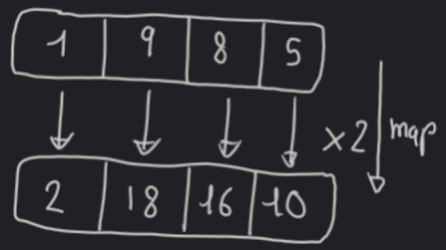
Advanced Data Analysis
map() & Custom Functions

Emanuele Della Valle
Prof. @ Politecnico di Milano
Founder & Partner @ Quantia Consulting
Marco Balduini
Founder & CEO @ Quantia Consultin

Exploring flux
map()

Map()

- How does it work?
  - It applies a function to each element of a collection
  - e.g., double each number

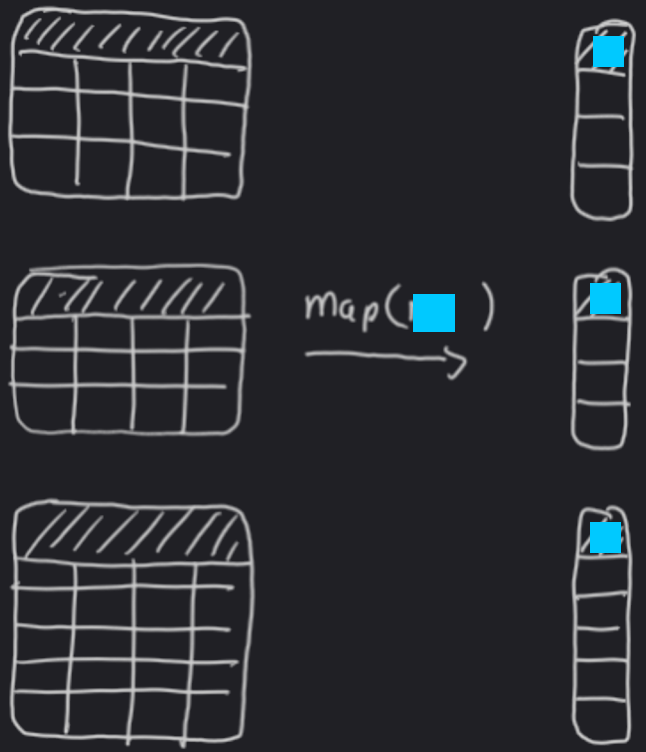


- What's for?
  - Combine data
  - Derive data
  - Enrich data
  - Clean data
  - …
  - Turn data into information!

Map() in flux

- The map() function applies a function to each record in the input tables
- The modified records are assigned to new tables based on the group key of the input table

Combine/derive data with map()



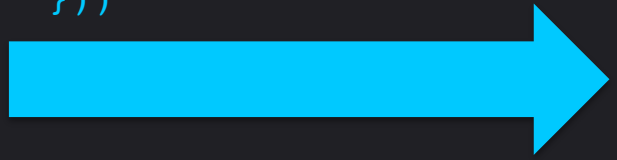
_time	_m	_field	_value
10:45	cpu	system	1000
11:45	cpu	system	3000
GroupKey[cpu, system]			

_time	_m	_field	_value
11:00	cpu	system	2000
11:30	cpu	system	2000
GroupKey[cpu, system]			

_time	_m	_field	_value
10:35	cpu	idle	100
GroupKey[cpu, idle]			

_time	_m	_field	_value
11:15	cpu	idle	200
11:50	cpu	idle	400
GroupKey[cpu,, idle]			

```
map(fn: (r) =>
  ({ hour:
      date.hour(t: r._time)
  }))
```



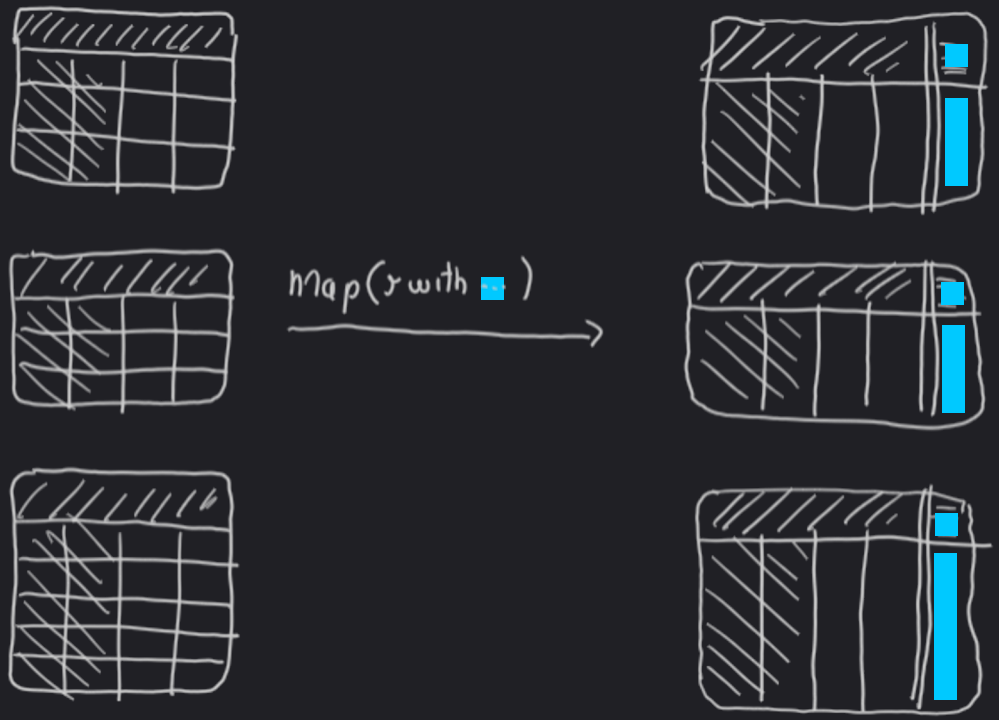
hour
10
11

hour
11
11

hour
10

hour
11
11

Enrich data using map(r with …)



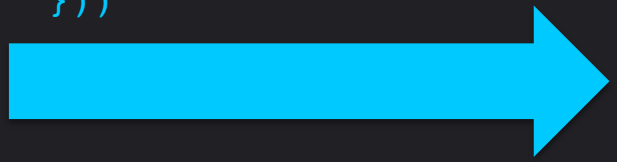
_time	_m	_field	_value
10:45	cpu	system	1000
11:45	cpu	system	3000
GroupKey[cpu, system]			

_time	_m	_field	_value
11:00	cpu	system	2000
11:30	cpu	system	2000
GroupKey[cpu, system]			

_time	_m	_field	_value
10:35	cpu	idle	100
GroupKey[cpu, idle]			

_time	_m	_field	_value
11:15	cpu	idle	200
11:50	cpu	idle	400
GroupKey[cpu,, idle]			

```
map(fn: (r) =>
  ({ r with hour:
      date.hour(t: r._time)
  }))
```



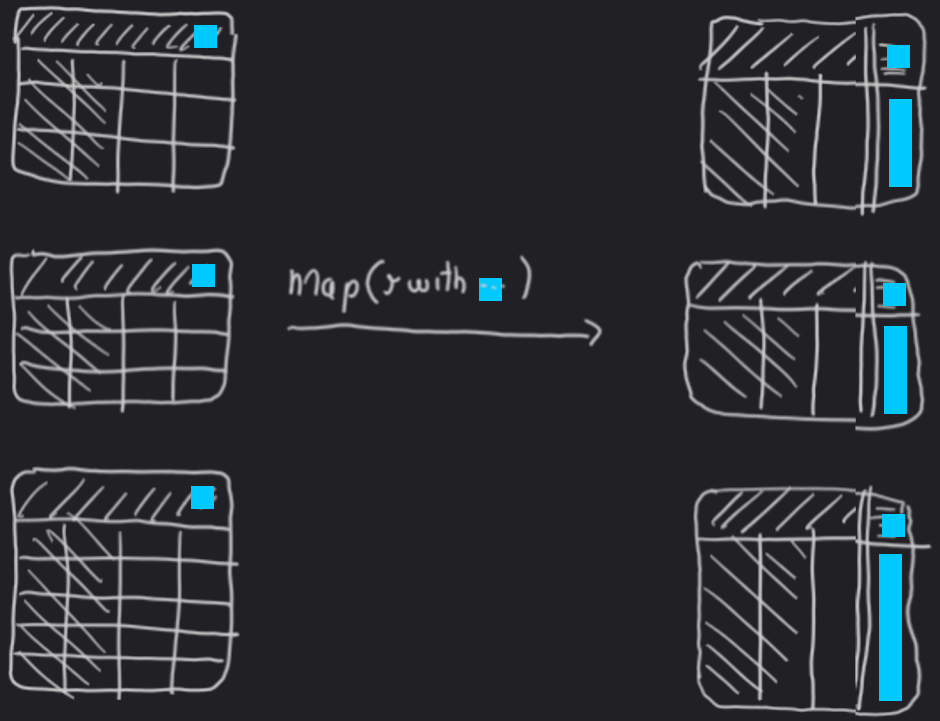
_time	_m	_field	_value	hour
10:45	cpu	system	1000	10
11:45	cpu	system	3000	11
GroupKey[cpu, system]				

_time	_m	_field	_value	hour
11:00	cpu	system	2000	11
11:30	cpu	system	2000	11
GroupKey[cpu, system]				

_time	_m	_field	_value	hour
10:35	cpu	idle	100	10
GroupKey[cpu, idle]				

_time	_m	_field	_value	hour
11:15	cpu	idle	200	11
11:50	cpu	idle	400	11
GroupKey[cpu,, idle]				

Clean data using map(r with …)



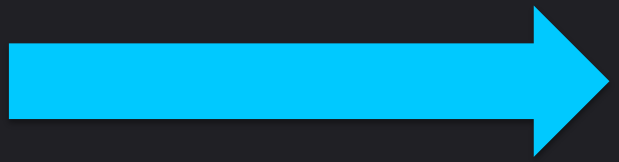
_time	_m	_field	_value
10:45	cpu	system	1000
11:45	cpu	system	3000
GroupKey[cpu, system]			

_time	_m	_field	_value
11:00	cpu	system	2000
11:30	cpu	system	2000
GroupKey[cpu, system]			

_time	_m	_field	_value
10:35	cpu	idle	100
GroupKey[cpu, idle]			

_time	_m	_field	_value
11:15	cpu	idle	200
11:50	cpu	idle	400
GroupKey[cpu,, idle]			

```
map(fn: (r) =>
  ({ r with _value :
       r._value/3000.0 }))
```



_time	_m	_field	_value
10:45	cpu	system	33.3%
11:45	cpu	system	100%
GroupKey[cpu, system]			

_time	_m	_field	_value
11:00	cpu	system	66.6%
11:30	cpu	system	66.6%
GroupKey[cpu, system]			

_time	_m	_field	_value
10:35	cpu	idle	3.3%
GroupKey[cpu, idle]			

_time	_m	_field	_value
11:15	cpu	idle	6.6%
11:50	cpu	idle	13.3%
GroupKey[cpu,, idle]			

Exploring flux

Custom Functions

Recall: What Is Flux?

- Flux is a functional data scripting and query language
- Written to be:
  - Useable: easy to learn
  - Readable: developers read more code than we write
  - **Composable: developers can build onto the language**
  - Testable: queries are code
  - Contributable: open source contributions matter

Defining and using a custom function

- Syntax

  <<function name>> **= (**<<variable>>*__)__ **=>** <<implementation>>

- Example

```
squared = (r) => r*r
from(bucket:"foo")
    |> range(start: -1h)
    |> filter(fn: (r) => r._measurement == "samples")
    |> map(fn: (r) => ({ _value: squared(r._value)}))
    |> filter(fn: (r) => r._value > 23.2)
```

Defining a custom pipe forwardable function

- Syntax

  <<function name>> = (<<table>>=<-, <<variable>>*) =>
  <<table>> |> << implementation >>

- Example

```
allSquared = (tables=<-) =>
        tables |> map(fn: (r) => squared(r._value))

from(bucket:"foo")
        |> range(start: -1h)
        |> filter(fn: (r) => r._measurement == "samples")
        |> allSquared()
        |> filter(fn: (r) => r._value > 23.2)
```

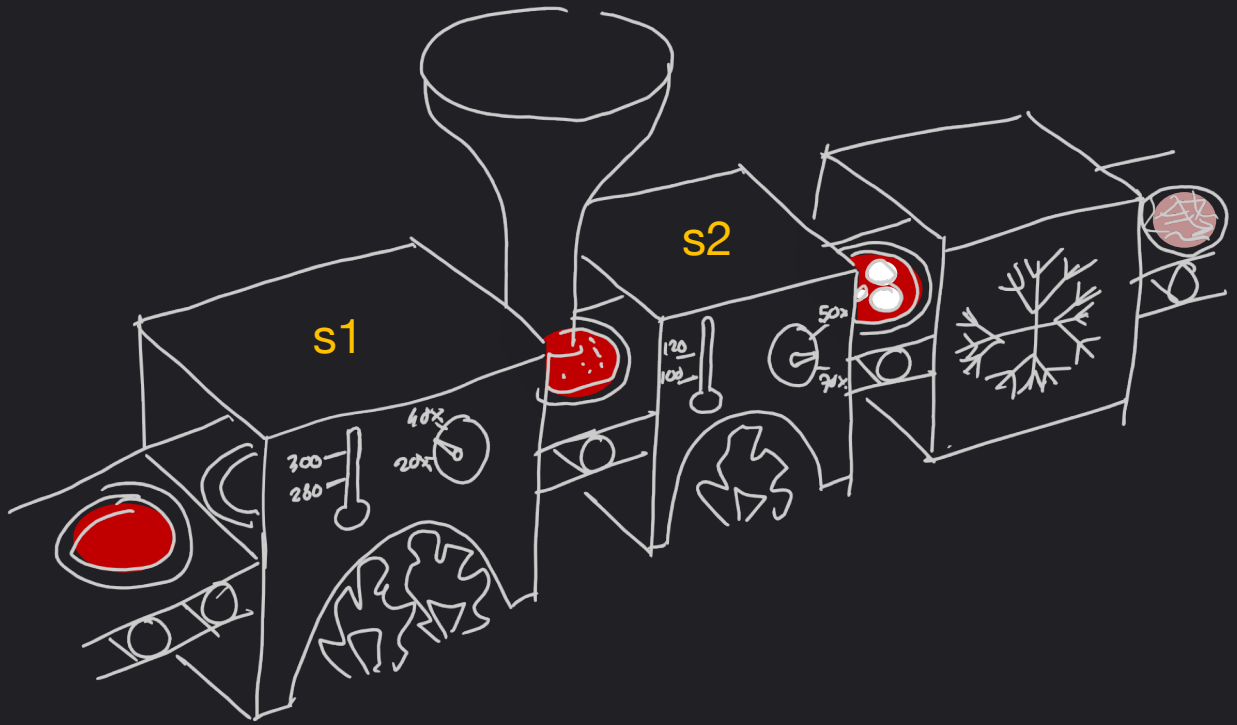
Let’s get dirty!

7

Continuous Linear Pizza Oven

Learning goals:

- Map functions
- Custom functions



Task

Correct the temperature observations of the cooking base area by by subtracting a delta of 5°C to each value

- Use an inline map
- Create a custom function to be used in the inline map
- Create a custom pipe forwardable function
  that contains a map

Quiz

- True or false
  - in a functional programming language, map() applies a function to each element of a collection – ?
  - in Flux, map() always adds a column to each table – ?
  - map() using the `with` clause can add a column to each table – ?
  - you can pass a custom functions only to a map() – ?
  - in a pipe-forwardable custom function with a parameter `(t=<-)`, `t` represents input tables that the function applies to – ?

Quiz answers

- True or false
  - in a functional programming language, map() applies a function to each element of a collection – T
  - in Flux, map() always adds a column to each table – F
  - map() using the `with` clause can add a column to each table – T
  - you can pass a custom functions only to a map() – F
  - in a pipe-forwardable custom function with a parameter `(t=<-)`, `t` represents input tables that the function applies to – T

Advanced Data Analysis
map() & Custom Functions

Emanuele Della Valle Prof. @ Politecnico di Milano & Partner @ Quantia Consulting
Marco Balduini Founder & CEO @ Quantia Consulting