Fluid:
Explorable, Transparent Data Visualisation

Roly Perera[1, 2]

[1] Institute of Computing for Climate Science, University of Cambridge
[2] School of Computer Science, University of Bristol

- **Project overview**
- Demos & UX discussion
- Creating content with Fluid
- Future directions
- Call for Collaboration!

Project overview

Contributors (past & present)

Joe Bond [1]
Haofei Chen [4]
Colin Crawford [4]
Cristina David [1]
Thomas Frith [2]
Harleen Gulati [1]
Hana Iza Kim [2]
Minh Nguyen [1]
Dominic Orchard [5, 2]
Roly Perera [2, 1]
Tomas Petricek [3]
Achintya Rao [6, 2]
Meng Wang [1]

[1]University of Bristol
[2]University of Cambridge
[3]Charles University
[4]University of Edinburgh
[5]University of Kent
[6]University of West of England

https://f.luid.org
https://github.com/explorable-viz/fluid

Demo: renewables

```
let series type country = [
   { x: row.year, y: row.output }
   | year ← [2013..2018], row ← renewables,
   row.year == year, row.energyType == type, row.country == country
] in LineChart {
   caption: "Output of USA relative to China",
   plots: [
      LinePlot { name: type, data: plot }
      | type ← ["Bio", "Hydro", "Solar", "Wind"],
      let plot = zipWith (fun p1 p2 → { x: p1.x, y: p1.y / p2.y })
                         (series type "USA") (series type "China")
   ]
}
```

Demo: convolution

```
let zero n = const n;
    wrap n n_max = ((n - 1) `mod` n_max) + 1;
    extend n = min (max n 1);

let convolve image kernel method =
    let ((m, n), (i, j)) = (dims image, dims kernel);
        (half_i, half_j) = (i `quot` 2, j `quot` 2);
        area = i * j
    in  [| let weightedSum = sum [
              image!(x, y) * kernel!(i' + 1, j' + 1)
              | (i', j') ← range (0, 0) (i - 1, j - 1),
                let x = method (m' + i' - half_i) m,
                let y = method (n' + j' - half_j) n,
                x ⩾ 1, x ⩽ m, y ⩾ 1, y ⩽ n
           ] in weightedSum `quot` area
           | (m', n') in (m, n) |];
```

Language overview

Current design

- Purely functional (no side-effects)
- Untyped
- Records, lists, dictionaries, 2D arrays
- Graphics library based on **d3.js**

Implemented in **PureScript**
(Haskell clone for the web)

What's missing

- Modules and imports
- I/O — load from file, db or URL
- User-defined datatypes
- Type system (with units of measure?)

Future directions

How to enable a smooth transition from content consumption to content creation?

Readers:

- Don’t care how it works
- Want responsive, self-explanatory, intuitive UI
- Should be able to transition from passive reading to active engagement
- UI affordances (opportunities for interaction) should present themselves

We wear different hats at different times..

Authors:

- Proficient in technology
- Invested in specific workflows and skills
- Benefits of new technology need to be obvious
- Barriers to entry need to be low

What are the prospects of doing something like Fluid for R or Python?

thanks!

Roly Perera[1, 2] roly.perera@cl.cam.ac.uk

github.com/explorable-viz/fluid

[1] Institute of Computing for Climate Science, University of Cambridge
[2] School of Computer Science, University of Bristol