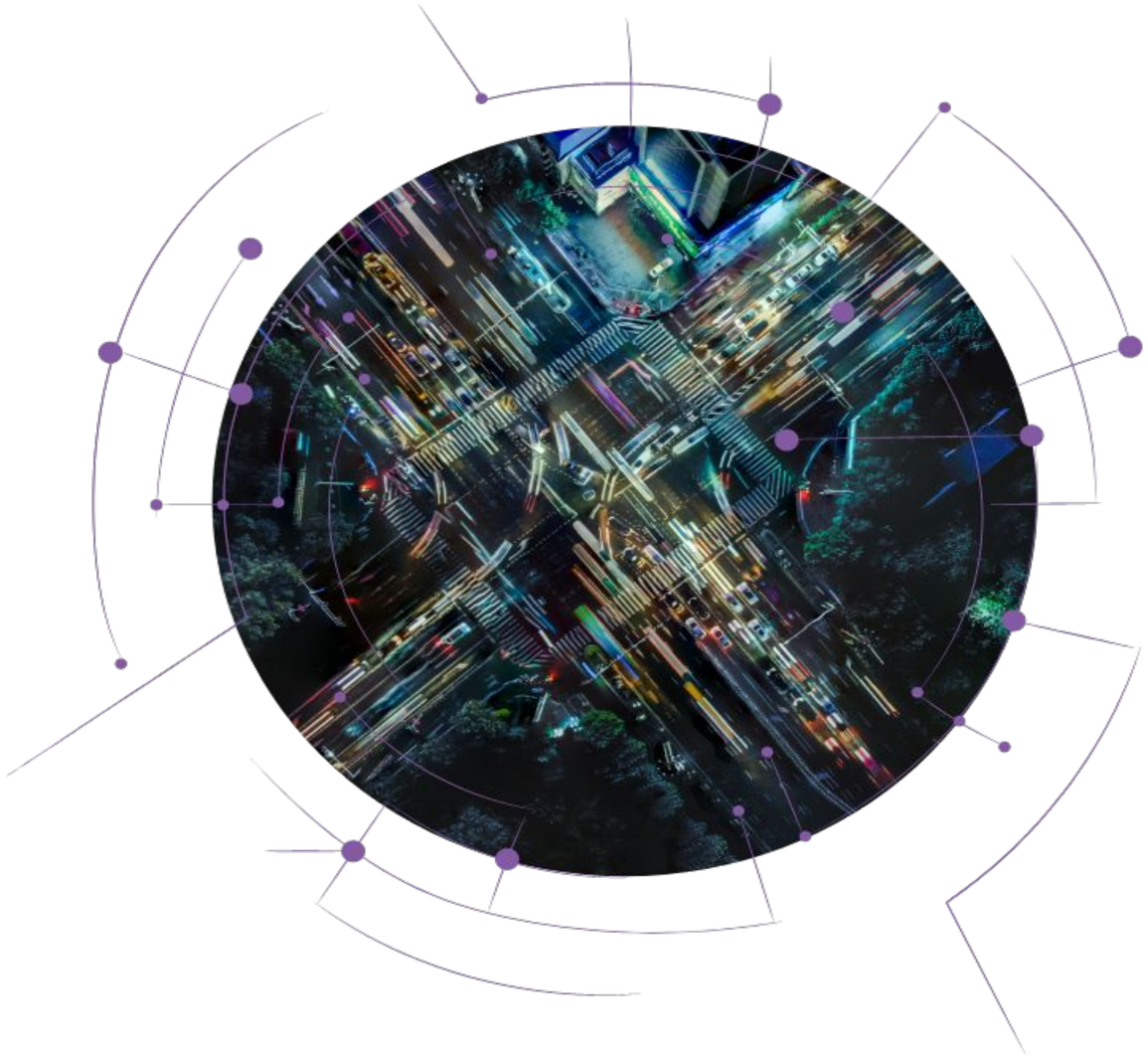
DATASOCIETY:

Introduction to Tableau

Part 9



Warm-up: top LOD functions

- In the last module, we discussed a few different types of function, including **LOD (level of detail) functions.**

- Skim through the following blog post:

Top 15 Tableau LOD Expressions (Practical Examples) [link]

- **How might some of these functions apply to your Capstone data or data you may encounter in future projects?**

Module completion checklist

Objective	Complete
Implement string calculations on a given dataset	
Implement date calculations on a given dataset	
Implement type calculations on a given dataset	
Implement logic calculations on given dataset	

Recap: function types

- Just like SQL functions, Tableau functions are classified into several types.

- These include:
  - Number Functions
  - String Functions
  - Date Functions
  - Type Conversion
  - Logical Functions
  - Aggregate Functions
  - Pass-Through Functions (RAWSQL)
  - User Functions
  - Table Calculation Functions
  - Spatial Functions
  - Additional Functions

- To see Tableau functions separated by type, visit **this page [link]**

String functions

- **String functions** allow you to manipulate string data (i.e. text data).

- Using string functions, you can perform a lot of common operations:
  - Capture part of a field.
  - Recast part of a string as an int.
  - Clean messy data.
  - Remove extraneous fields.

- Notable functions include **MID**, **REPLACE**, **and TRIM**, which we will practice using on the **country data** today.

String function: MID

- **MID**
  - Returns the string starting at a particular **position.**
  - If the optional argument **length** is added, the returned string includes only that number of characters.

- Example:

```
MID("Calculation", 2) =
"alculation"
MID("Calculation", 2, 5) ="alcul"
```

String function: REPLACE

- **REPLACE**
  - Searches a string for a substring and replaces it with a replacement substring.

- Example:

```
REPLACE("Version8.5", "8.5",
"9.0") = "Version9.0"
```

String function: TRIM

- **TRIM**
  - Returns the string with leading and trailing spaces removed.

- Example:

```
TRIM(" Calculation ") =

"Calculation"
```

Using string functions to clean data

- In our data, **local country name** has characters that might cause problems.

- For example **spaces**, ‘, and / can be problematic since they can break up strings or cannot be interpreted in some programs.

- Let’s make a column with the local names after data cleaning.

Local Name	LocalName (country.csv1)
Al-Jaza'ir/Algérie	Al-Jaza'ir/Algérie
Al-Jaza'ir/Algérie	Al-Jaza'ir/Algérie
Al-Jaza'ir/Algérie	Al-Jaza'ir/Algérie
Amerika Samoa	Amerika Samoa
Amerika Samoa	Amerika Samoa
Andorra	Andorra
Angola	Angola
Angola	Angola
Angola	Angola
Angola	Angola
Angola	Angola
Anguilla	Anguilla
Anguilla	Anguilla
Antigua and Barbuda	Antigua and Barbuda
Al-Imarat al-´Arabiya al-Muttahida	Al-Imarat al-´Arabiya al-Muttahida
Al-Imarat al-´Arabiya al-Muttahida	Al-Imarat al-´Arabiya al-Muttahida
Al-Imarat al-´Arabiya al-Muttahida	Al-Imarat al-´Arabiya al-Muttahida
Al-Imarat al-´Arabiya al-Muttahida	Al-Imarat al-´Arabiya al-Muttahida
Al-Imarat al-´Arabiya al-Muttahida	Al-Imarat al-´Arabiya al-Muttahida
Argentina	Argentina
Argentina	Argentina
Argentina	Argentina
Argentina	Argentina
Argentina	Argentina

Nesting string functions

- We can simplify the process by using nested replacements.

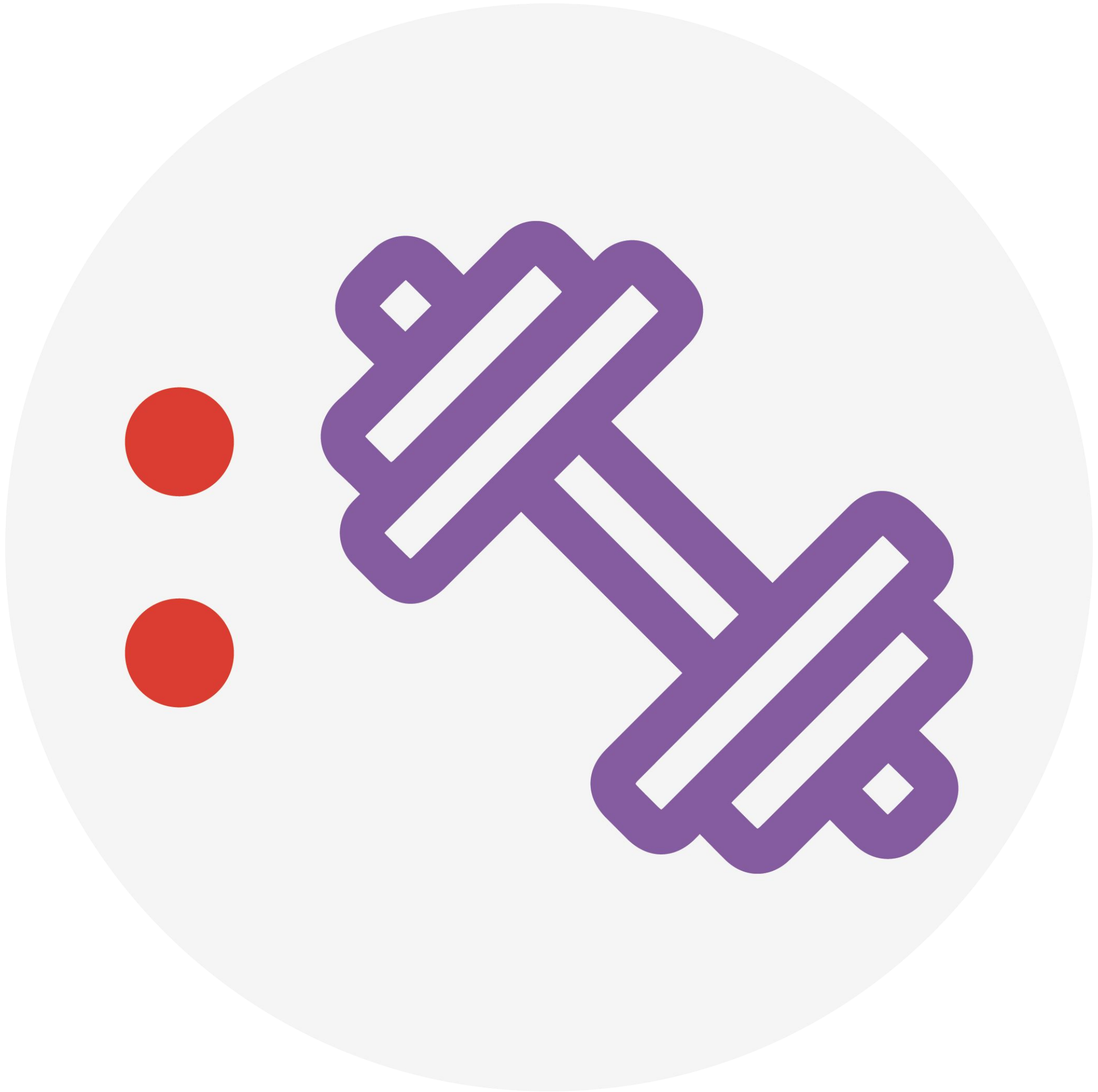
Localnames_clean

```
trim(REPLACE(REPLACE([LocalName (country.csv1)]," ","_"),"/","_"))
```

LocalName (country.csv1)	Localnames_clean
Al-Jaza'ir/Algérie	Al-Jaza'ir_Algérie
Al-Jaza'ir/Algérie	Al-Jaza'ir_Algérie
Al-Jaza'ir/Algérie	Al-Jaza'ir_Algérie
Amerika Samoa	Amerika_Samoa
Amerika Samoa	Amerika_Samoa
Andorra	Andorra
Angola	Angola
Angola	Angola
Angola	Angola
Angola	Angola
Angola	Angola
Anguilla	Anguilla
Anguilla	Anguilla
Antigua and Barbuda	Antigua_and_Barbuda

Knowledge check 9

Exercise 9



Module completion checklist

Objective	Complete
Implement string calculations on a given dataset	✓
Implement date calculations on a given dataset	
Implement type calculations on a given dataset	
Implement logic calculations on given dataset	

End of Part 9