

my current reality

Agile Coach

Entrepreneur

Engineer



@haroldcampbell
twitter | LinkedIn | github

A different way to
access, see and think
about data

Show DevCon:

groundTap

An **early preview** of
my side-project

The **GOAL**

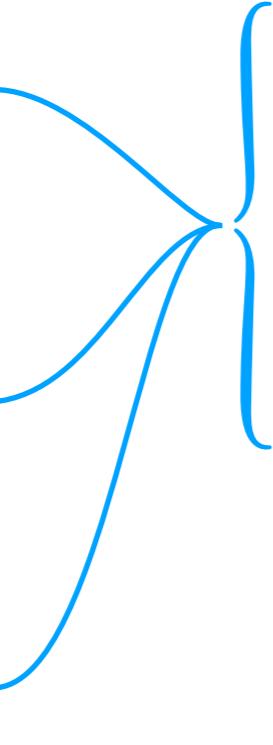


**HELP COMPANIES
REDUCE THE TIME
IT TAKES TO MAKE DECISIONS**

Light-weight
Computational
Notebook

JS Visualisation
Library

Docker
Build Process



**REDUCE THE TIME
IT TAKES TO MAKE DECISIONS**

There was the plan.

This was the plan.

1. Talk to a few potential customers
2. Build it all in a **few short months**
3. Marvel at my creation.

This was the plan.

-  1. Talk to a few potential customers
- 2. Build it all in a few short months
- 3. Marvel at my creation.



COVID infused dose of reality!

DATA BY ITSELF IS USELESS IF WE CAN'T
**USE DATA TO MAKE DECISIONS IN A
TIMELY MANNER**

DATA BY ITSELF IS USELESS IF WE CAN'T
USE DATA TO MAKE DECISIONS IN A
TIMELY MANNER

**Light-weight
Computational
Notebook**

**JS Visualisation
Library**

**Docker
Build Process**



**REDUCE THE TIME
IT TAKES TO MAKE DECISIONS**

Find the simplest way,
using CSV data:

Light-weight
Computational
Notebook

JS Visualisation
Library

Docker
Build Process

**REDUCE THE TIME
IT TAKES TO MAKE DECISIONS**

The **GOAL**

Medium - Large
HELP[^] COMPANIES
REDUCE THE TIME
IT TAKES TO MAKE DECISIONS

What is
this thing
anyway



SIMPLE TOOL



Web-based
Easy to use
Helps people to
make decisions

...still in development



SIMPLE TOOL

The background of the image is a vibrant, abstract composition of overlapping geometric shapes. It features several large, semi-transparent circles in shades of yellow, blue, red, and purple. These are set against a backdrop of solid-colored rectangles in black, white, and grey. The overall effect is reminiscent of a stained-glass window or a modern abstract painting.

VISUALISATION



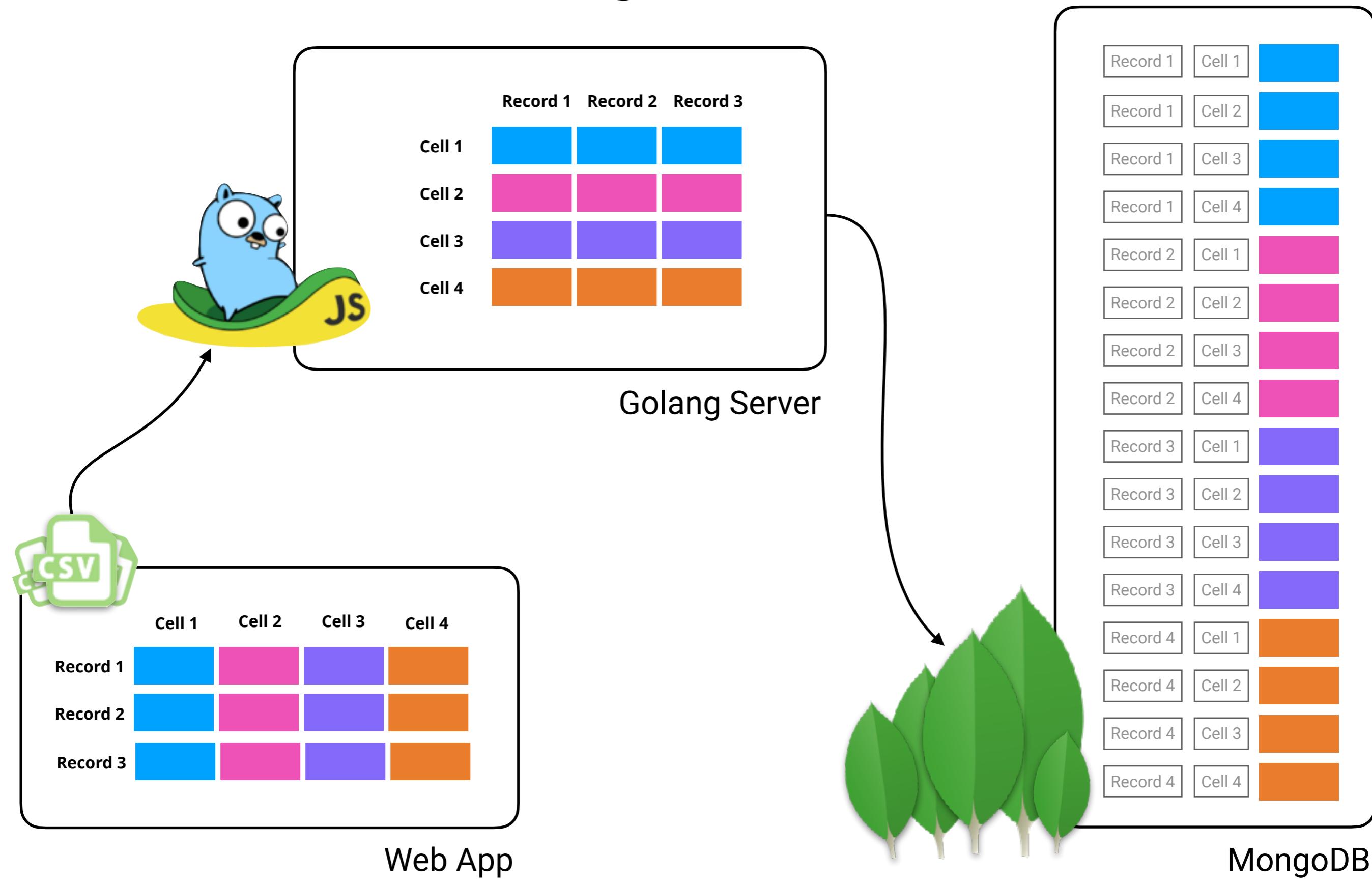
VISUALISATION

Flexible library
Renders SVG
Pluggable

fairly mature...still in dev.

DEMO #1

What did we just see?



The Tech

The Tech

Back-End

Front-End

Back-End

Front-End

Web-only

😭 Not responsive

😭😭 Not mobile first

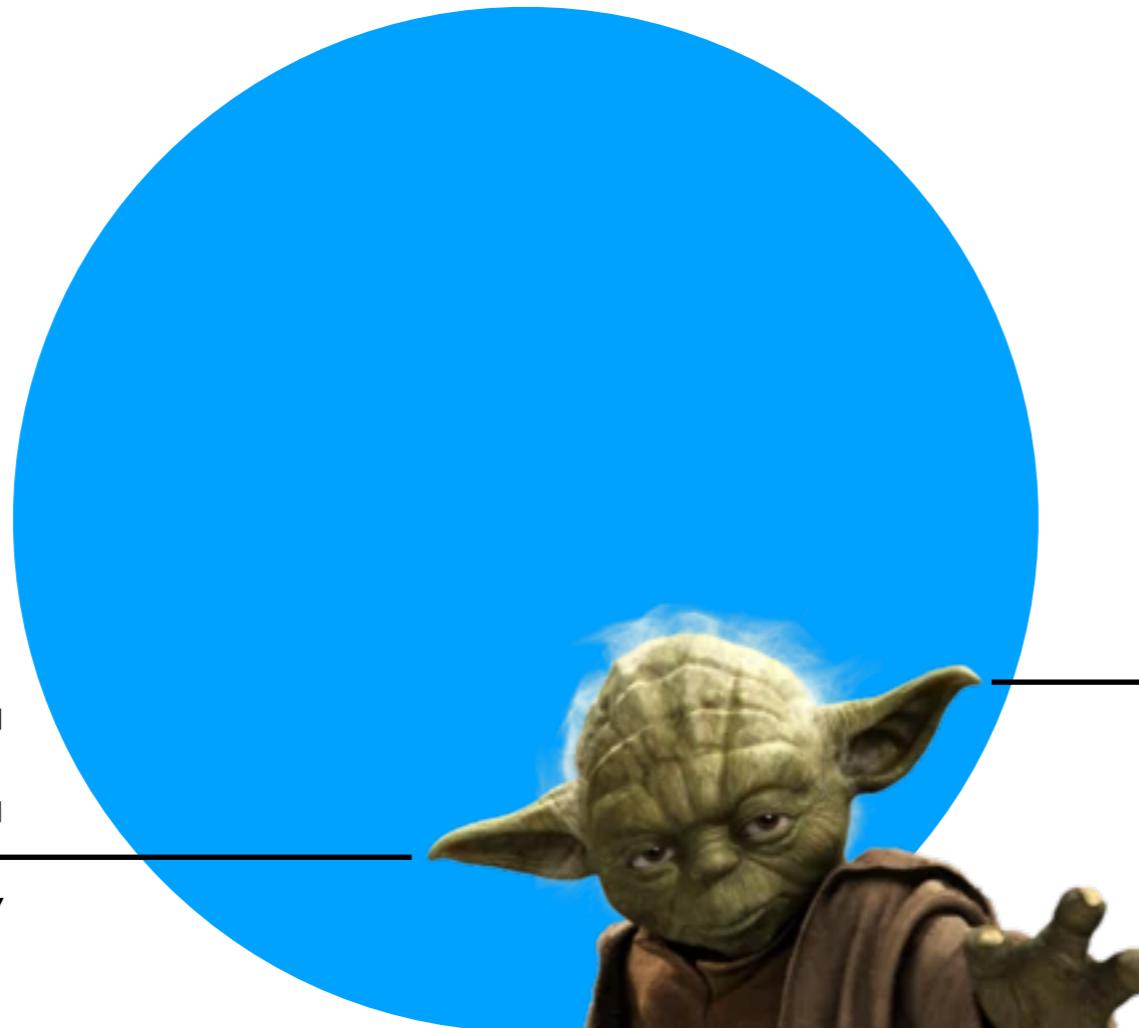


BE

LAZY



BE
—
LAZY

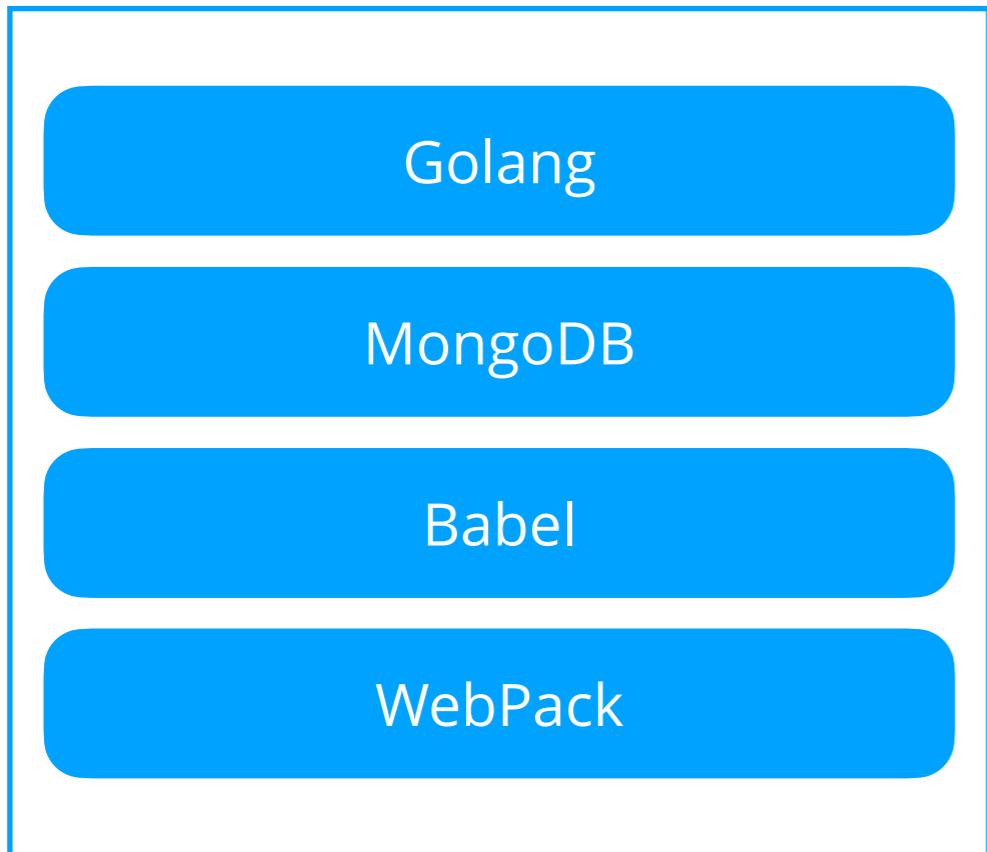


SIMPLICITY
—
...WORK NOT DONE

Web App

Web Server

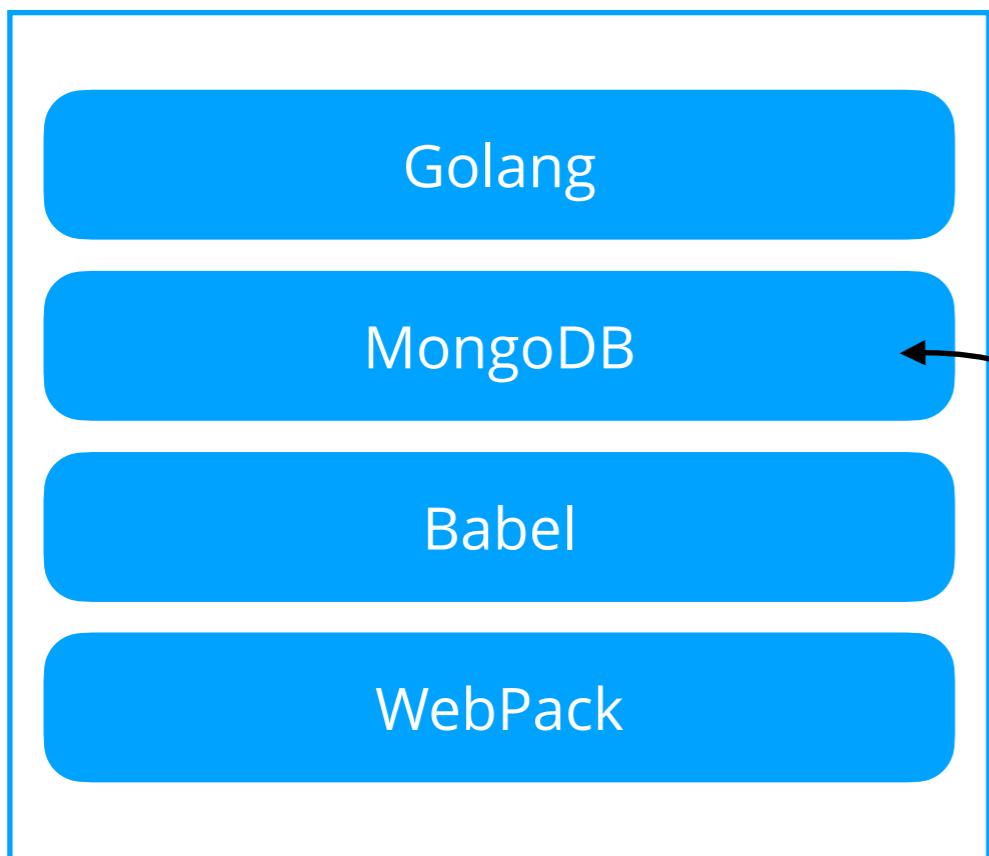
Web Server



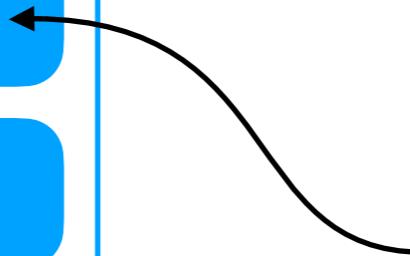
Web App



Web Server



JavaScript



Docker

Build

Docker

Package

Docker: Alpine

Run

Local Machine

Cloud

Why Go for me?



From Google

Compiled Language

Static Code Analysis

Concurrency

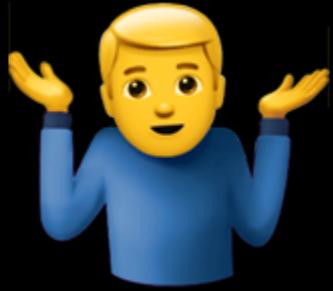
Race Condition Detection

Built-in Testing Framework

= Great For Tools & Servers

How ^{currently} I use Go

Data Orchestration | GUID Creation | Web Server



Future plans with Go

Security | Automation | Intelligence

Map Reduce | Micro-services | Batch Processing

Why MongoDB?



Opensource

Native MapReduce

Handles millions of records with ease

Native Scaling

Support server-side JavaScript

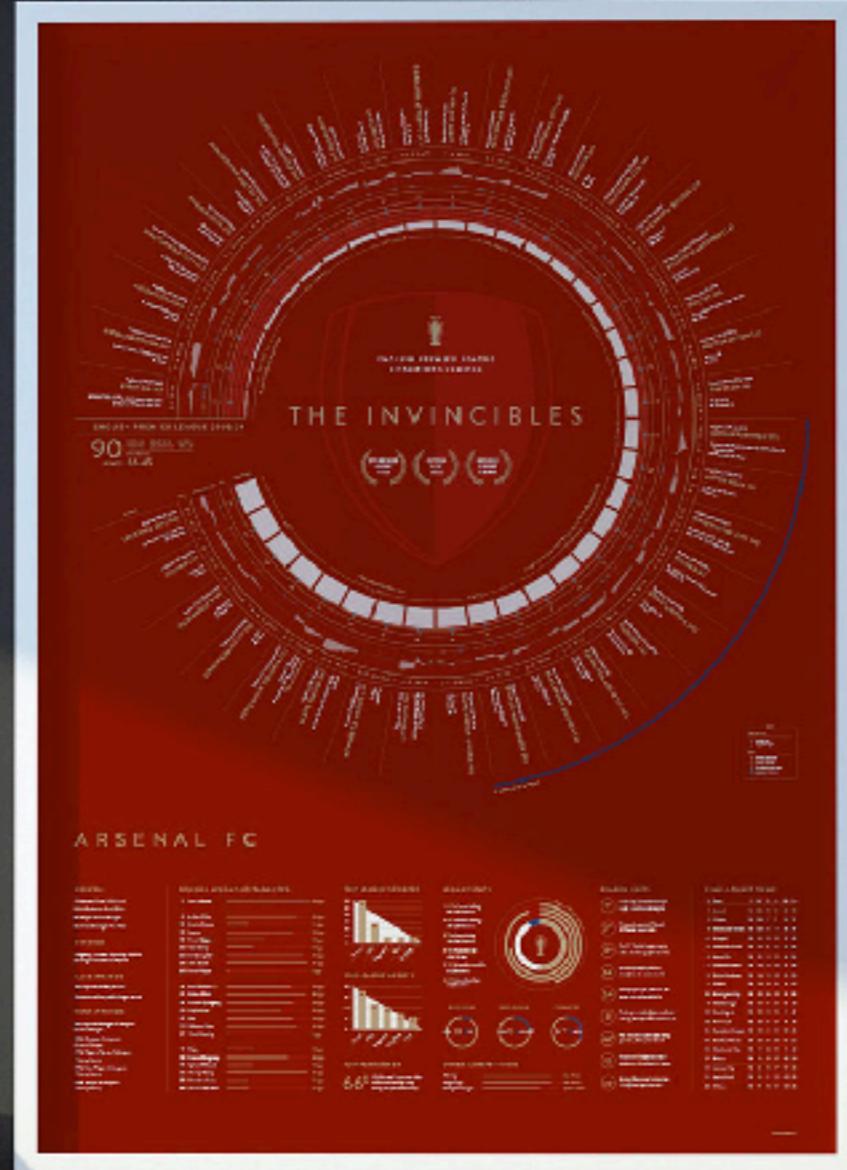
Plug-in for Hadoop integration

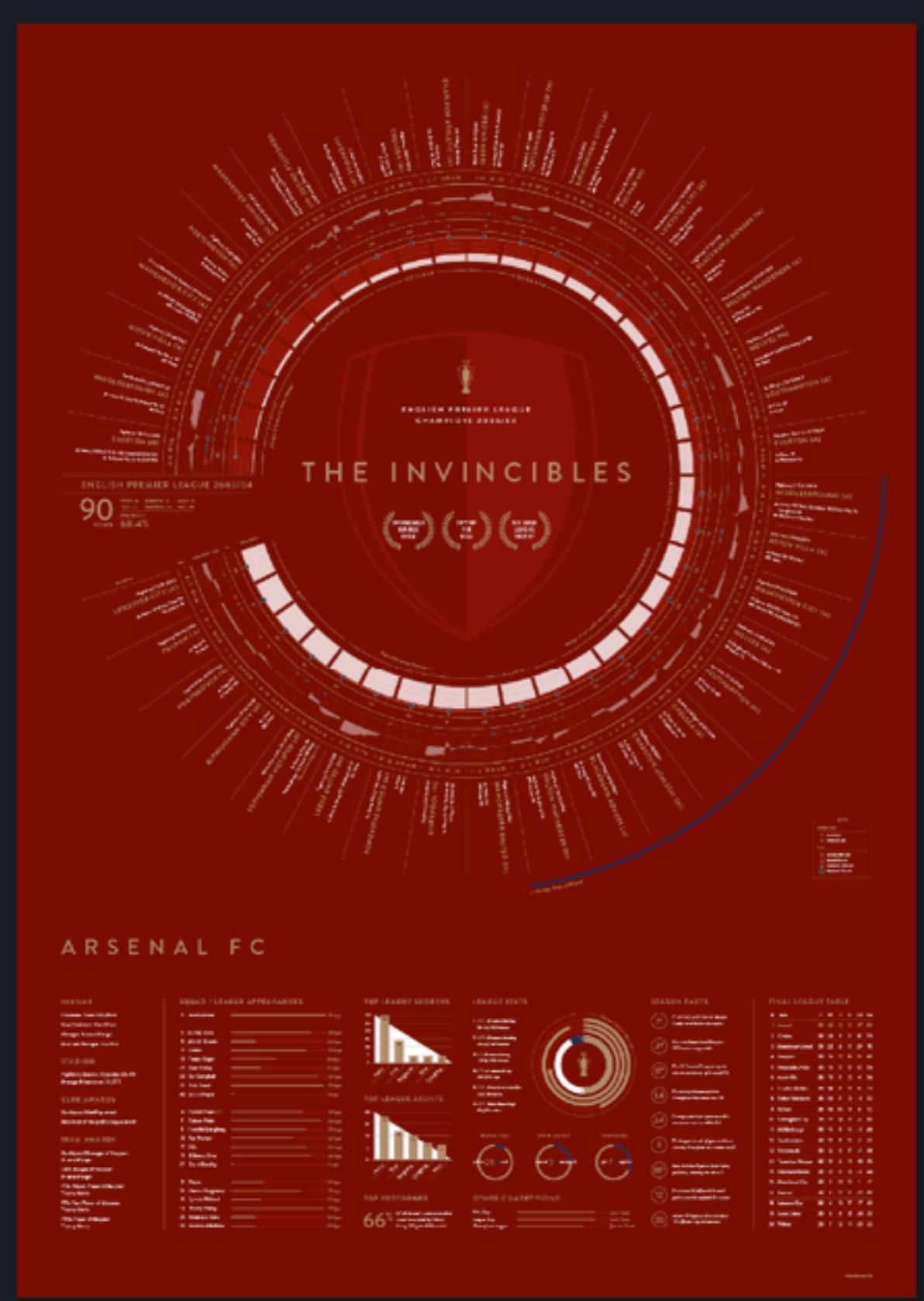
= Great For Data Analytics Tools

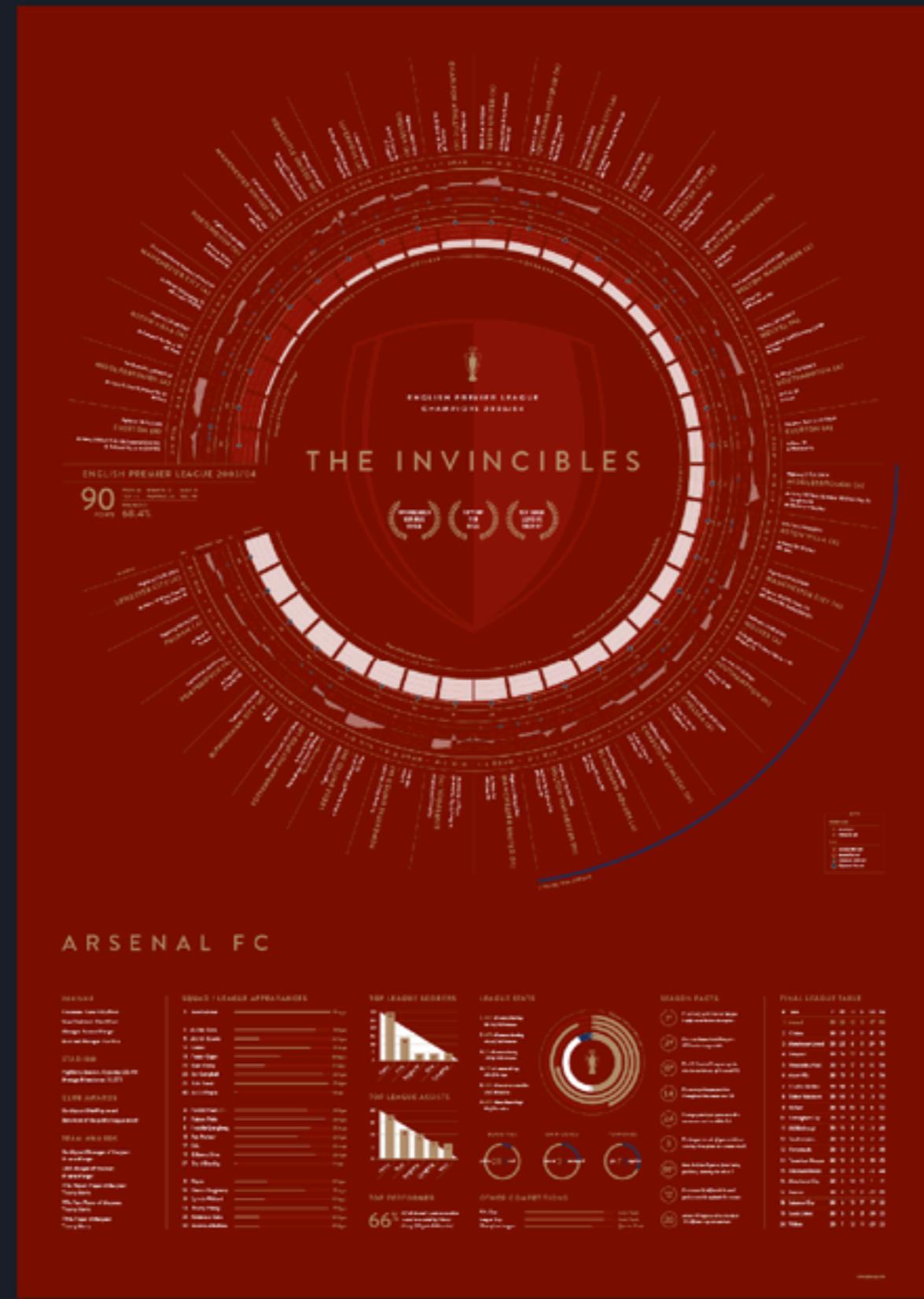
Creativity

VS

Tool





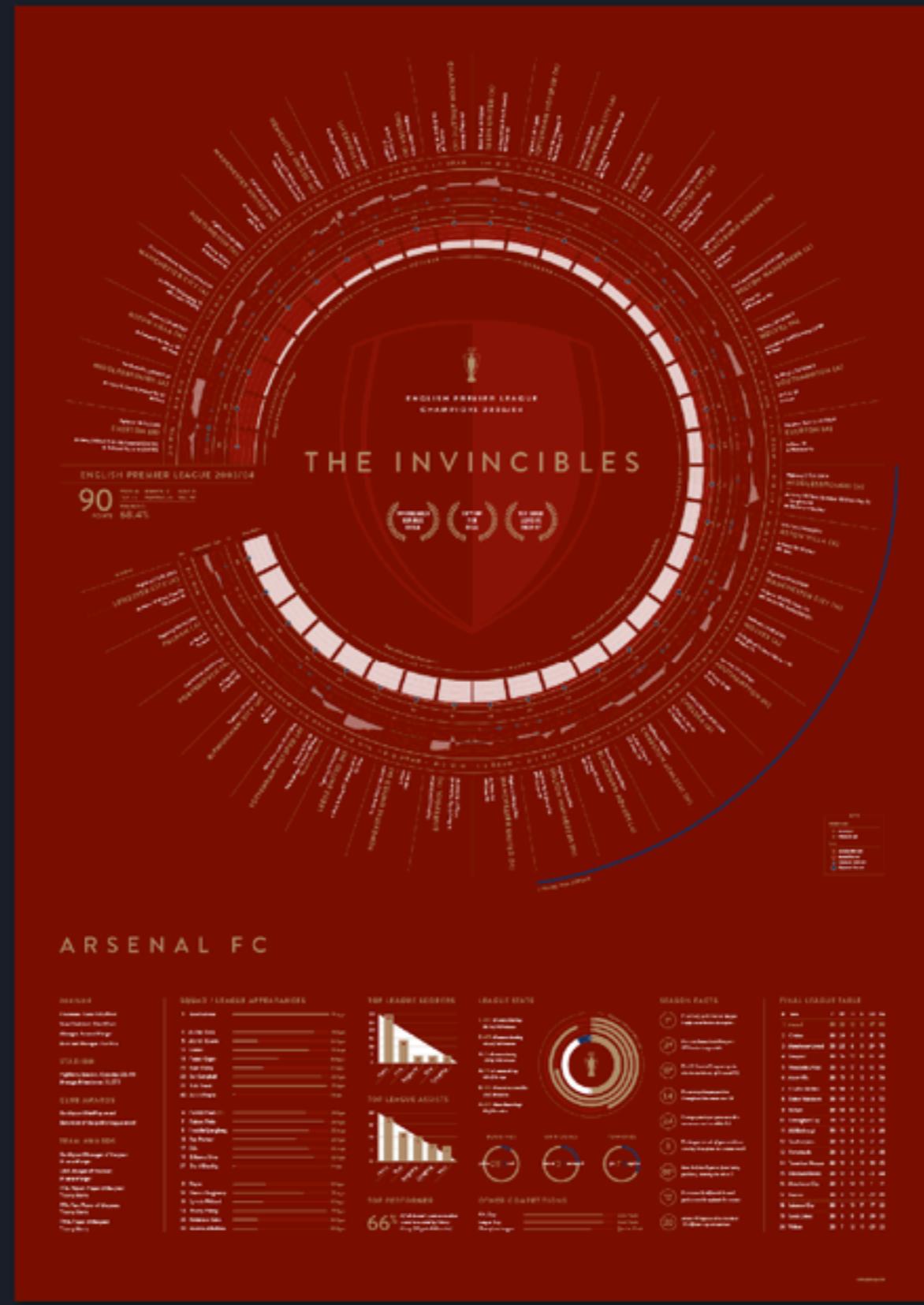


What would be required to build this in

?

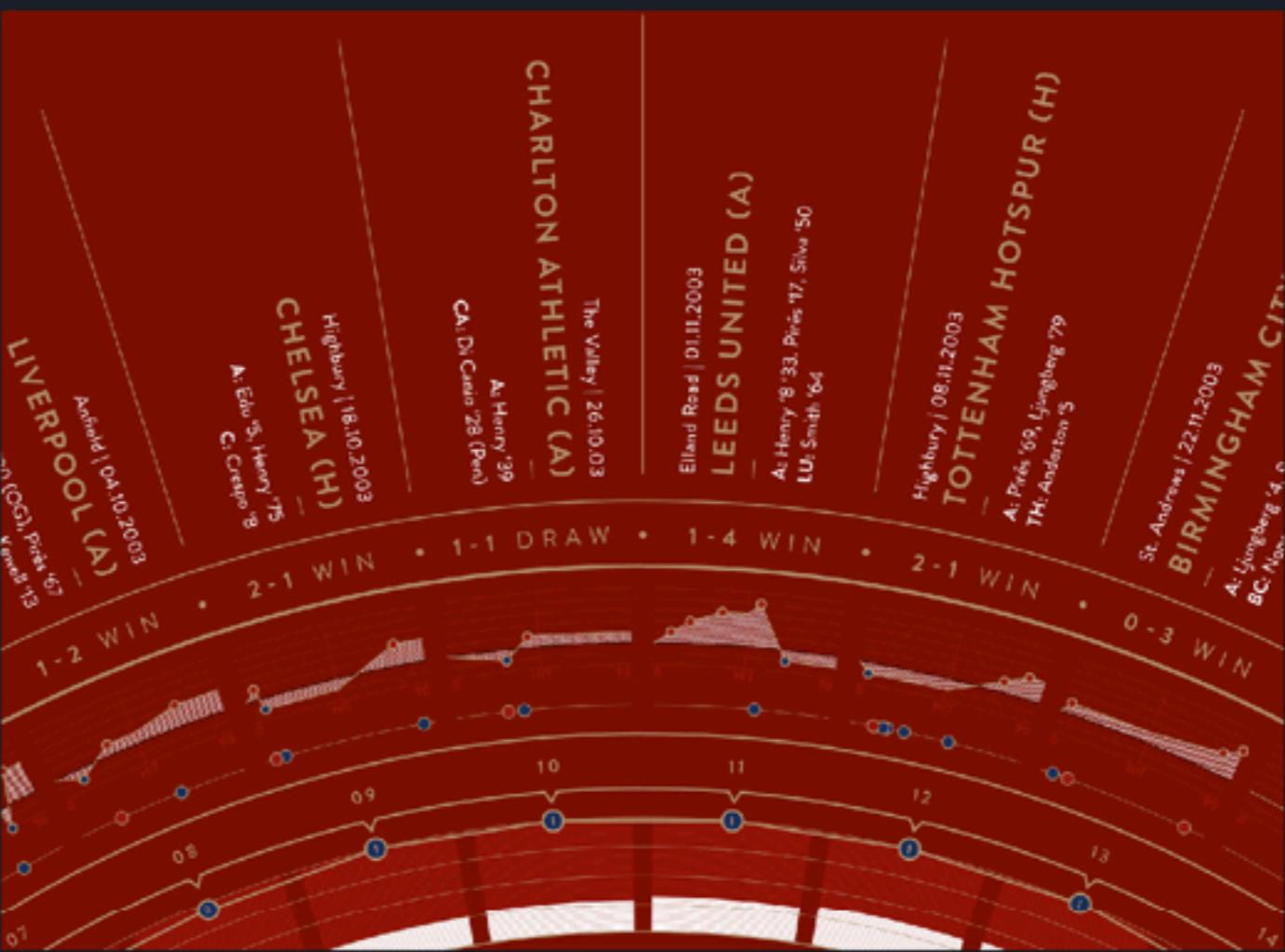
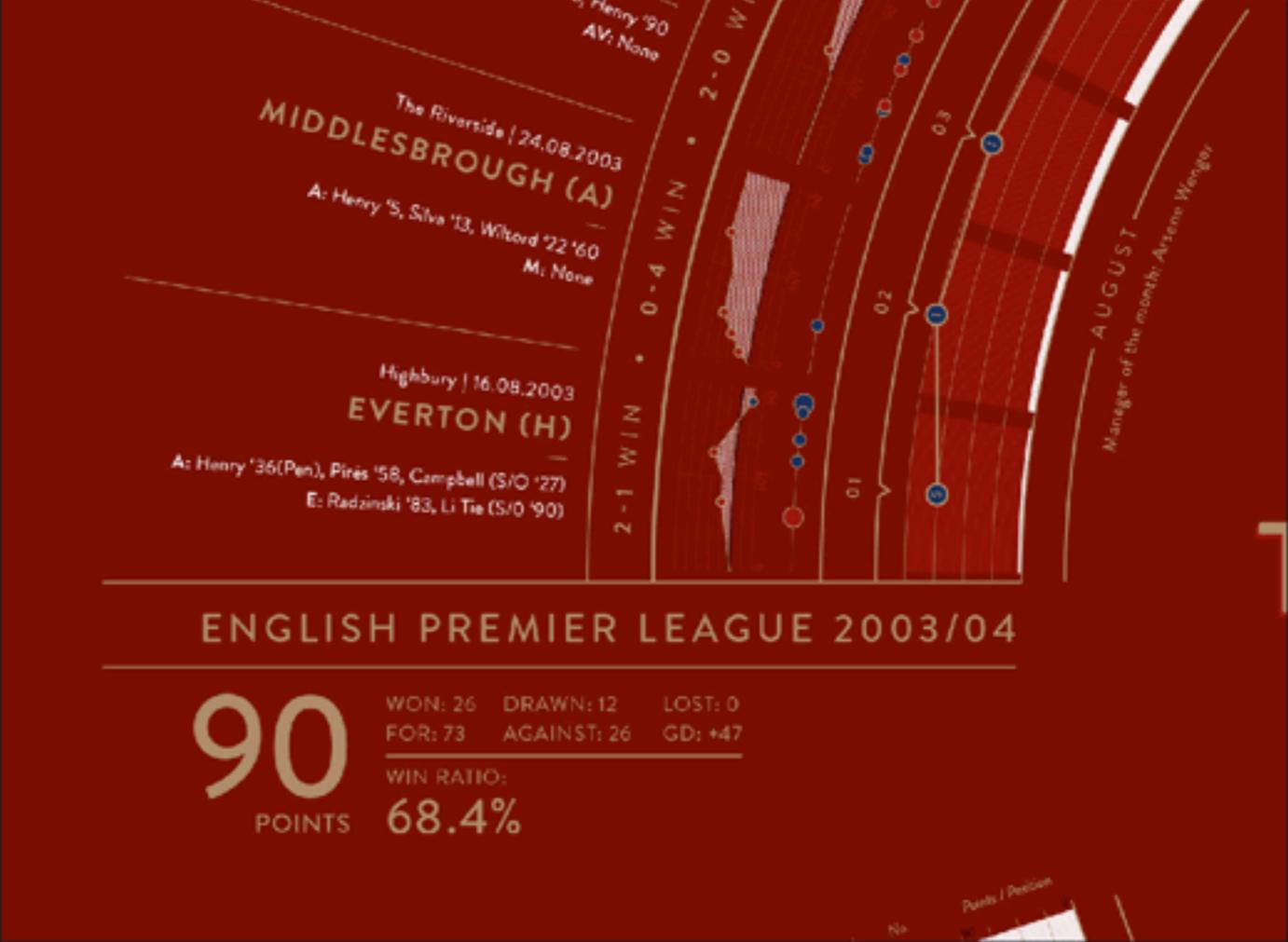
Insert current tool here

PowerBI
Tableau
Excel
MatPlotLib
D3JS

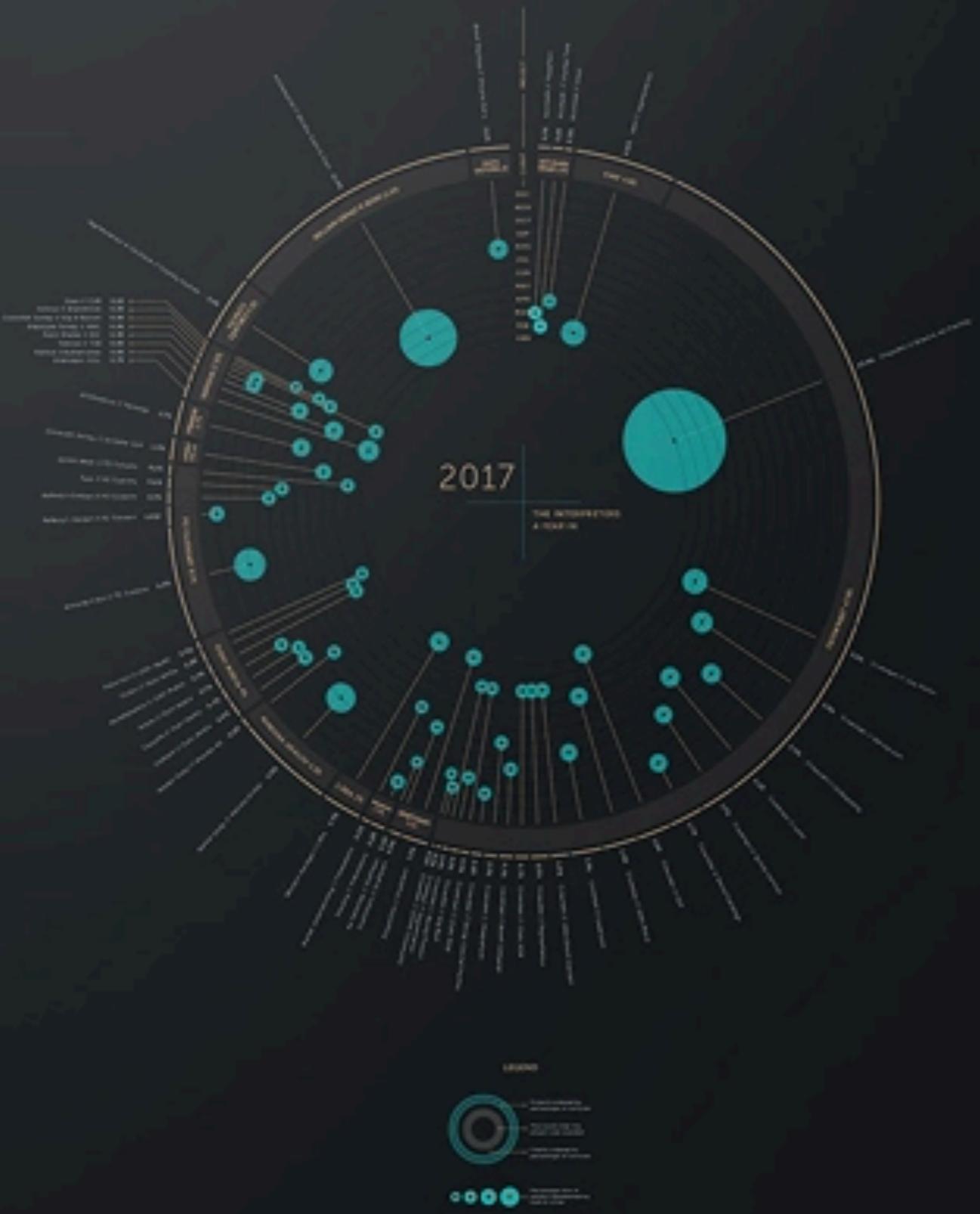


What would be
required to build
this in

Insert current tool here



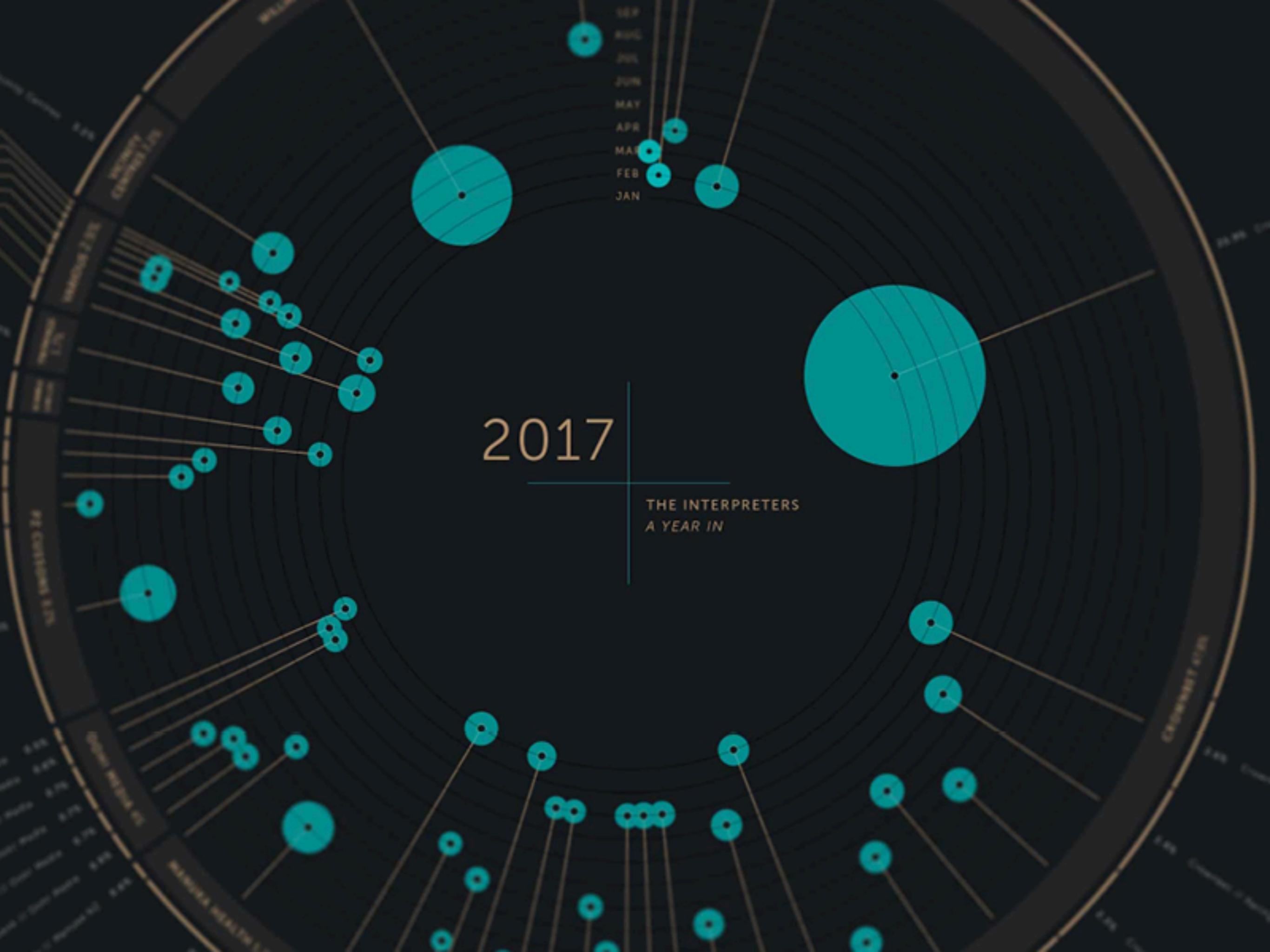
DEMO #2



Theinterpreters Yearreview Dribbble2

By Rileigh Design

<https://dribbble.com/shots/4818703>

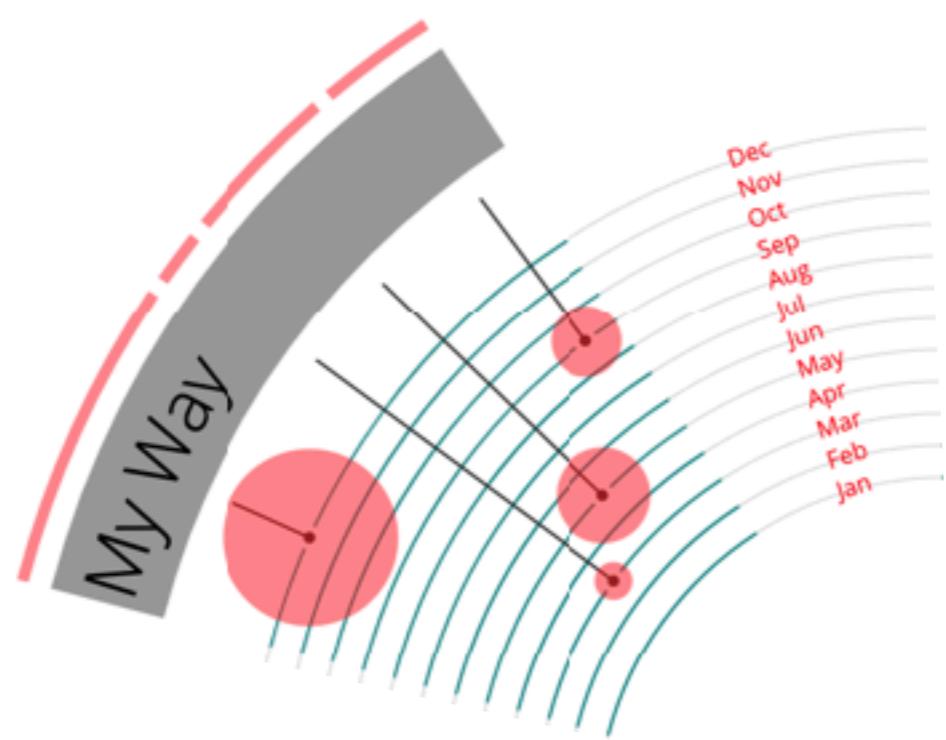


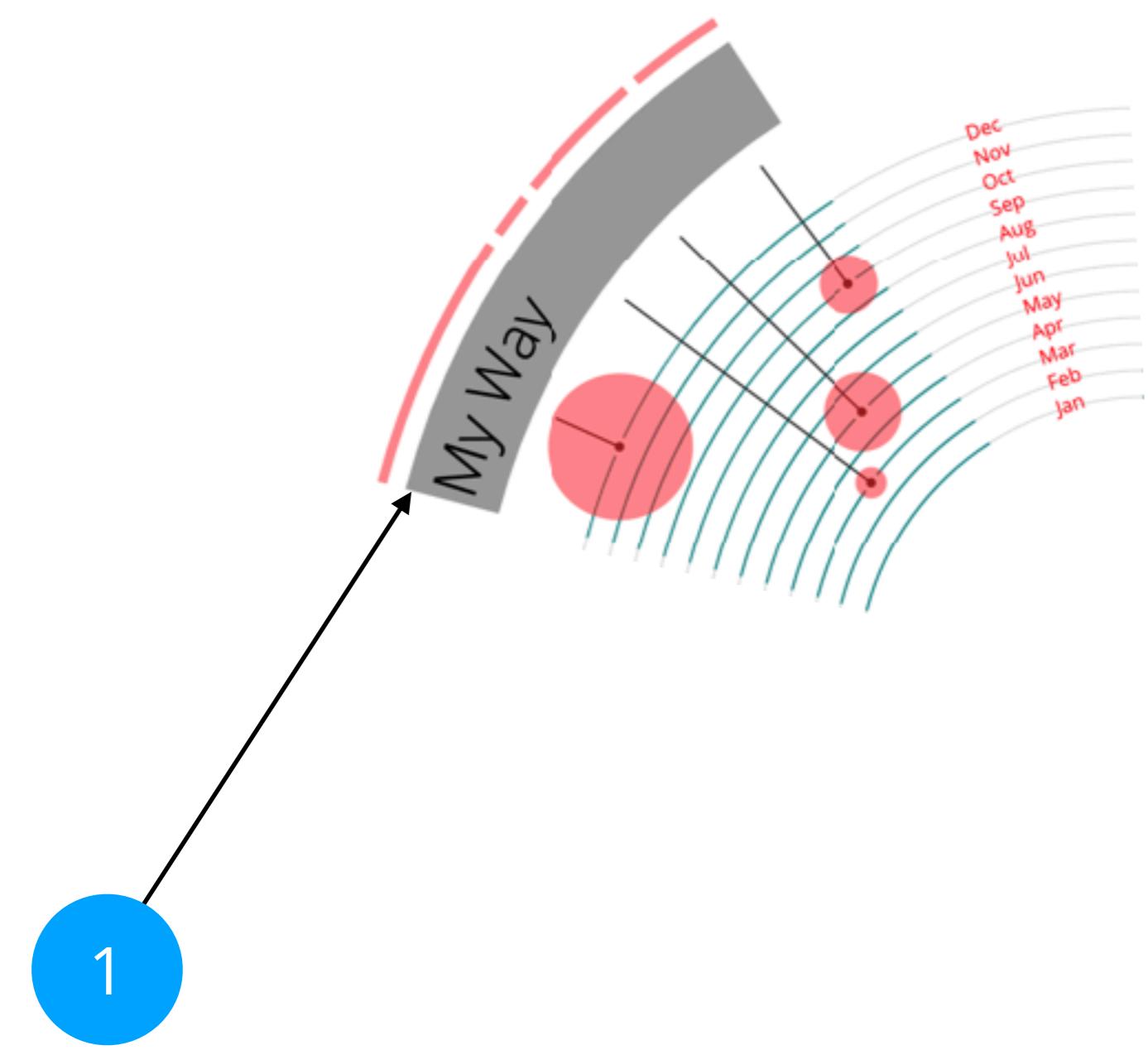
2017

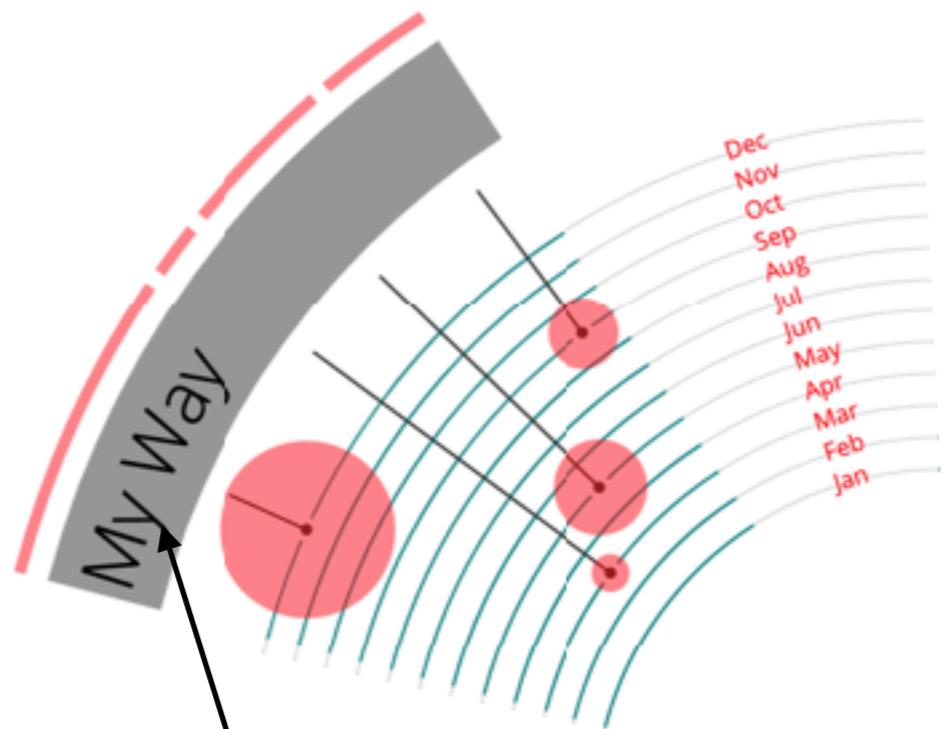
THE INTERPRETERS
A YEAR IN



The visual elements?

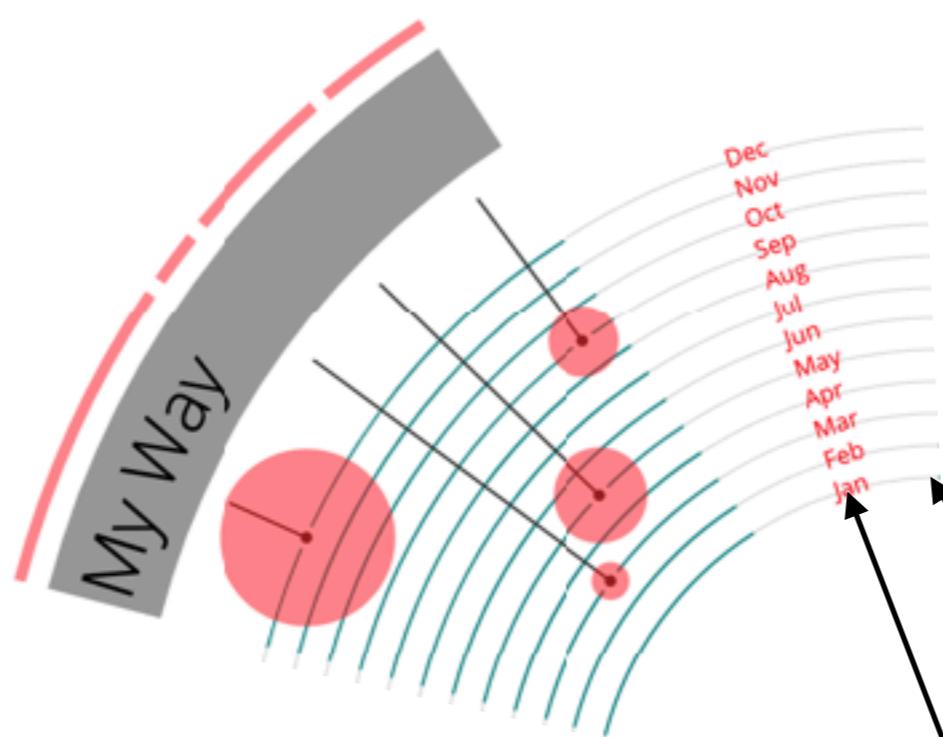






1

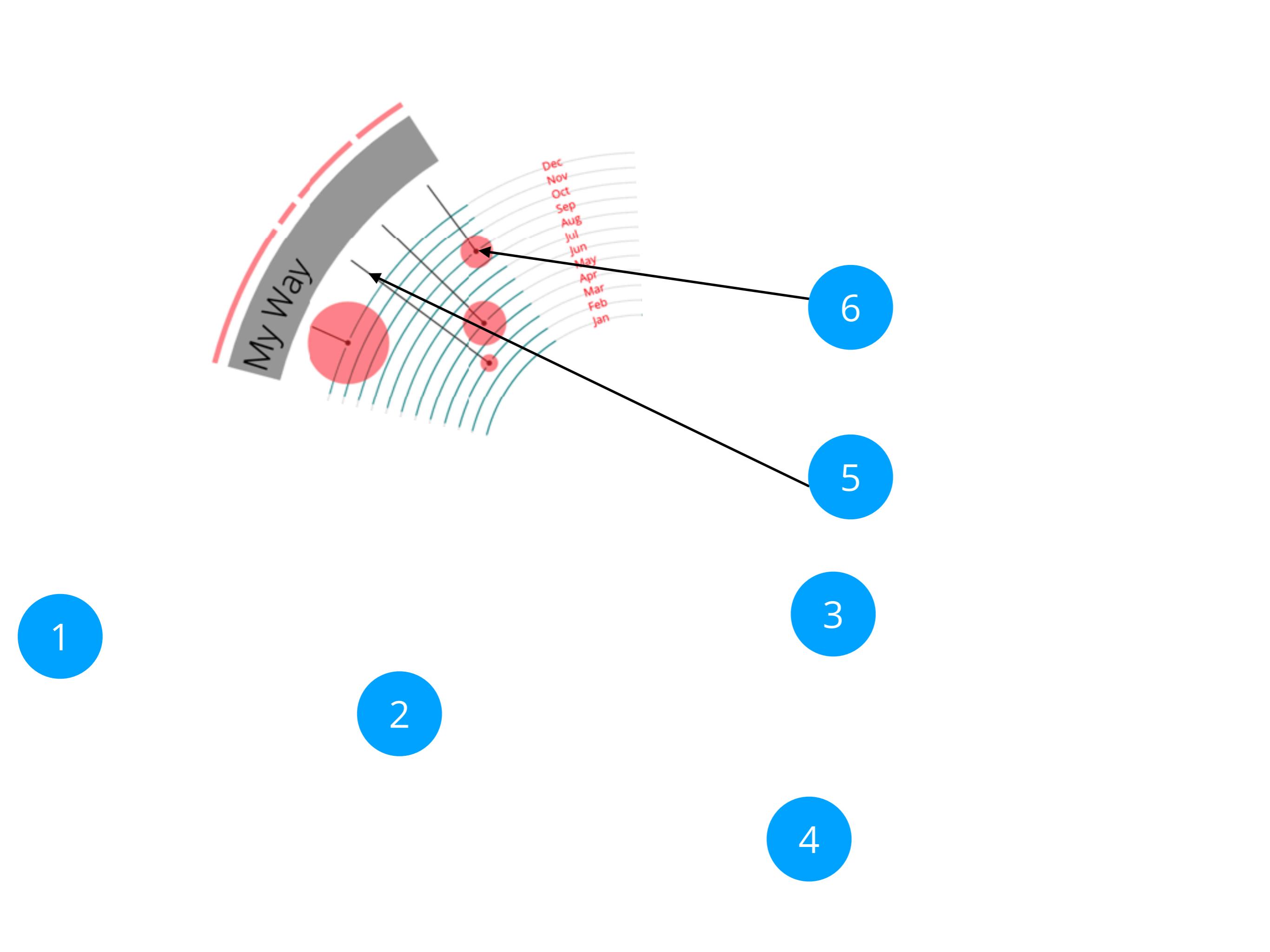
2

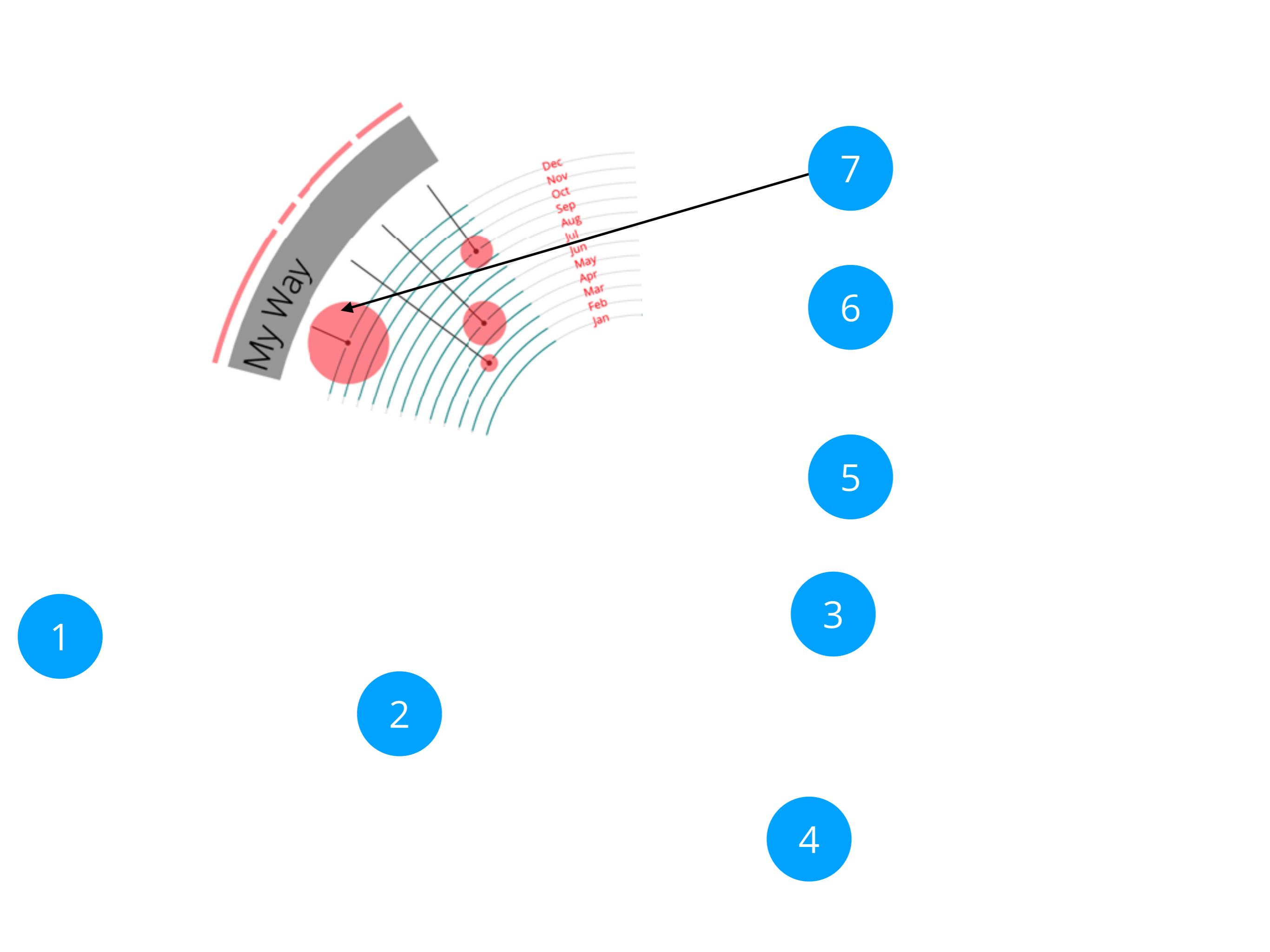


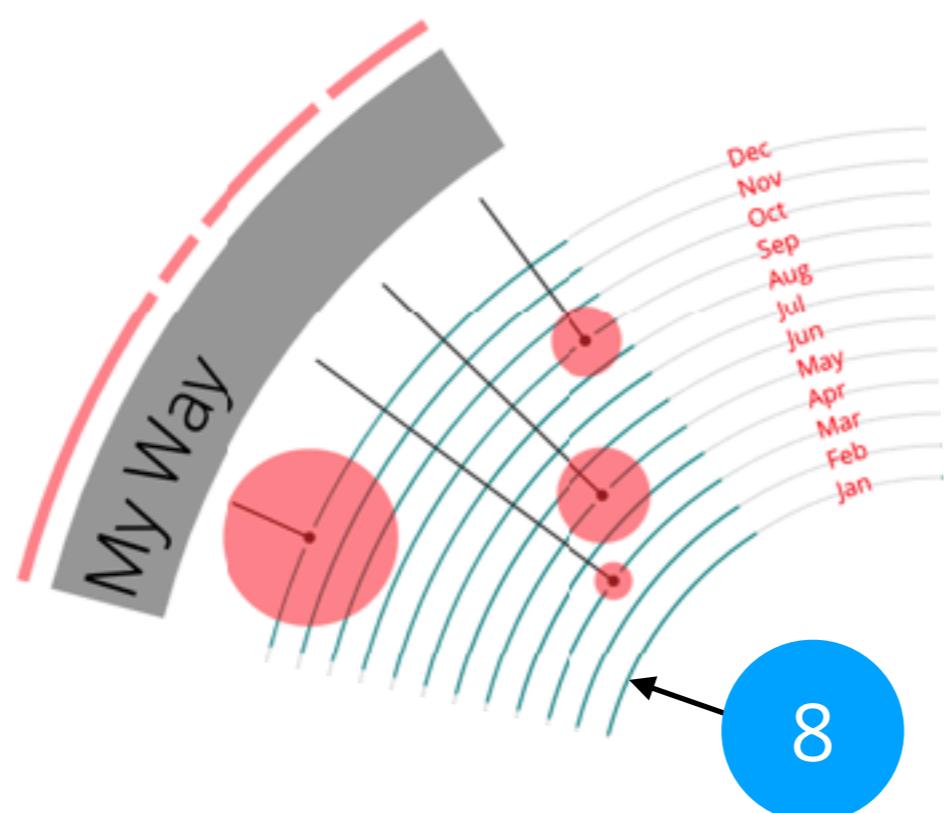
1

2

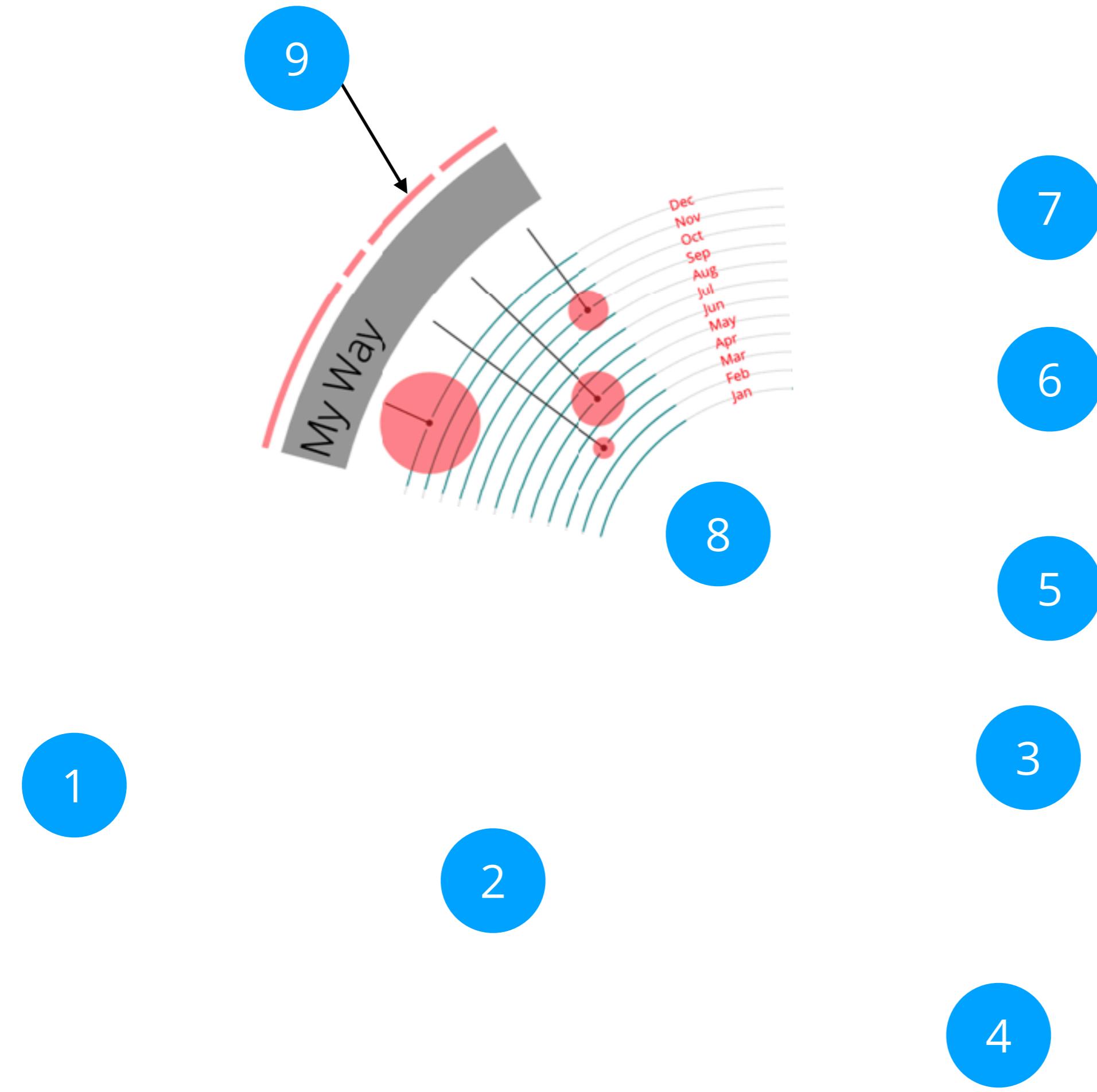
3
4





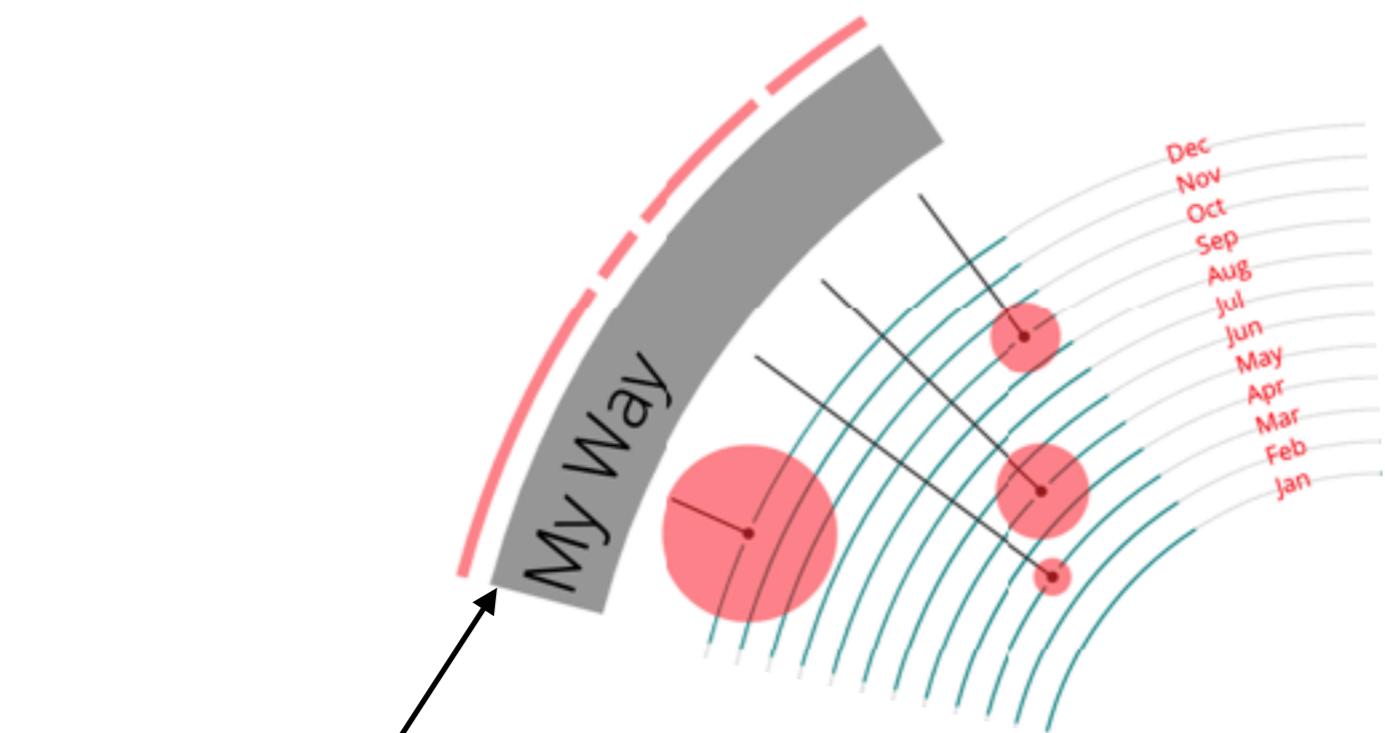


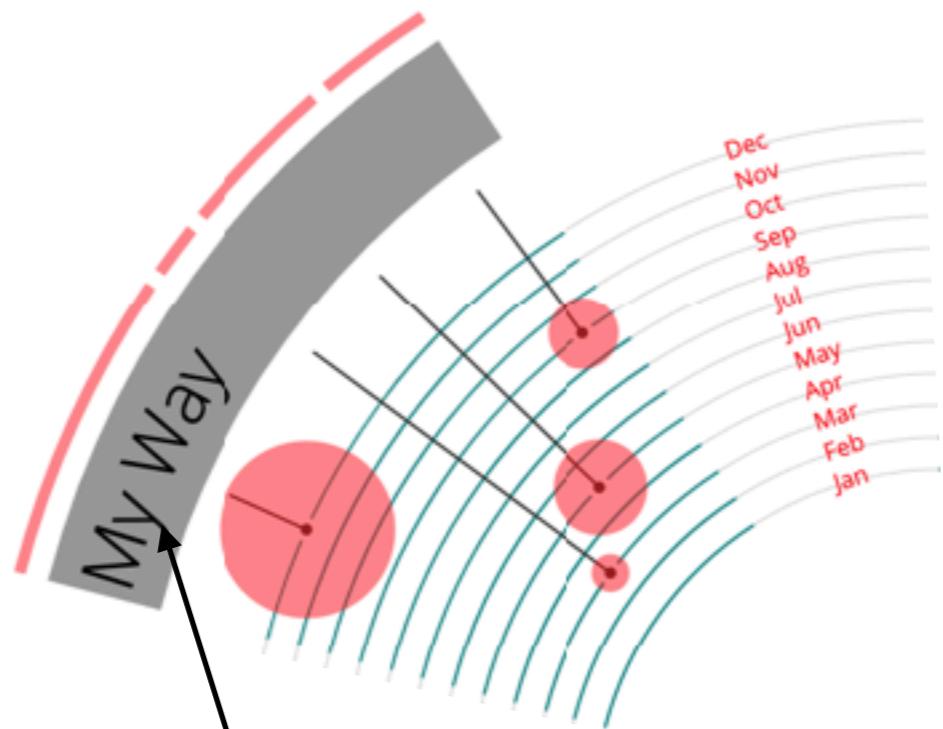
- 1
- 2
- 3
- 4
- 5
- 6
- 7



The code?

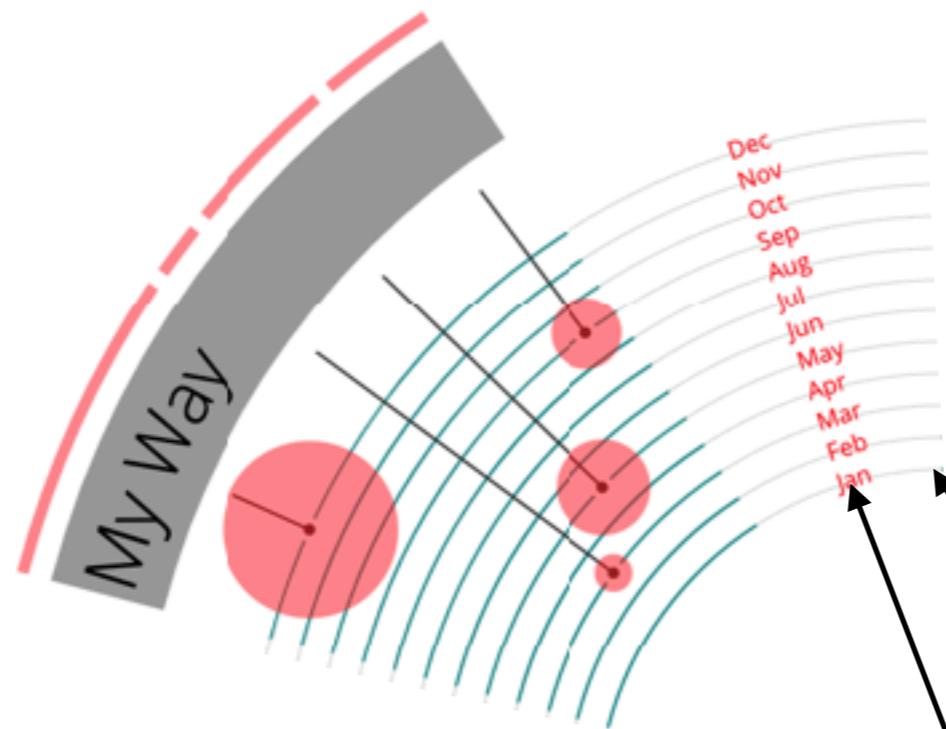
1
`gtap.$arcs(clientAllocationData, [
 gtap.$xy([x, y]),
 gtap.$arcRadius(clientAllocationRadius),
 gtap.$arcSpread(300),
 gtap.$arcSpanOffset(3),
 gtap.$strokeLineCap(gtap.LineCap.BUTT),
 gtap.$style(`...`),
 gtap.$each((v) => {
 ...
 }),
]);`





2

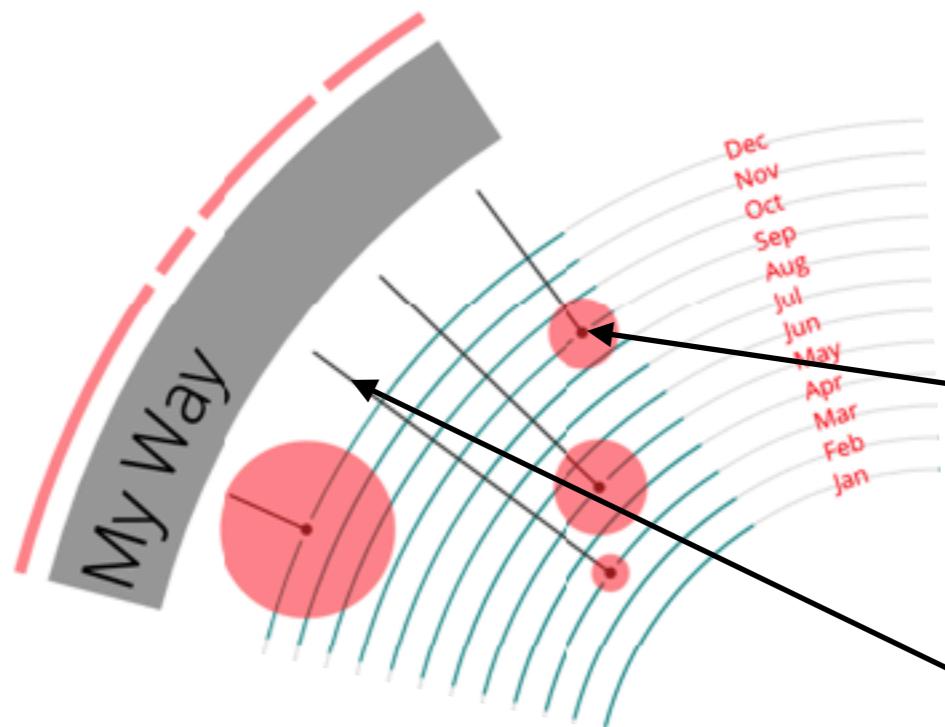
```
gtap.$textAlongPaths(visualPath, textData, [
    gtap.$textVerticalAlign(...MIDDLE),
    gtap.$each((v, index) => {
        ...font-size
    }),
    gtap.$wrapTextAlongPath(),
], ) ,
```



3

```
gtap.$ellipses(gtap.$dataWithIncrement(12, 1), [  
    gtap.$xy([x, y]),  
    gtap.$height(60),  
    gtap.$heightOffset(5.5),  
    gtap.$style(`...`),  
    gtap.$each((v, index) => {  
        const angle = -18;  
        const text = gtap.text(v.$parentElm);  
        text.$toPoint(...);  
        text.$style("... fill:red");  
        text.$text(...$shortMonthNames(index));  
        text.$rotateText(angle, ...);  
    }),  
], [
```

4

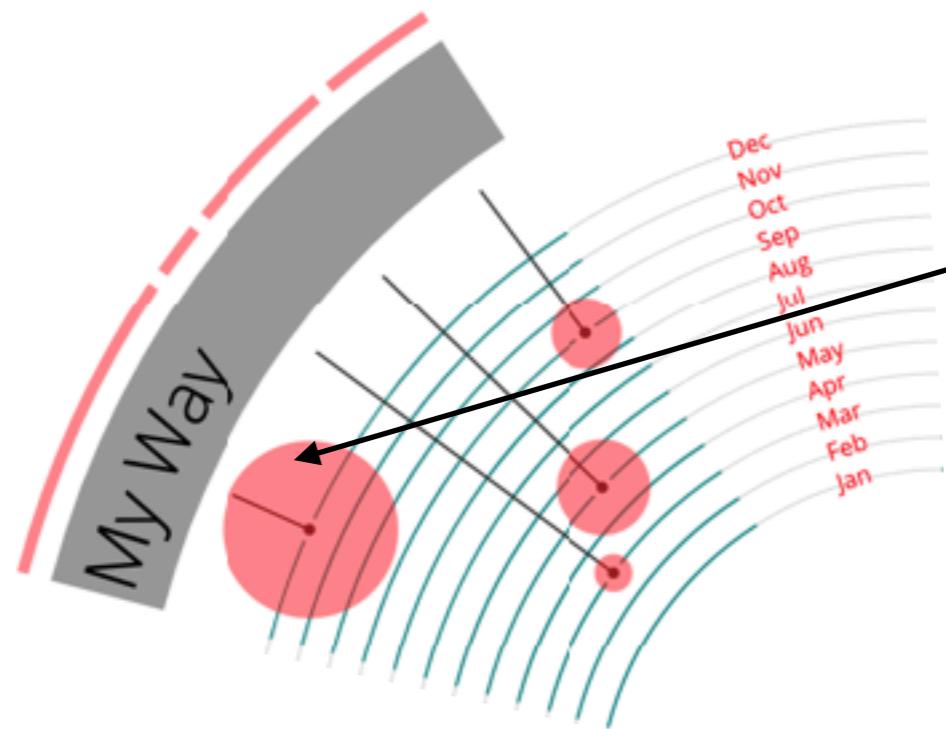


```
gtap.$ellipses(workDistributionData, [
    gtap.$size(1.5, 1.5),
    gtap.$each((v, index) => v.$toPoint(...)),
    gtap.$style(`...`),
]) ,
```

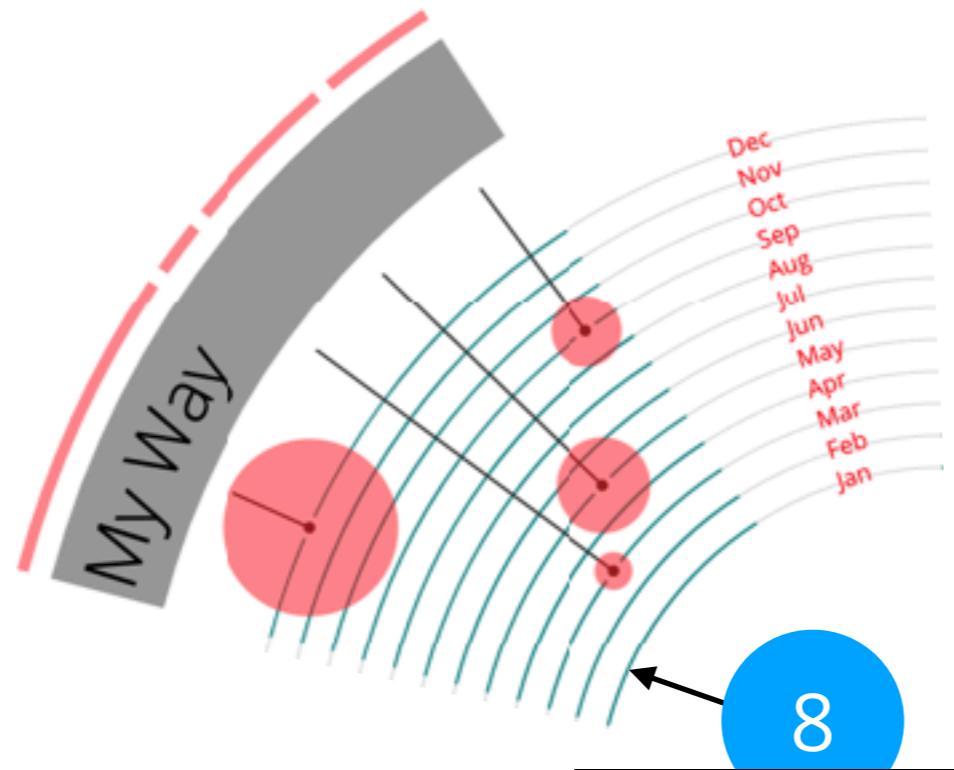
```
gtap.$wrappedShape(gtap.line, ...Data, [
    gtap.$each((v, index) => {
        ...
        v.$point1(point1);
        v.$point2(point2);
        v.$style(`...`);
    }),
]) ,
```

6

5

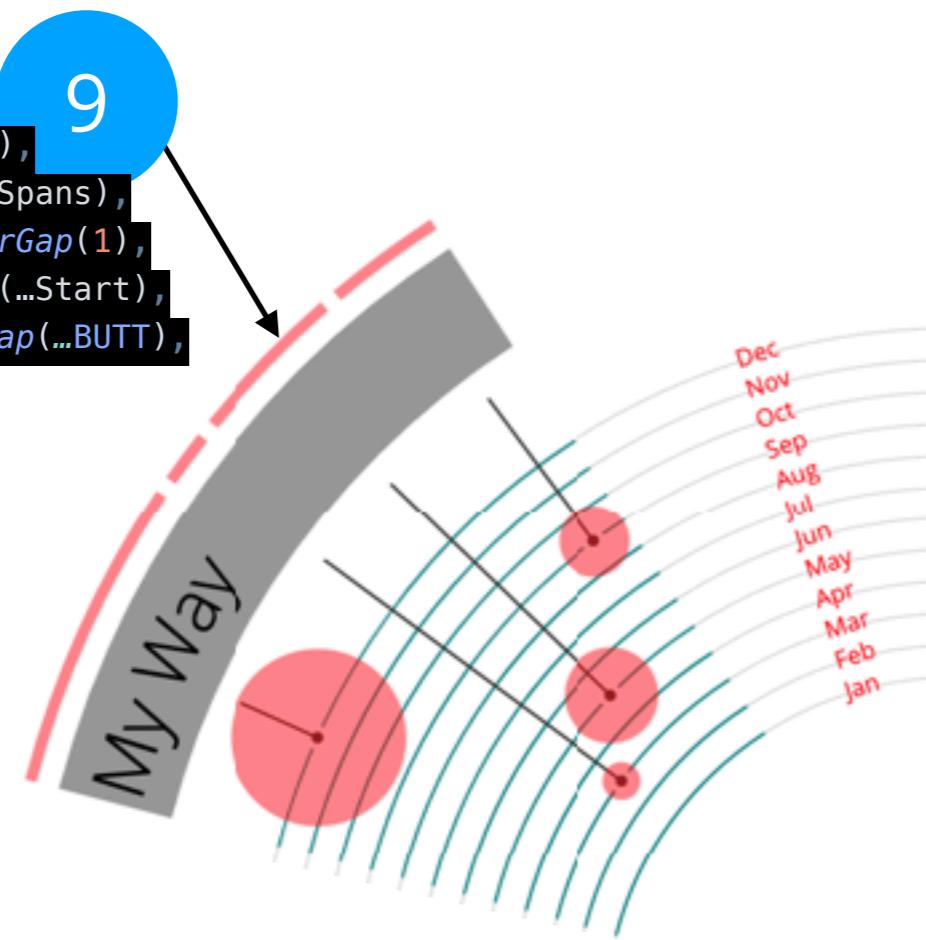


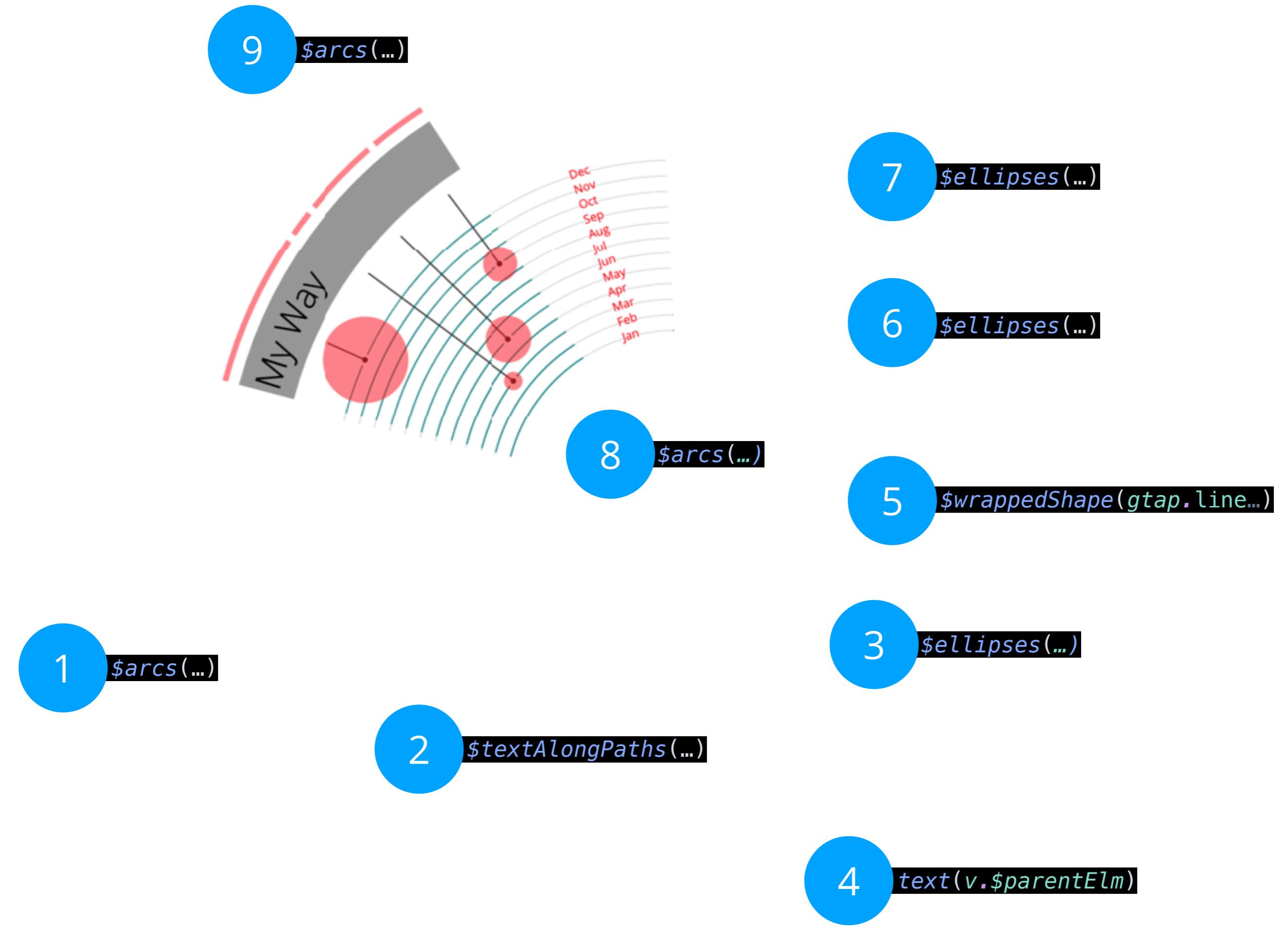
```
gtap.$ellipses(workDistributionData, [
  gtap.$size((index) => {...}),
  gtap.$each((v, index) => v.$toPoint(...)),
  gtap.$style(`...`),
]) ,
```



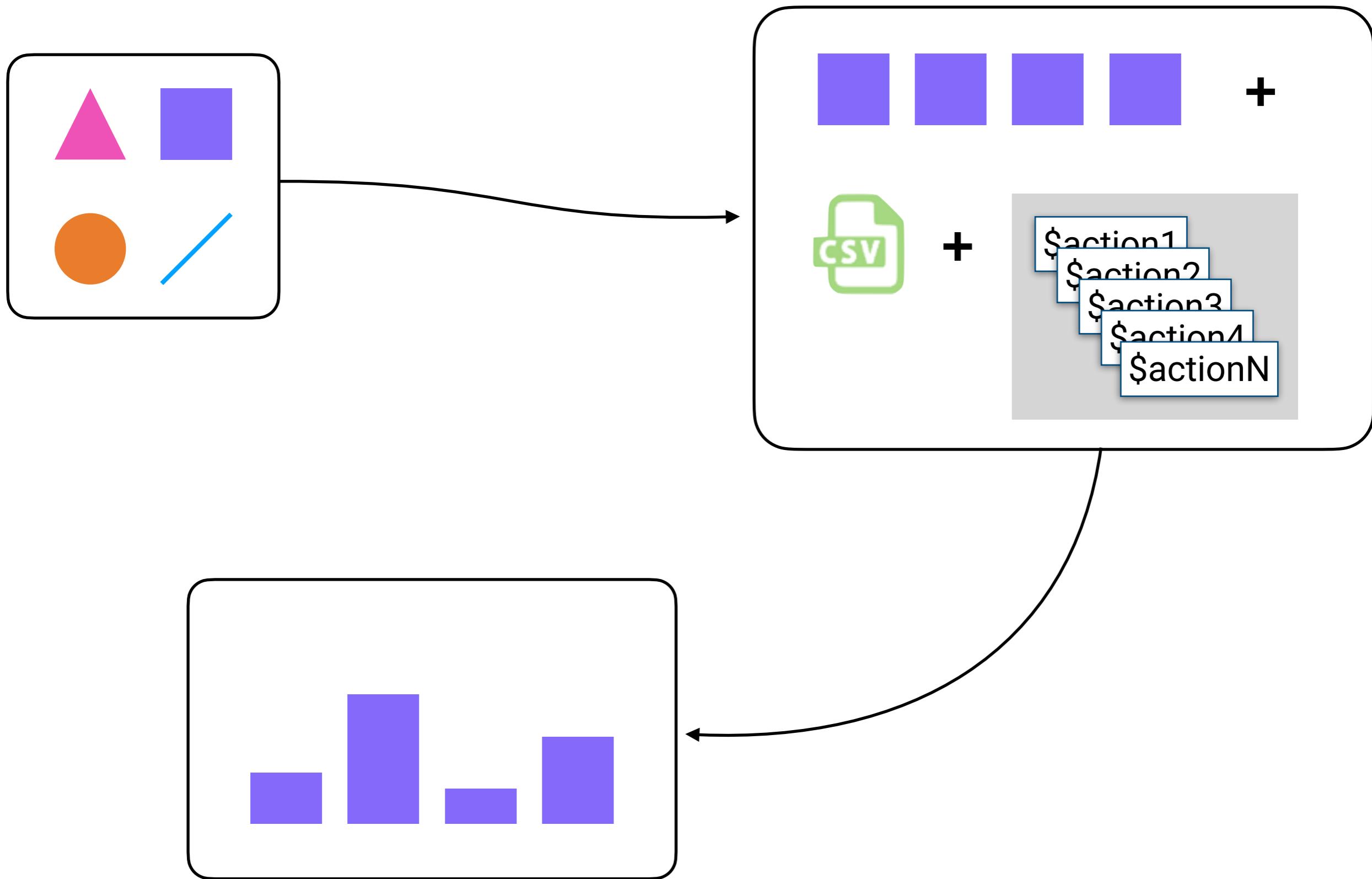
```
gtap.$arcs(gtap.$data(...)), [  
    gtap.$xy([x, y]), [  
        gtap.$arcSpanUnit(..Spans),  
        gtap.$arcRadius(60), [  
            gtap.$arcRadiusOffset(5.5),  
            gtap.$arcRotateBy(..Start), [  
                gtap.$style(`stroke: teal;...`),  
            ]), [  
        ]), [  
    ]), [  
]
```

```
gtap.$arcs(...Data, [  
  gtap.$xy([x, y]), 9  
  gtap.$arcRadius(...),  
  gtap.$arcSpread(...Spans),  
  gtap.$arcSpanInterGap(1),  
  gtap.$arcRotateBy(...Start),  
  gtap.$strokeLineCap(...BUTT),  
  gtap.$style(`...`)  
),  
) ,
```

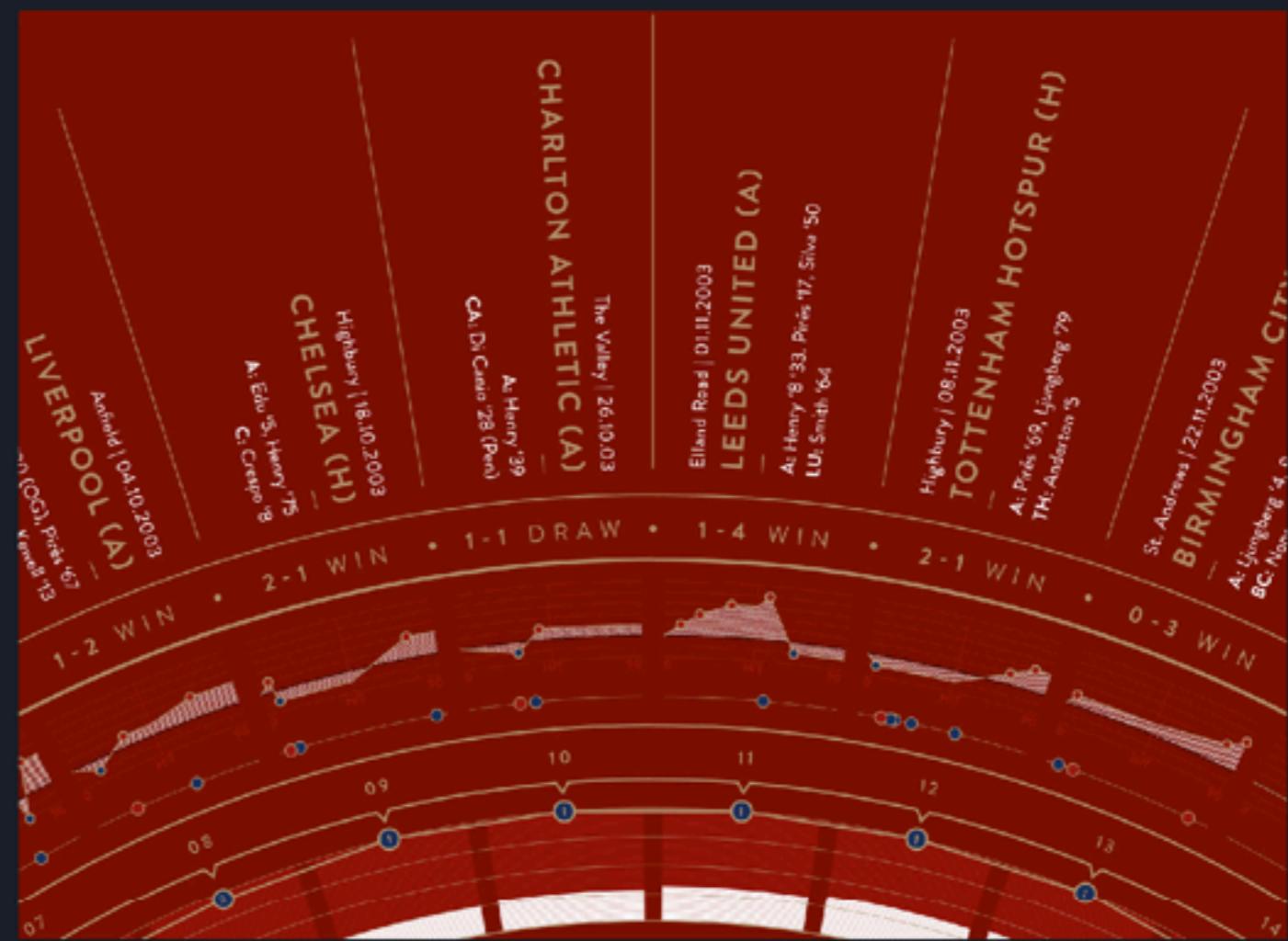
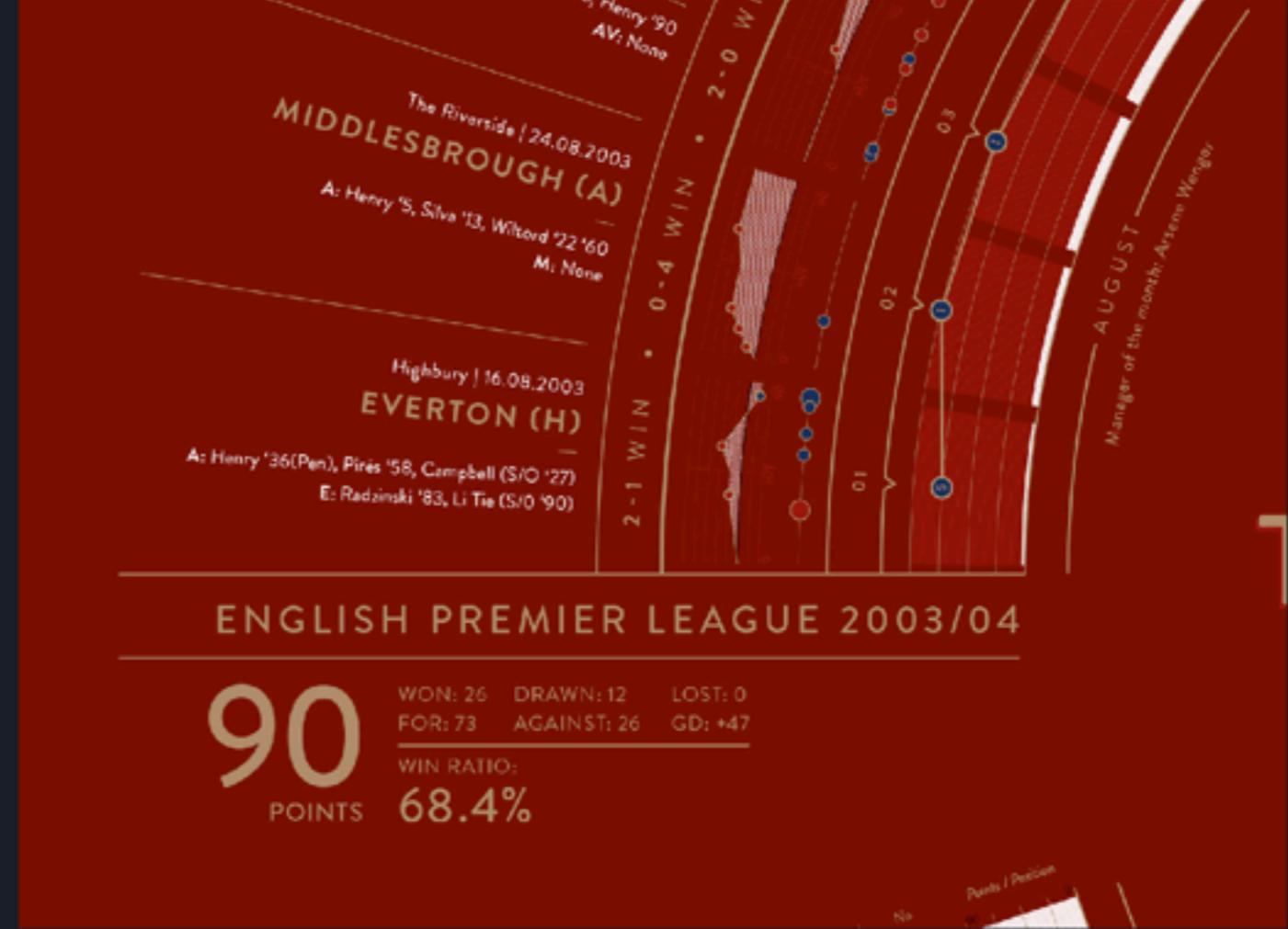




What did we just see?



Limited By Creativity Not Limited By Tool



my **LEARNINGS**
& **FAILURES**

My Approach

idea validation

The first step is our new
version which isn't
completely ready by the
way, but well, we're
working on it. We want to
take feedback and
make it better.

good enough...

Good Enough!

Working

Simple

Minimal

Good Enough!

Working

Simple

Minimal

Code builds and runs without major errors
Each build is usable (even if incomplete)



*the important
bit of context*

Good Enough!

Working

Simple

Minimal

Code builds and runs without ***major*** errors

Each build is usable (even if incomplete)

Mindset: Every build should get shit done!

Good Enough!

Working

Simple

Minimal

Experience is designed for others
Complexity is abstracted away in end user
Mindset: Obvious by design.

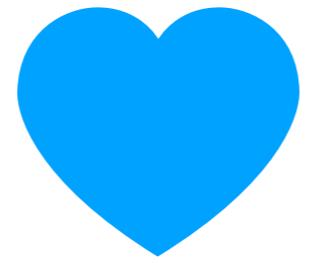
Working
Simple
Minimal

Good Enough!

Don't build Sand Castles. Perfect is the enemy!
Only necessary features
Mindset: 80% of features aren't used. **Build the 20%.**

VERSION CONTROL EVERYTHING

No changes	Commit Changes	
No changes	Commit Changes	
Update front-end with trend-field-values	haroldcampbell	master
Extract the options panels into their own folder	haroldcampbell	
Move sidepanel to its own folder	haroldcampbell	
[bug] Ensure on load that the sidepanel options are correctly loaded	haroldcampbell	
Add ability to deselect a widget when it is selected	haroldcampbell	
[chore] Code clean-up	haroldcampbell	
Make the measure options interactive	haroldcampbell	origin/master
Add TrendOptionsSidePanel to SidePanel	haroldcampbell	
Update sidepanel to show widget header	haroldcampbell	
Add ability to select a widget and notify others	haroldcampbell	
Refactored the heavy logic from the entire sidepanel	haroldcampbell	
Update the project measures cell when a measure is selected	haroldcampbell	
Load the project measures with their aggregate values	haroldcampbell	
Fetch the measures with their aggregate values	haroldcampbell	
Add ability to calculate the aggregate values	haroldcampbell	
No changes	Commit Changes	
Persist layout attributes to the database	haroldcampbell	6
Add input fields to change the size and position of the side panel	haroldcampbell	19:44
Renamed and restructured the side panels	haroldcampbell	16:26
Make escape key deactivate active widget	haroldcampbell	16:09
[bug] Add fix to allow adding of measures	haroldcampbell	16:09
Add simple bars	haroldcampbell	4
Sync the options side panel buttons to the front-end	haroldcampbell	14:19
Update front-end with trend-field-values	haroldcampbell	10:42
Update server to calc-trend-field-values for trends	haroldcampbell	09:31
Wire front-end to call server api for trends	haroldcampbell	08:08
Simplify the common logic to baseOption	haroldcampbell	07:35
Extract the options panels into their own component	haroldcampbell	07:12
Move sidepanel to its own folder	haroldcampbell	4
[bug] Ensure on load that the sidepanel options are correctly loaded	haroldcampbell	07:06



Types



Harold Campbell

@haroldcampbell

▼

Yep. Seems like moving back to [#TypeScript](#) is in my near future.

8:07 PM · Aug 16, 2020 · Twitter Web App

**JavaScript + Types =
TypeScript**

~~JavaScript~~ + Types =
TypeScript

ES-2015

ES-2016

ES-2017

~~JavaScript + Types =~~
TypeScript

ES-2018

ES-2019

ES-2020

View Go Run Terminal Window Help

optionsSidePanelTrend.tsx — groundtap (Workspace)

ts bars.tsx ts sidePanelControllerView.tsx ts projectMeasureContainerController.tsx ts _defs.ts ts optionsSidePanelTrend.tsx

```
4  import { OptionsSidePanel,
5    TrendFunctionTypes,
6    TrendFunctionTypesNames,
7  } from "../../_defs";
8  import { BaseOptionsSidePanel } from "./baseOptionsSidePanel";
9  import { ProjectMeasureContainerController } from "../projectMeasureContainerController";
10 import { BaseMeasuresOptionsSidePanel } from "./baseMeasuresOptionSidePanel";
11
12 /** @jsx gtap.$jsx */
13
14 type TrendResource = { trendType: TrendFunctionTypesNames; trendName: string };
15
16 You, a few seconds ago | 1 author (You)
17 export class OptionsSidePanelTrend extends BaseMeasuresOptionsSidePanel {
18   private content =
19     <div className='options-side-panel-container trends-options-side-panel-container'></div>;
20
21   readonly sidePanelOption = OptionsSidePanel.TrendOptions;
22
23   initView() {
24     this.initButtons();
25     super.initView();
26   }
27
28   viewContent() {
29     return this.content;
30   }
31
32   initButtons() {
33     const resources: TrendResource[] = [
34       { trendType: TrendFunctionTypes.arcs, trendName: "Arcs" },
35       { trendType: TrendFunctionTypes.bars, trendName: "Bars" },
36       { trendType: TrendFunctionTypes.chords, trendName: "Chords" },
37       { trendType: TrendFunctionTypes.lines, trendName: "Lines" },
38     ];
39
40     resources.forEach((resource) => {
41       const node = this.createNodeFromResource(resource);
42
43       this.nodes[resource.trendType] = node;
44       this.content.appendChild(node);
45       node.onclick = () => {
46         this.changeMeasure(node);
47       };
48     });
49   }
50 }
```

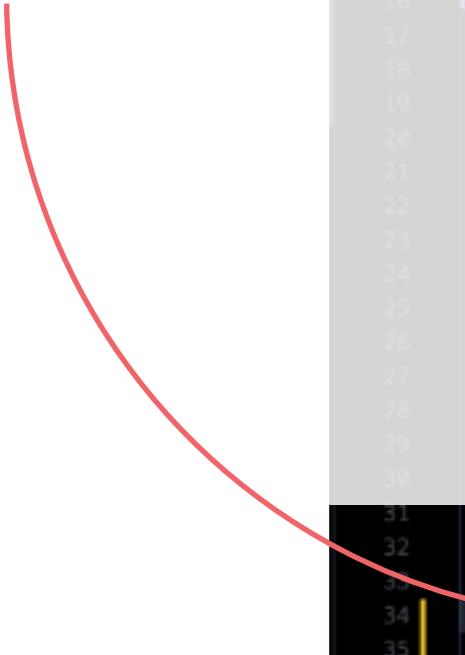
You, a few seconds ago | 1 author (You)

Uncommitted changes

✓ typescriptreact | ✓ optionsSidePanelTrend.tsx

⌚ You, a few seconds ago Ln 34, Col 47 Spaces: 4 UTF-8 LF TypeScript React 3.9.7 ⚡ [off] Prettier: ✓ ⌂ ⌂

This code
is failing



View Go Run Terminal Window Help

optionsSidePanelTrend.tsx — groundtap (Workspace)

ts bars.tsx ts sidePanelControllerView.tsx ts projectMeasureContainerController.tsx ts _defs.ts ts optionsSidePanelTrend.tsx

```
4 |     OptionsSidePanel,
5 |     TrendFunctionTypes,
6 |     TrendFunctionTypesNames,
7 |   } from "../../_defs";
8 |   import { BaseOptionsSidePanel } from "./baseOptionsSidePanel";
9 |   import { ProjectMeasureContainerController } from "../projectMeasureContainerController";
10 |  import { BaseMeasuresOptionsSidePanel } from "./baseMeasuresOptionSidePanel";
11 |
12 |  /** @jsx gtap.$jsx */
13 |
14 |  type TrendResource = { trendType: TrendFunctionTypesNames; trendName: string };
15 |
16 |  You, a few seconds ago | 1 author (You)
17 |  export class OptionsSidePanelTrend extends BaseMeasuresOptionsSidePanel {
18 |    private content =
19 |      <div className='options-side-panel-container trends-options-side-panel-container'></div>
20 |
21 |    readonly sidePanelOption = OptionsSidePanel.TrendOptions;
22 |
23 |    initView() {
24 |      this.initButtons();
25 |      super.initView();
26 |    }
27 |
28 |    viewContent() {
29 |      return this.content;
30 |    }
31 |
32 |    initButtons() {
33 |      const resources: TrendResource[] = [
34 |        { trendType: TrendFunctionTypes.arcs, trendName: "Arcs" }, You, a few seconds ago * Uncommitted changes
35 |        { trendType: TrendFunctionTypes.bars, trendName: "Bars" },
36 |        { trendType: TrendFunctionTypes.chords, trendName: "Chords" },
37 |        { trendType: TrendFunctionTypes.lines, trendName: "Lines" },
38 |      ];
39 |
40 |      resources.forEach((resource) => {
41 |        const node = this.createNodeFromResource(resource);
42 |
43 |        this.nodes[resource.trendType] = node;
44 |        this.content.appendChild(node);
45 |        node.onclick = () => {
46 |          this.changeMeasure(node);
47 |        };
48 |      });
49 |
50 |    }
51 |
52 |  }
53 |
54 |  You, a few seconds ago | 1 author (You)
55 |  export default OptionsSidePanelTrend;
```

✓ typescriptreact ✓ optionsSidePanelTrend.tsx

You, a few seconds ago | Ln 34, Col 47 | Spaces: 4 | UTF-8 | LF | TypeScript React | 3.9.7 | ⚡ [off] | Prettier: ✓ | ⚡ | ⚡

View Go Run Terminal Window Help

optionsSidePanelTrend.tsx — groundlap (Workspace)

```
1 bars.tsx      10 sidePanelControllerView.tsx    15 projectMeasureContainerController.tsx    15 _defs.ts      15 optionsSidePanelTrend.tsx ×  16 40 50 75 100% Mon 7 Sep 7:09 AM
```

```
1 | import { OptionsSidePanel } from "groundlap";
2 | import { TrendFunctionTypes, TrendFunctionTypesNames, TrendFunctionTypesNamesString } from "./../../_defs";
3 | import { BaseOptionsSidePanel } from "./baseOptionsSidePanel";
4 | import { ProjectMeasureContainerController } from "./projectMeasureContainerController";
5 | import { BaseMeasuresOptionsSidePanel } from "./baseMeasuresOptionsSidePanel";
6 |
7 | // tsx imports
8 |
9 | type TrendResource = { trendType: TrendFunctionTypesNames; trendName: string };
10 |
11 | class OptionsSidePanelTrend extends BaseMeasuresOptionsSidePanel {
12 |     private content: ReactElement;
13 |     readonly sidePanelOption = OptionsSidePanel.TrendOptions;
14 |
15 |     constructor(props: OptionsSidePanelProps) {
16 |         super(props);
17 |         this.initButtons();
18 |     }
19 |
20 |     render() {
21 |         return (
22 |             <div className="options-side-panel-container__trends-options-side-panel-container">
23 |                 <div>
24 |                     <div>
25 |                         <div>
26 |                             <div>
27 |                                 <div>
28 |                                     <div>
29 |                                         <div>
30 |                                             <div>
31 |                                                 <div>
32 |                                                     <div>
33 |                                                         <div>
34 |                                                             <div>
35 |                                                                 <div>
36 |                                                                     <div>
37 |                                                                         <div>
38 |                                                                             <div>
39 | 
```

(property) trendType: "arcs" | "bars" | "chords" | "lines"

Type 'TrendFunctionTypes.arcs' is not assignable to type '"arcs" | "bars" | "chords" | "lines". ts(2322)

optionsSidePanelTrend.tsx(14, 24): The expected type comes from property 'trendType' which is declared here on type 'TrendResource'

initButton

const **Peek Problem (xF8)** No quick fixes available

{ trendType: TrendFunctionTypes.arcs, trendName: "Arcs" }, { trendType: TrendFunctionTypes.bars, trendName: "Bars" }, { trendType: TrendFunctionTypes.chords, trendName: "Chords" }, { trendType: TrendFunctionTypes.lines, trendName: "Lines" },

```
resources.forEach((resource) => {
    const node = this.createNodeFromResource(resource);

    this.nodes[resource.trendType] = node;
    this.content.appendChild(node);
    node.onclick = () => {
        this.changeMeasure(node);
    };
});
```

View Go Run Terminal Window Help

optionsSidePanelTrend.tsx — groundtap (Workspace)

rs bars.tsx ts sidePanelControllerView.tsx ts projectMeasureContainerController.tsx ts _defs.ts ts optionsSidePanelTrend.tsx

```
4 |     OptionsSidePanel,
5 |     TrendFunctionTypes,
6 |     TrendFunctionTypesNames,
7 |   } from "../../_defs";
8 |   import { BaseOptionsSidePanel } from "./baseOptionsSidePanel";
9 |   import { ProjectMeasureContainerController } from "../projectMeasureContainerController";
10 |  import { BaseMeasuresOptionsSidePanel } from "./baseMeasuresOptionSidePanel";
11 |
12 |  /** @jsx grap.$jsx */
13 |
14 |  type TrendResource = { trendType: TrendFunctionTypesNames; trendName: string };
15 |
16 |  You, a few seconds ago | 1 author (you)
17 |  export class OptionsSidePanelTrend extends BaseMeasuresOptionsSidePanel {
18 |    private content =
19 |      <div className="options-side-panel-container trends-options-side-panel-container"></div>
20 |
21 |    readonly sidePanelOption = OptionsSidePanel.TrendOptions;
22 |
23 |    initView() {
24 |      this.initButtons();
25 |      super.initView();
26 |    }
27 |
28 |    viewContent() {
29 |      return this.content;
30 |    }
31 |
32 |    initButtons() {
33 |      const resources: TrendResource[] = [
34 |        { trendType: TrendFunctionTypes.arcs, trendName: "Arcs" },
35 |        { trendType: TrendFunctionTypes.bars, trendName: "Bars" },
36 |        { trendType: TrendFunctionTypes.chords, trendName: "Chords" },
37 |        { trendType: TrendFunctionTypes.lines, trendName: "Lines" },
38 |      ];
39 |
40 |      resources.forEach((resource) => {
41 |        const node = this.createNodeFromResource(resource);
42 |
43 |        this.nodes[resource.trendType] = node;
44 |        this.content.appendChild(node);
45 |        node.onclick = () => {
46 |          this.changeMeasure(node);
47 |        };
48 |      });
49 |
50 |    }
51 |
52 |  }
53 |
54 |  You, a few seconds ago | 1 author (you) * Uncommitted changes
```

You, a few seconds ago | 1 author (you) * Uncommitted changes

You, a few seconds ago | Ln 34, Col 47 | Spaces: 4 | UTF-8 | LF | TypeScript React | 3.9.7 | ☰ [off] | Prettier: ✓ | ⚡ | ⌂

Life saving
Type
Annotation

View Go Run Terminal Window Help

optionsSidePanelTrend.tsx — groundlap (Workspace)

bars.tsx sidePanelControllerView.tsx projectMeasureContainerController.tsx _defs.ts optionsSidePanelTrend.tsx

```
1  import { OptionsSidePanel } from "./OptionsSidePanel";
2  import { TrendFunctionTypes, TrendFunctionTypesNames, TrendResource } from "../../_defs";
3  import { BaseOptionsSidePanel } from "./baseOptionsSidePanel";
4  import { ProjectMeasureContainerController } from "../../projectMeasureContainerController";
5  import { BaseMeasuresOptionsSidePanel } from "../../baseMeasuresOptionsSidePanel";
6
7  export interface Trend {
8    trendType: TrendFunctionTypes;
9    trendName: string;
10   arcs: boolean;
11 }
12
13 type TrendResource = { trendType: TrendFunctionTypesNames; trendName: string };
14
15 export class OptionsSidePanelTrend extends BaseMeasuresOptionsSidePanel {
16   private content: React.ReactNode;
17
18   constructor() {
19     super();
20     this.initButtons();
21   }
22
23   get content(): React.ReactNode {
24     return this.content;
25   }
26
27   private initButtons() {
28     const resources: TrendResource[] = [
29       { trendType: TrendFunctionTypes.bars, trendName: "Bars" },
30       { trendType: TrendFunctionTypes.chords, trendName: "Chords" },
31       { trendType: TrendFunctionTypes.lines, trendName: "Lines" },
32       { trendType: TrendFunctionTypes.arc, trendName: "Arcs" }
33     ];
34
35     resources.forEach((resource) => {
36       const node = this.createNodeFromResource(resource);
37
38       this.nodes[resource.trendType] = node;
39       this.content.appendChild(node);
40       node.onclick = () => {
41         this.changeMeasure(node);
42       };
43     });
44   }
45 }
```

Life saving
Type
Annotation



arcs = "arc"

#NoFrontEndLibraries

ES6+

React
Angular
VueJS

TypeScript

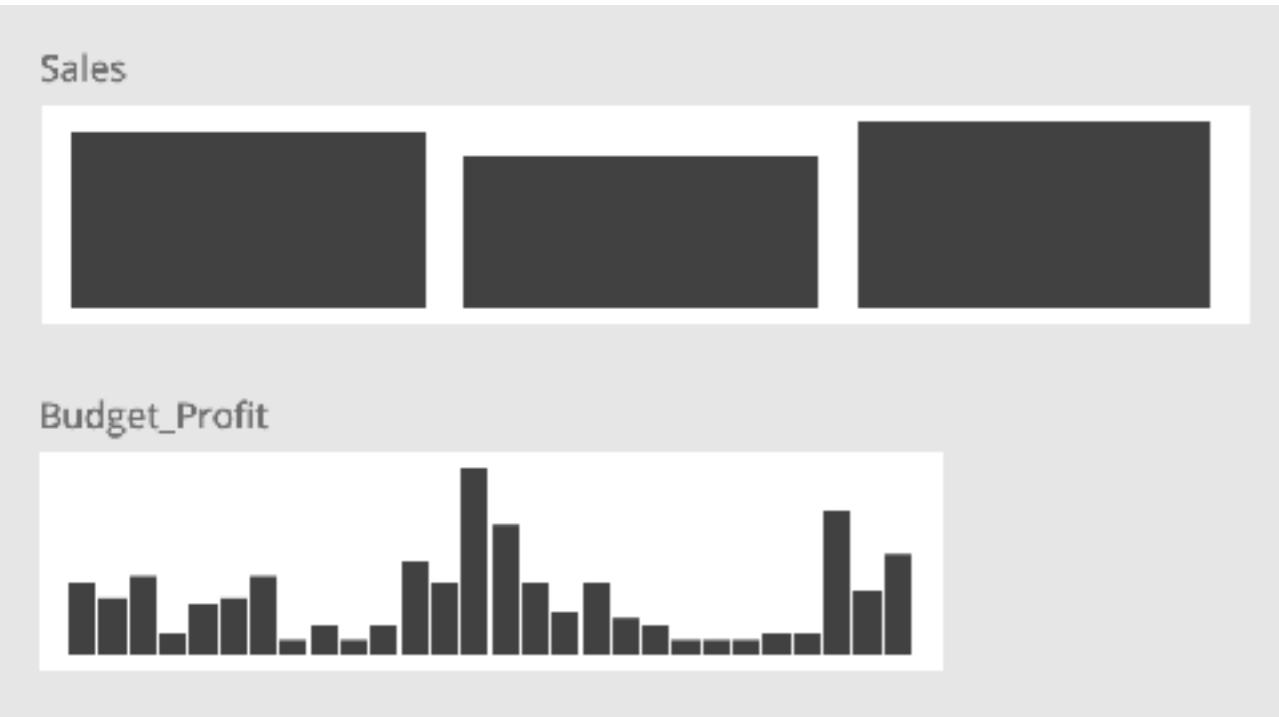


ES6+



ESXXXX allows us to do
more things with fewer
libraries!

ES6: Arrow Functions

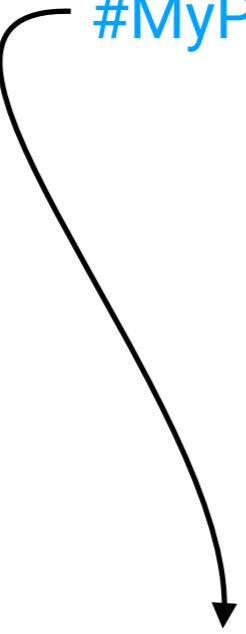


```
// Client facing code
```

```
gtap.$each(v => v.$rxy(2, 2)), // Add rounded corners
```

#MyPerspective

The future of the Web



~~React~~
~~Angular~~
~~VueJS~~
ESXXXX + Types

ES6: Front-end views

```
import {  
  $jsx,  
} from "../../www/assets/dist/gtap"  
  
/** @jsx $jsx */  
  
export function sketchIntro(data) {  
  const content = (  
    <section className="sketch-intro">  
      <div className="sketch-title">{data.sketchTitle}</div>  
      <div className="sketch-desc">{data.sketchDesc}</div>  
    </section>);  
  
  return content;  
}
```

ES6: Front-end views

```
export function $jsx(tag, properties, ...children) {
  const node = $node("http://www.w3.org/1999/xhtml", tag, null);

  if (utils.$isTruthy(properties)) {
    processNodeProperty(node, properties);
  }

  if (utils.$isTruthy(children)) {
    children.forEach(child => processChildNode(node, child));
  }

  return node;
}
```

FRUSTRATIONS

- It's still JavaScript™
- Numbers and precisions
- Browser Inconsistencies
- Getting started and scaling

Library Structure

Tips for rolling your own library

Make it Usable.

Is it possible? **Usable**.

Is there a better UX? **Reusable**.

Tips for rolling your own library

Examples

Text

Bars

Ellipses

Arcs

Segments

Chords

Polygons

Radial Lines

Advanced

Bar Example 1: Data-dri



The height of the bars is based on the `$maxHeight(...)`
each bar is defined by `$maxHeight(...)`
the bars is determined by `$yOffset(...)`

```
container("bar-1", _ => {
  _.addVisuals([
    bars(barsData, [$maxHeight])
  ])
  $id("example-bar-1")
})
```

Tips for rolling your own library

Make it Usable.

Is it possible? **Usable.**

Is there a better UX? **Reusable.**

```
Code File Edit Selection View Go Debug Terminal Window Help
examples-bar.js
JS examples-bar.js × index.html
EXPLORER JS examples-bar.js × index.html
GROUNDTAP JS examples-bar.js × index.html
ancui-dom.js M JS examples-bar.js × index.html
ancui-pipeline.js M JS examples-bar.js × index.html
ancui.js JS examples-bar.js × index.html
utils.js JS examples-bar.js × index.html
node_modules JS examples-bar.js × index.html
tests JS examples-bar.js × index.html
www JS examples-bar.js × index.html
assets JS examples-bar.js × index.html
examples JS examples-bar.js × index.html
dist JS examples-bar.js × index.html
src JS examples-bar.js × index.html
examples-advanced.js JS examples-bar.js × index.html
examples-arc.js JS examples-bar.js × index.html
examples-bar.js JS examples-bar.js × index.html
examples-chord.js JS examples-bar.js × index.html
examples-ellipse.js JS examples-bar.js × index.html
examples-polygon.js JS examples-bar.js × index.html
examples-radial-lines.js JS examples-bar.js × index.html
examples-segment.js JS examples-bar.js × index.html
examples-text.js JS examples-bar.js × index.html
index.html JS examples-bar.js × index.html
# styles.css JS examples-bar.js × index.html
# visuals.css JS examples-bar.js × index.html
experimental JS examples-bar.js × index.html
index.html JS examples-bar.js × index.html
.babelrc JS examples-bar.js × index.html
.eslintrc.js JS examples-bar.js × index.html
...
OUTLINE JS examples-bar.js × index.html
master* 0↓ 7↑ 0△ 28 ① 4 gitd: Express Off ✓ javascript | ✓ examples-bar.js
3 let barsData = gtap.$data([50, 20, 30, 10,
4
5 let ctx = gtap.container("bar-1", gtap.$id("bar-1"))
gtap.renderVisuals(ctx, [
6
7   gtap.$bars(barsData, [
8     gtap.$maxHeight(50),
9     gtap.$animate(.8, "ease-in-out", gtap.$x(200), gtap.$yOffset(30)),
10    gtap.$animate(1.2, "ease-in-out", gtap.$x(200), gtap.$yOffset(30)),
11    gtap.$animate(1.0, "ease-out", gtap.$x(200), gtap.$yOffset(30))
12  ]),
13
14  gtap.$bars(barsData, [
15    gtap.$x(200),
16    gtap.$maxHeight(50),
17    gtap.$yOffset(30),
18    gtap.$width(150)
19  ]),
20]);
21
22 ctx = gtap.container("bar-2", gtap.$id("bar-2"))
gtap.renderVisuals(ctx, [
23
24   gtap.$bars(barsData, [
25     gtap.$maxHeight(50),
26     gtap.$xOffset(30),
27     gtap.$animate(0.8, "ease-out", gtap.$x(200), gtap.$yOffset(30))
28  ]),
29
30
31  gtap.$bars(barsData, [
32    gtap.$x(200),
33    gtap.$maxHeight(50),
34    gtap.$xOffset(30),
35    gtap.$width(150),
36    gtap.$yOffset(30)
37  ]),
38]);
```

Tips for rolling your own library

[SOLID] is your friend.

Single Responsibility

Open-closed

Liskov substitution

Interface Segregation

Dependency Inversion

Tips for rolling your own library

[SOLID] is your friend.

Single Responsibility

Open-closed

Liskov substitution

Interface Segregation

Dependency Inversion

Single Responsibility

Open-closed

Liskov substitution

Interface Segregation

Dependency Inversion

...single responsibility per unit of code

many client-specific interfaces are better than one general-purpose interface

Tips for rolling your own library

Future Proofing [🤦]

Tips for rolling your own library

Future Proofing [🤦]

Refactor for **Simplicity & Readability**

Adopt **FP Patterns** when possible

Prefer **Declarative** over Imperative

Be **Opinionated**, have intelligent defaults

Have a bias for good **Test** hygiene

Adopt **FP Patterns**
when possible.



Pure functions

Immutable data

Avoid shared state

Separation between data and behaviour

Pure functions

Immutable data

Avoid shared state

Separation between data and behaviour

Adopt **FP Patterns** when possible

Immutability & Pure Functions

```
const ctx = gtap.container("chord-1", ...);  
gtap.renderVisuals(ctx, [  
  gtap.$arcs(..., [  
    ...  
  ]),  
]);
```



```
export function renderVisuals(containerContext, visuals) {  
  const newContainerContext =  
    createVisualContexts(containerContext, ...);  
  
  /* Modify newContainerContext */  
  prepareVisuals(newContainerContext);  
  animateVisuals(newContainerContext);  
  
  return newContainerContext;  
}
```

```
export function renderVisuals(containerContext, visuals) {  
  const newContainerContext =  
    createVisualContexts(containerContext, ...);  
  
  /* Modify newContainerContext */  
  prepareVisuals(newContainerContext);  
  animateVisuals(newContainerContext);  
  
  return newContainerContext;  
}
```

```
export function renderVisuals(containerContext, visuals) {  
  const newContainerContext =  
    createVisualContexts(containerContext, ...);  
  
  /* Modify newContainerContext */  
  prepareVisuals(newContainerContext);  
  animateVisuals(newContainerContext);  
  
  return newContainerContext;  
}
```

```
function createVisualContexts(containerContext, visuals) {  
  const newContainerContext =  
    cloneContainerContext(containerContext);  
  
  visuals.forEach(visual => {  
    newContainerContext.visuals.push(  
      createVisualContext(newContainerContext, visual)  
    );  
  });  
  
  return newContainerContext;  
}
```

```
function createVisualContexts(containerContext, visuals) {  
  const newContainerContext =  
    cloneContainerContext(containerContext);  
  
  ...cloning children  
  
  return newContainerContext;  
}
```

```
export function renderVisuals(containerContext, visuals) {  
  const newContainerContext =  
    createVisualContexts(containerContext, ...);  
  
  /* Modify newContainerContext */  
  prepareVisuals(newContainerContext);  
  animateVisuals(newContainerContext);  
  
  return newContainerContext;  
}
```

Prefer **Declarative**,
over Imperative.



DECLARATIVE

Describes **What**

```
students
  .filter(s => isAdult(s))
  .map(s => formatName(s))
```

IMPERATIVE

Describes **How**

```
const filteredStudents = [];
for(i=0; i<students.length; i++) {
  if (students[i].age >= 18) {
    const s = students[i];
    s.name = s.name.toUpperCase();
    filteredStudents.push(s);
  } else {
    // do more things
  }
}
```

IMPERATIVE EXAMPLE

What is the
intent here? _____

```
gtap.$vLine([  
    gtap.$x(30),  
    gtap.$height(100),  
    gtap.$style("stroke:#404040;"),  
    gtap.$each(v => {  
        v.$y(206);  
        v.$y1(v.$y2() - v.$height())  
    })  
])
```

DECLARATIVE **EXAMPLE**

```
gtap.$vLine( [ _____  
  gtap.$x(30),  
  gtap.$height(100),  
  gtap.$style("stroke:#404040;"),  
  gtap.$alignBottom(206) _____ ] )
```

What is the intent here?

DECLARATIVE

Describes **What**

```
gtap.$vLine([  
    gtap.$x(30),  
    gtap.$height(100),  
    gtap.$style("stroke:#404040;"),  
    gtap.$alignBottom(206),  
)
```

 **clearer** intent

IMPERATIVE

Describes **How**

```
gtap.$vLine([  
    gtap.$x(30),  
    gtap.$height(100),  
    gtap.$style("stroke:#404040;"),  
    gtap.$each(v => {  
        v.$y(206);  
        v.$y1(v.$y2() - v.$height())  
    })  
)
```

DECLARATIVE

Promotes Better
Engineering Practices

“Simpler” Logic

Improved Readability

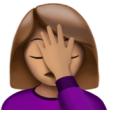
Easier Testing

Faster Debugging

IMPERATIVE

Promotes
“You Touch it/You own it”

Convoluted Spaghetti Logic

Readability feels like 

Testing Gymnastics

Debugging *is a sport*

\$2.80 B

Avengers: Endgame
All-time Highest Grossing Movie



Star Wars: Episode VII - The Force Awakens (2015)

Avengers: Infinity War (2018)

Star Wars: Episode VIII (2017)

Captain America: Civil War (2016)

41%

\$1.15 Billion

48%

\$1.33 Billion

73%

\$2.05 Billion

74%

\$2.07 Billion



arcs
bars
chords
connectingLines
ellipses
empty
hLine
label

labels
lines
polygon
radialLines
segment
textAlongPaths
vLine
wrappedShapes

Visuals are
Declarative

Be **Opinionated**,
have intelligent defaults.



Actions are Declarative

...with predictable Defaults

Core

Actions

x
y
xIncrement
xMargin
xOffset
max
maxWidth
maxX
maxHeight
lambda

Arcs & Ellipses

Actions

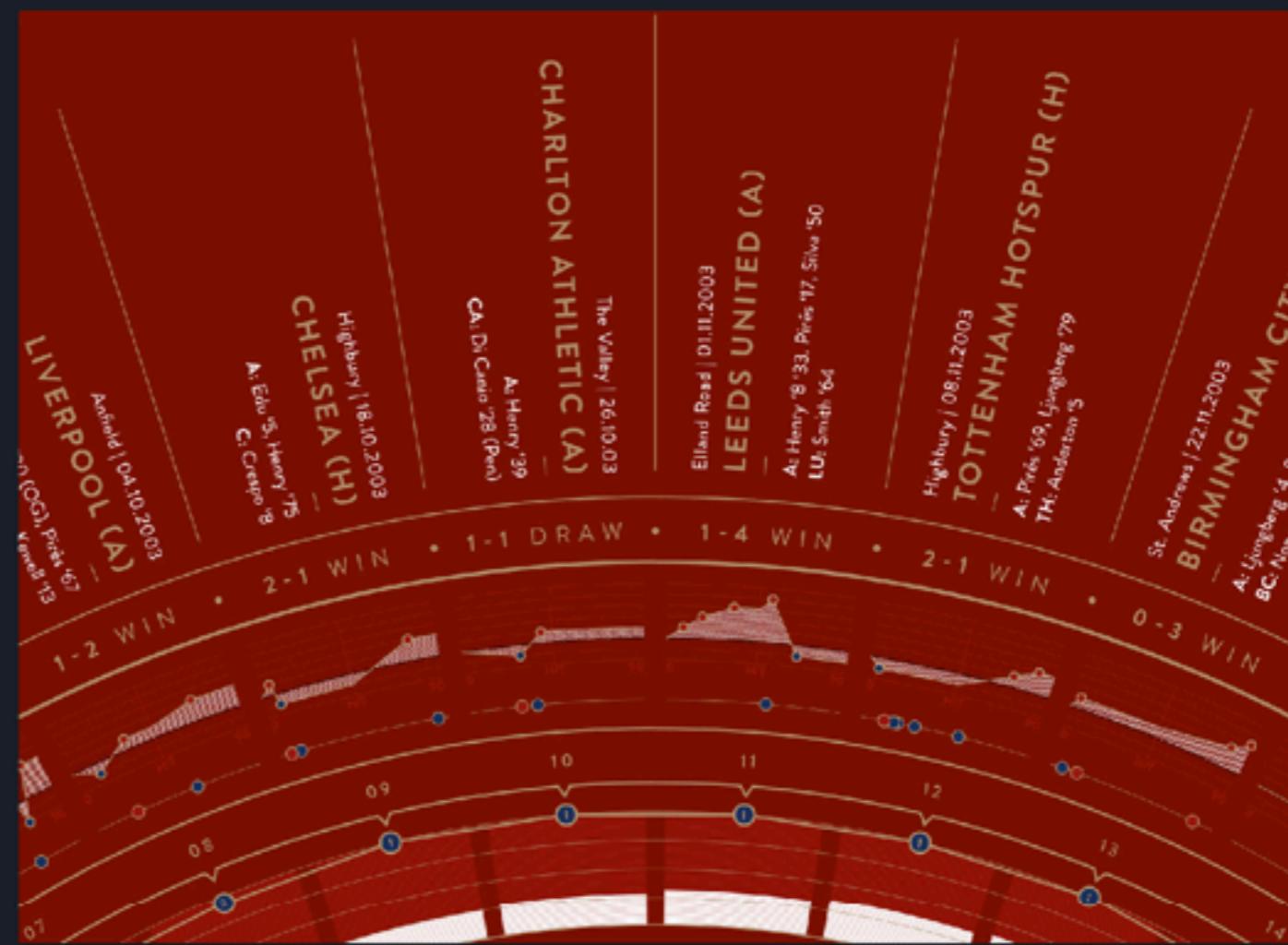
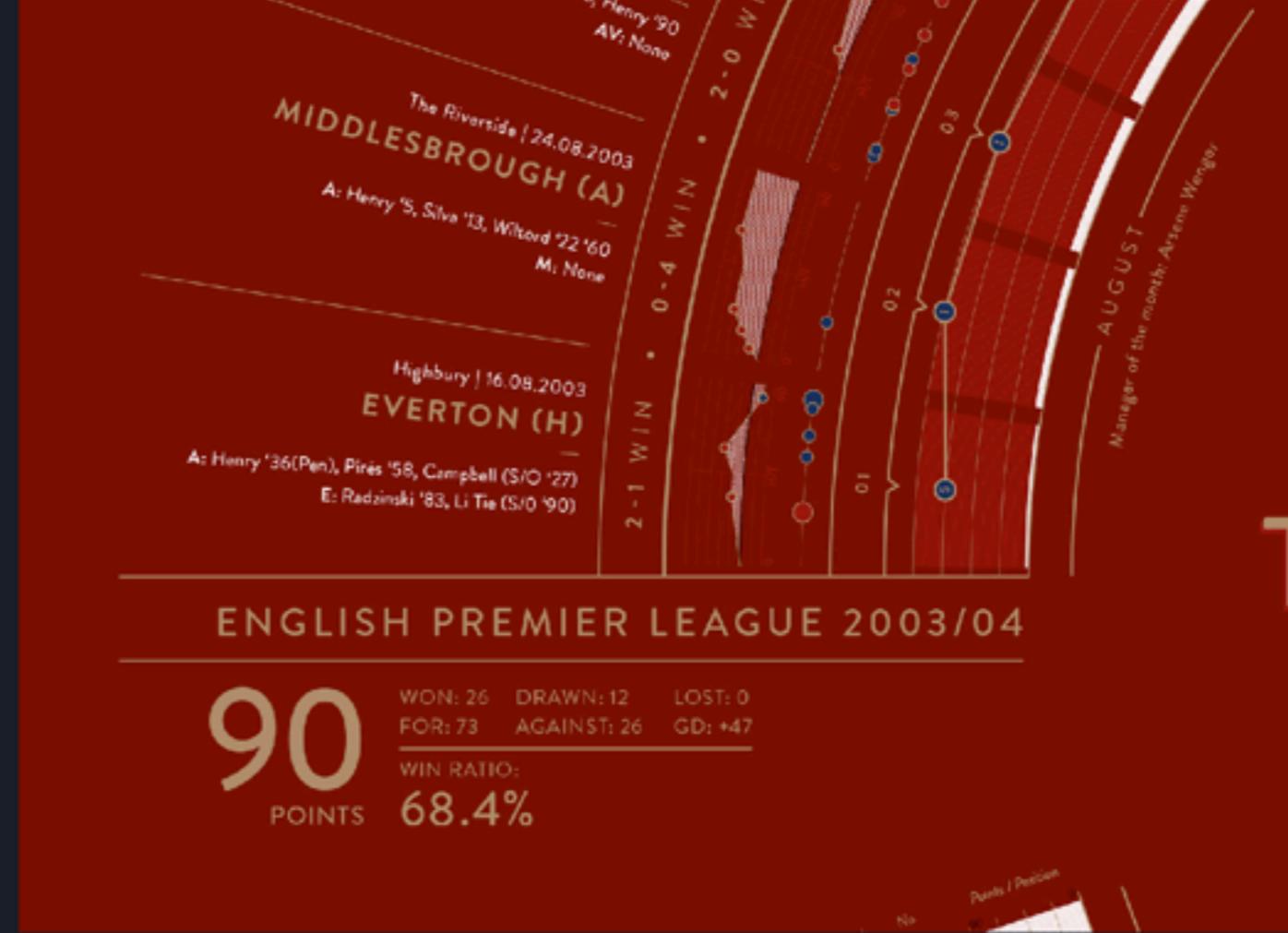
arcAlignArcs
arcEquiSpan
arcLambda
arcMaxRadius
arcRadius
arcRadiusInset
arcRotateBy
arcSpanUnit
maxDiameter
maxRadialOffset

Styles & Text

Actions

css
appendCSS
dataIndex
style
appendStyle
rawDataValue
styleLambda
textRotate
textVerticalAlign
textAnchor

Go Make The
AWEOME
THINGS



Thanks for Listening



@haroldcampbell

twitter | LinkedIn | github

SLIDES: <https://bit.ly/32cukyP>



@haroldcampbell

twitter | LinkedIn | github

SLIDES: <https://bit.ly/32cukyP>

groundTap.com



Do you wrangle with
applications, or are you
just to get in?

You Should

groundTap
visualise data