table expression.
- GROUPBY(<table> [, <groupBy_columnName>[, [<column_name>] [<expression>]]…) Create a summary of the input table grouped by specific columns.
- INTERSECT(<left_table>, <right_table>) Returns the rows of the left-side table that appear in the right-side table.
- NATURALINNERJOIN(<left_table>, <right_table>) Joins two tables using an inner join.
- NATURALLEFTOUTERJOIN(<left_table>, <right_table>) Joins two tables using a left outer join.
- UNION(<table>, <table>[, <table> [,…]]) Returns the union of tables with matching columns.



> Text functions

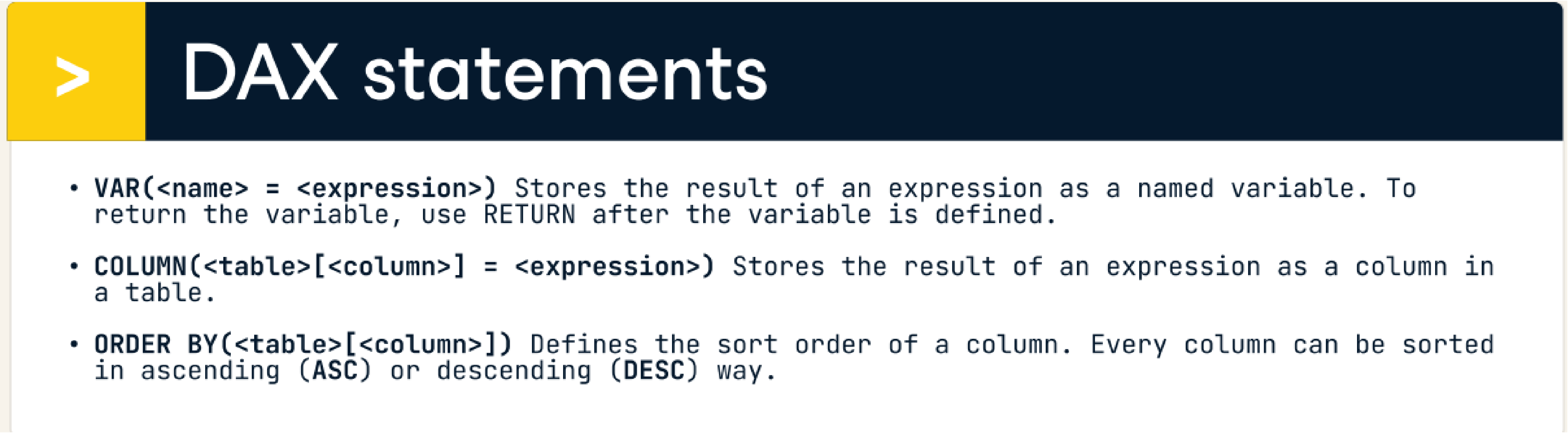
- EXACT(<text_1>, <text_2>) Checks if two strings are identical (EXACT() is case sensitive).
- FIND(<text_tofind>, <in_text>) Returns the starting position a text within another text (FIND() is case sensitive).
- FORMAT(<value>, <format>) Converts a value to a text in the specified number format.
- LEFT(<text>, <num_chars>) Returns the number of characters from the start of a string.
- RIGHT(<text>, <num_chars>) Returns the number of characters from the end of a string.
- LEN(<text>) Returns the number of characters in a string of text.
- LOWER(<text>) Converts all letters in a string to lowercase.
- UPPER(<text>) Converts all letters in a string to uppercase.
- TRIM(<text>) Remove all spaces from a text string.
- CONCATENATE(<text_1>, <text_2>) Joins two strings together into one string.
- SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>) Replaces existing text with new text in a string.
- REPLACE(<old_text>, <start_posotion>, <num_chars>, <new_text>) Replaces part of a string with a new string.



> Information functions

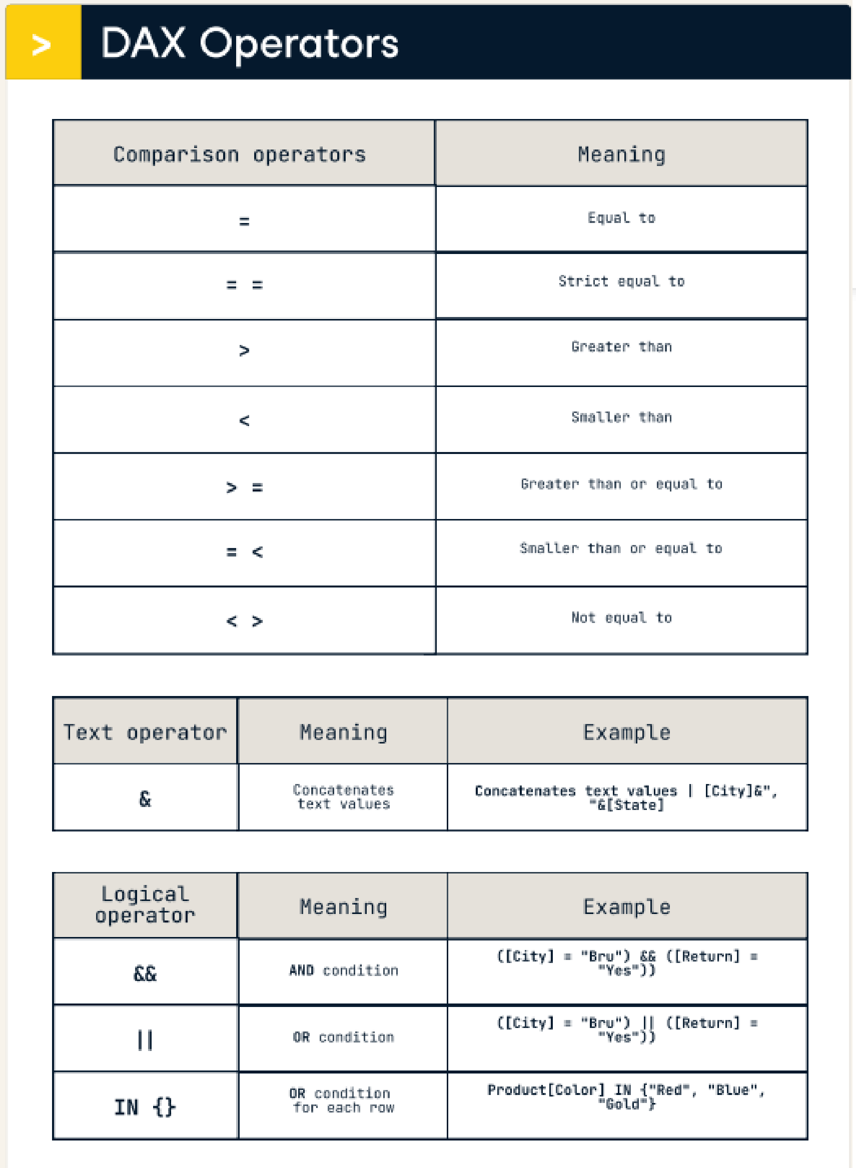
- **COLUMNSTATISTICS()** Returns statistics regarding every column in every table. This function has no arguments.
- **NAMEOF(<value>)** Returns the column or measure name of a value.
- **ISBLANK(<value>) // ISERROR(<value>)** Returns whether the value is blank // an error.
- **ISLOGICAL(<value>)** Checks whether a value is logical or not.
- **ISNUMBER(<value>)** Checks whether a value is a number or not.
- **ISFILTERED(<table> | <column>)** Returns true when there are direct filters on a column.
- **ISCROSSFILTERED(<table> | <column>)** Returns true when there are crossfilters on a column.
- **USERPRINCIPALNAME()** Returns the user principal name or email address. This function has no arguments.

Advanced DAX



> DAX statements

- **VAR(<name> = <expression>)** Stores the result of an expression as a named variable. To return the variable, use RETURN after the variable is defined.
- **COLUMN(<table>[<column>] = <expression>)** Stores the result of an expression as a column in a table.
- **ORDER BY(<table>[<column>])** Defines the sort order of a column. Every column can be sorted in ascending (**ASC**) or descending (**DESC**) way.



> DAX Operators

Comparison operators	Meaning
=	Equal to
= =	Strict equal to
>	Greater than
<	Smaller than
> =	Greater than or equal to
= <	Smaller than or equal to
< >	Not equal to

Text operator	Meaning	Example
&	Concatenates text values	Concatenates text values | [City]&", "&[State]

Logical operator	Meaning	Example
&&	AND condition	([City] = "Bru") && ([Return] = "Yes"))
||	OR condition	([City] = "Bru") || ([Return] = "Yes"))
IN {}	OR condition for each row	Product[Color] IN {"Red", "Blue", "Gold"}

Applying DAX

Insert New Columns by Calculations

- Compute Delay in Power BI using DAX
- DATEDIFF: returns the difference between 2 dates (both as column references in date formats) in days, months, quarters, years, etc.
- Delay = Delivery Date - Scheduled Date in DAYS

- Delay.DAX =
  ```
  DATEDIFF('Table 1 (SC dataset - Power BI)'[Scheduled Delivery Date], 'Table 1 (SC dataset - Power BI)'[Delivery Date],DAY)
  ```

Insert New Columns by IF

- Compute Delay Category using DAX
- Use the IF Function to output 3 categories → On-time, Delayed, Early

-
  ```
  Delay.Cat = IF('Table 1 (SC dataset - Power
  BI)'[Delay.DAX]=0,"On-time",IF('Table 1 (SC dataset -
  Power BI)'[Delay.DAX]>0,"Delayed","Early"))
  ```

MAX Delay

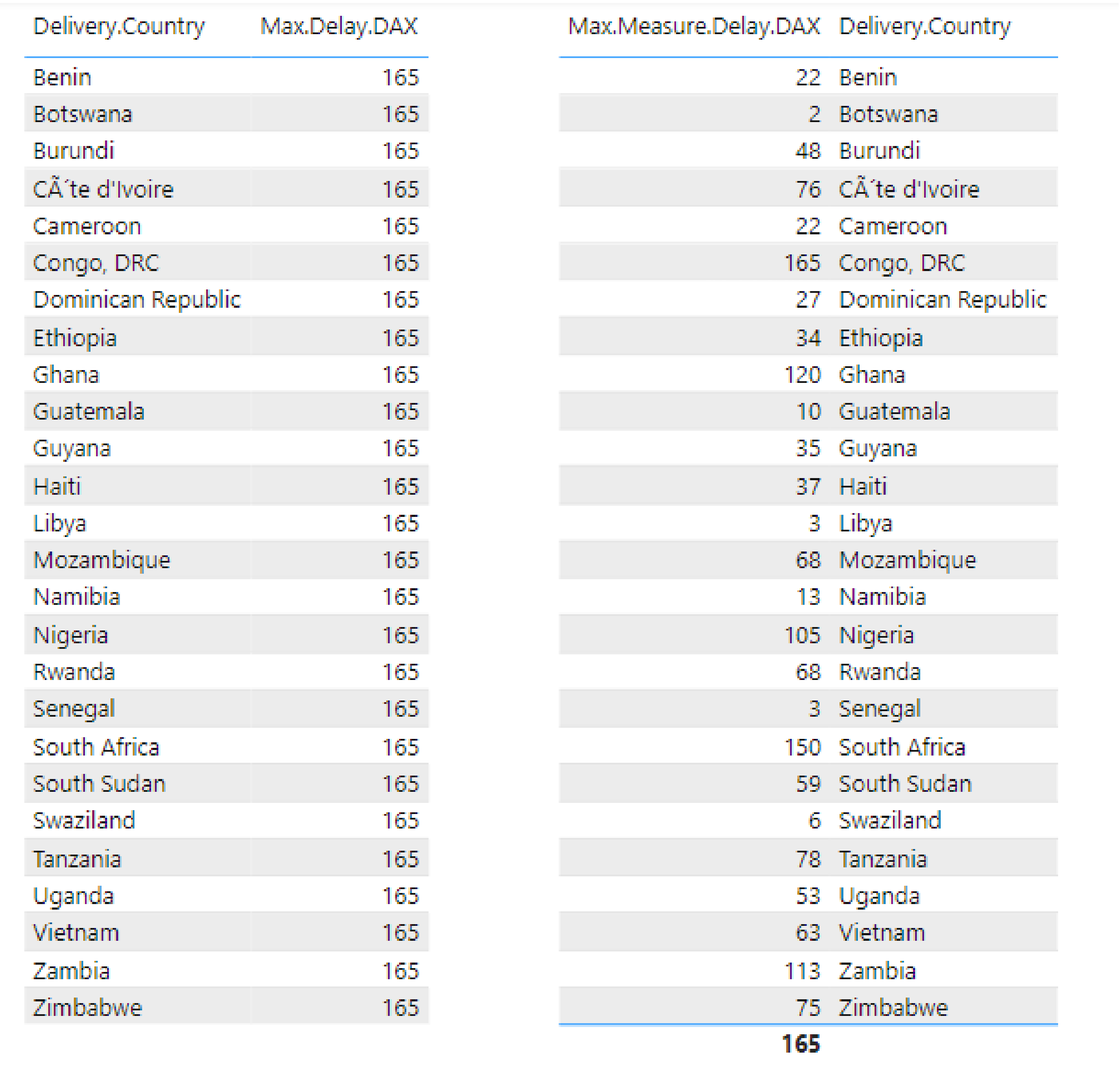
Column

- Max.Delay.DAX = MAX('Table 1 (SC dataset - Power BI)'[Delivery.Delay])

Measure

```
Max.Measure.Delay.DAX = MAX('Table 1 (SC dataset - Power
BI)'[Delivery.Delay])
```

Create Tables for each with Manufacturing Country



Delivery.Country	Max.Delay.DAX
Benin	165
Botswana	165
Burundi	165
CÃ´te d'Ivoire	165
Cameroon	165
Congo, DRC	165
Dominican Republic	165
Ethiopia	165
Ghana	165
Guatemala	165
Guyana	165
Haiti	165
Libya	165
Mozambique	165
Namibia	165
Nigeria	165
Rwanda	165
Senegal	165
South Africa	165
South Sudan	165
Swaziland	165
Tanzania	165
Uganda	165
Vietnam	165
Zambia	165
Zimbabwe	165

Max.Measure.Delay.DAX	Delivery.Country
22	Benin
2	Botswana
48	Burundi
76	CÃ´te d'Ivoire
22	Cameroon
165	Congo, DRC
27	Dominican Republic
34	Ethiopia
120	Ghana
10	Guatemala
35	Guyana
37	Haiti
3	Libya
68	Mozambique
13	Namibia
105	Nigeria
68	Rwanda
3	Senegal
150	South Africa
59	South Sudan
6	Swaziland
78	Tanzania
53	Uganda
63	Vietnam
113	Zambia
75	Zimbabwe
165	