BIG DATA ANALYSIS AND VISUALIZATION USING POWER BI AND TABLEAU

Data Transformation in Power BI

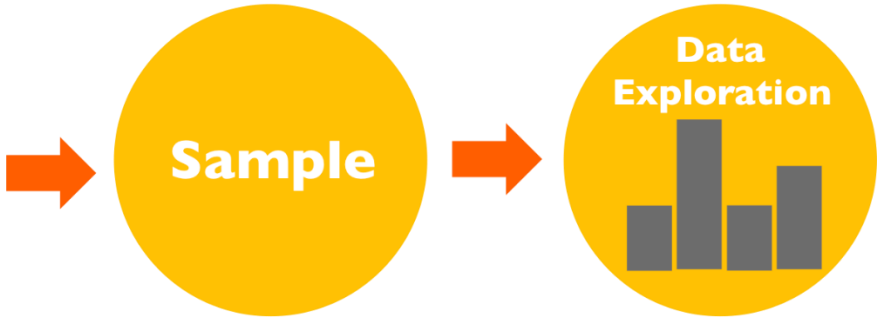
Peter Lo

Data Cleaning

- Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted.
- This data is usually not necessary or helpful when it comes to analyzing data because it may hinder the process or provide inaccurate results.

Bad Data

Country	UN R/P	UN R/P	World Bank	WB Gini	CIA R/P
Afghanistan	4.1;4.2		27.8	2008	
Albania	7.2	4.8	29	2012	7.2
Algeria	9.6	6.1	35.3		1995,9.6



Combine Queries

- There are two primary ways of combining queries:
  - Merging
  - Appending

ID	Title	Timing	Capacity
1	Math	Q1 2017	15
2	Physics	Q1 2017	5
3	English	Q1 2017	25

ID	Firstname	Lastname	Course
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math
10	Shaun	Mitchel	English
11	Mary	Stuart	English

	ID	Title	Timing	Capacity	NewColumn.ID	NewColumn.Firstname	NewColumn.Lastname	NewColumn.Course
1	1	Math	Q1 2017	15	1	Reza	Rad	Math
2	1	Math	Q1 2017	15	2	Mike	Woody	Math
3	1	Math	Q1 2017	15	3	Jimmy	Mitchel	Math
4	3	English	Q1 2017	25	10	Shaun	Mitchel	English
5	3	English	Q1 2017	25	11	Mary	Stuart	English
6	2	Physics	Q1 2017	5	null	null	null	null

Merge Queries

- When you have one or more columns that you'd like to add to another query, you merge the queries.

ID	Sex	HH
1	M	1
2	F	1
3	M	1
4	M	2
5	F	3
6	F	2

HH	Income
1	22.345
2	25.678
3	12.987
4	45.678
5	18.323
6	33.678

↘ ↙

ID	SEX	HH	Income
1	M	1	22.345
2	F	1	22.345
3	M	1	22.345
4	M	2	25.678
5	F	3	18.323
6	F	2	25.678

Append Queries

- When you have additional rows of data that you'd like to add to an existing query, you append the query.

ID	Firstname	Lastname	Course
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math

ID	Firstname	Lastname	Course
10	Shaun	Mitchel	English
11	Mary	Stuart	English

ID	Firstname	Lastname	Course
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math
10	Shaun	Mitchel	English
11	Mary	Stuart	English

Introduction to DAX

- DAX stands for **Data Analysis eXpressions**, and it is the formula language used throughout Power BI
- DAX is a functional language, which means the full executed code is contained inside a function.
- There are two primary calculations you can create using DAX:
  - Calculated Columns
  - Calculated Measures

Calculated Columns

- Calculated columns are useful when you want to slice or filter on the value, or if you want a calculation for every row in your table.
- The required elements for a calculated column are:
  - A new column name
  - At least one function or expression

Calculated Measures

- Use a calculated measure when you are calculating percentages or ratios, or you need complex aggregations.
- The required elements for a calculated measure are the same as they are for a calculated column:
  - A new measure name
  - At least one function or expression

DAX Function

- With DAX, there are many functions available to shape, form, or otherwise analyze your data.
- These functions can be grouped into a handful of categories:
  - Aggregation functions
  - Counting functions
  - Logical functions
  - Information functions
  - Text functions
  - Date functions

Aggregation Functions

- DAX has a number of aggregation functions, including the following commonly used functions:
  - SUM
  - AVERAGE
  - MIN
  - MAX

These functions work only on numeric columns, and generally can aggregate only one column at a time. However, special aggregation functions that end in X, such as SUMX, can work on multiple columns. These functions iterate through the table, and evaluate the expression for each row

Counting Functions

- Often-used counting functions in DAX include the following:
  - COUNT
  - COUNTA
  - COUNTBLANK
  - COUNTROWS
  - DISTINCTCOUNT

These functions count different elements, such as distinct values, non-empty values, and table rows

Logical Functions

- The collection of logical functions in DAX include:
  - AND
  - OR
  - NOT
  - IF
  - IFERROR

These special functions can also be expressed with operators. For example, AND can be typed as && in your DAX formula. You can use operators when you need more than two conditions in your formula, but otherwise, it's best practice use the function name itself for readability of your DAX code

Information Functions

- Information functions in DAX include:
  - ISBLANK
  - ISNUMBER
  - ISTEXT
  - ISNONTEXT
  - ISERROR

While these functions can be situationally useful, there is value in knowing the data type of your columns ahead of time, rather than depending on these functions to provide the data type. DAX uses the MAX and MIN functions to both aggregate values, and to compare values

Text Functions

- The text functions in DAX include the following:
  - CONCATENTATE
  - REPLACE
  - SEARCH
  - UPPER
  - FIXED

These text work very similarly to the Excel functions that have the same name, so if you're familiar with how Excel handles text functions, you're already a step ahead.

Date Functions

- DAX includes the following Date functions:
  - DATE
  - HOUR
  - NOW
  - EOMONTH
  - WEEKDAY

While these functions are useful to calculate and extract information from date values, they do not apply to time intelligence, which uses a date table

Relational Functions

- DAX has relational functions that enable you to interact with tables that have established relationships.
- The RELATED function works on many-to-one relationships, while RELATEDTABLE is for one-to-many relationships.

You can use relational functions to build expressions that include values across multiple tables. DAX will return a result with these functions, regardless of the length of the chain of the relationship

DAX Table Functions

- DAX has a rich set of table functions, including the following:
  - FILTER
  - ALL
  - VALUES
  - DISTINCT

These functions return a full table rather, rather than a value. Typically you'll use the results of a table function in further analysis as part of a greater expression, rather than using that returned table a final value. It's important to note that When you use a table function, the results inherit the relationships of their columns

Variables

- Using variables are an extremely powerful part of a DAX expression.
- You can define a variable anywhere in a DAX expression, using the following syntax:
  - VARNAME = RETURNEDVALUE

```
VAR
    TotalQuantity = SUM ( Sales[Quantity] )

RETURN

    IF (
        TotalQuantity > 1000,
        TotalQuantity * 0.95,
        TotalQuantity * 1.25
    )
```