DataHelix - the data generator
Andrew Carr & Simon Laing
1st May 2019
The problem
The Problem

01010101010101010101111100001101010111100001111000

FCA
FINANCIAL CONDUCT AUTHORITY
Solutions in the Market
# Solutions on the Market

<table>
<thead>
<tr>
<th>Tool name</th>
<th>Complex conditions</th>
<th>User functions to create complex types</th>
<th>Common FS types (RIC, ISIN, CUSIP, …)</th>
<th>Random mode</th>
<th>Profiler to auto generate profiler from real data</th>
<th>Violation mode</th>
<th>Ability to stream data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mockaroo</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Redgate (SQL) data generator</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes (seeded)</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>generatedata.com</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>FINRA data generator</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Data Helix
Open source Data Generator
Data Helix - USPs

- What is the Data Helix?
  - Generator (in beta)
  - Profiler (in development)

- Being considered as a contribution into FINOS
Data Helix - A possible solution

Data Profile + Test data attributes
### Data Helix - USPs

<table>
<thead>
<tr>
<th>Tool name</th>
<th>Complex conditions</th>
<th>User functions to create complex types</th>
<th>Common FS types (RIC, ISIN, CUSIP, …)</th>
<th>Random mode</th>
<th>Profiler to auto generate profiler from real data</th>
<th>Violation mode</th>
<th>Ability to stream data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Helix</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- Profiler → Production data → Data Profile using AI
- Data Profile from Production → review → clean Data Profile for generating Production like data
- Data Profile + Test Data attributes = random data | data violations
Data Profile

- Columns and types (integer, float, string, date)
- Complex types (ISINs, CUSIPs, RIC, …)
- Range for column
- Conditions for column (restrictions under certain conditions)
Data Generator
What can it do?
Conditions

Profile:
Fields
Constraints
Restrictions
Relationships

Constraints:
Relationships between fields
Conditional constraints
Optimisation & simplification
Data types, restrictions, ...

Boolean logic
Provides guarantees about data production
Examples

- If RIC_Code column = “BT.L” then Price Column Min = 20.0, Max = 40

```json
{
  "if": {
    "field": "ric_code",
    "is": "equalTo",
    "value": "BT.L"
  },
  "then": {
    "allOf": [
      {
        "field": "min",
        "is": "equalTo",
        "value": 20.0
      },
      {
        "field": "max",
        "is": "equalTo",
        "value": 40
      }
    ]
  }
}
```
Problem solving

- Constraint
- Constraint
- Constraint
- Constraint

- Memory
- CPU
- Time!

- Permitted values
- Contradiction analysis
- Interconnected fields
- Problem
- Complexity

- SharePercentage between 1 & 100
- ShareName shorter than 200
- TransactionDate before today

Representations of valid data
Profiler

What can it do?
Profiler

- Uses traditional stats and AI to magically guess data profile
- Rules, types, correlations between columns
- Creating Profile from Production data → quick and easy
- Built on Apache Spark to scale
Summary

- Data Helix is ready for use
  - Data Generator (in beta)
  - Profiler (in development - not needed)
- Being considered for FINOS contribution

- Visit - https://github.com/scottlogic/datahelix/
- Or chat to Simon, Colin or myself
- Thank you for your time and questions