

# D3 in practice

Ali Almossawi  
SFHTML5, August 2015

Mozilla

*A team of statisticians*

Engineering

*MetricsGraphics.js*

Design

*webwewant.mozilla.org*

Research

*How do we measure code quality?*

# Vissarion Belinsky

Manages Mikhail Bakunin

Team size 1

## Staff in each timezone

1 (median), 1 (mean)

## Potential lost hours\*

0 hours (between staff)

0 hours (between manager and staff)

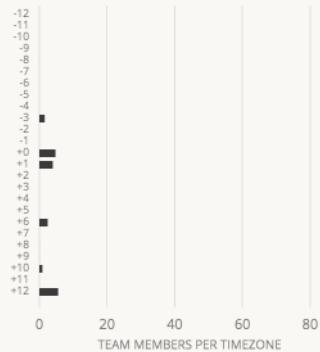
## Maximum time difference

0 hours (between staff)

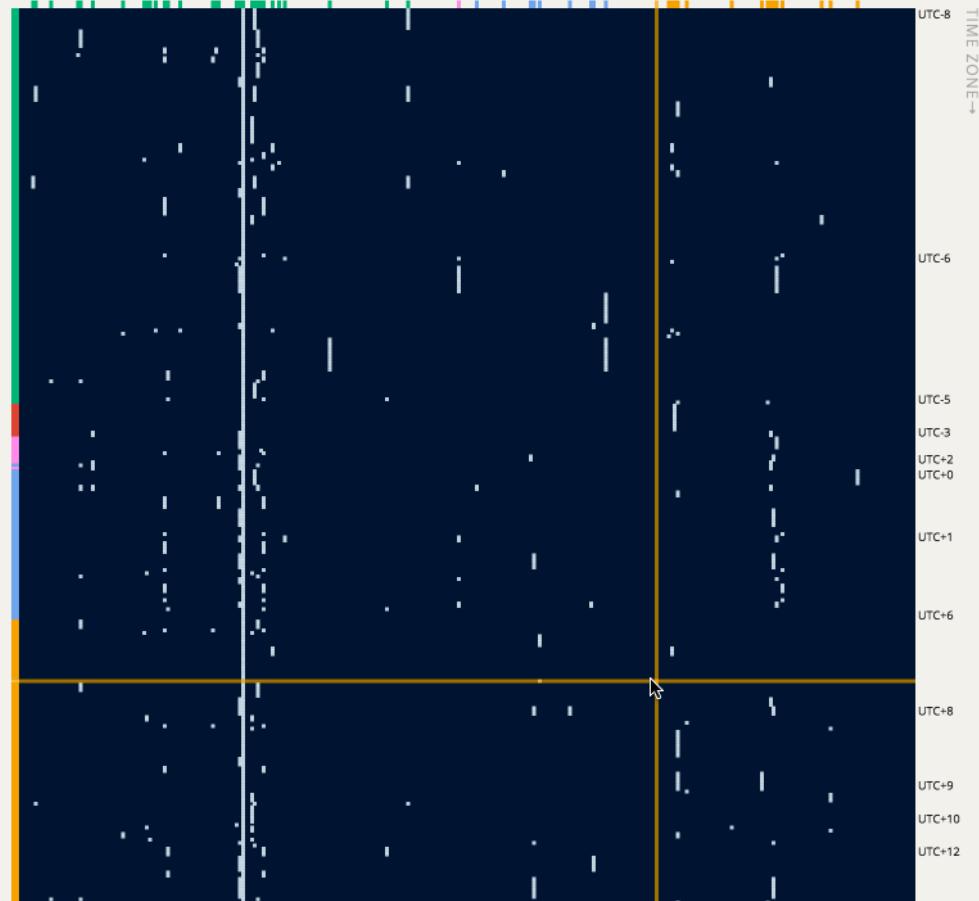
0 hours (between manager and staff)

## Largest group in the same time zone

1 staff, 100% of team are in UTC+6

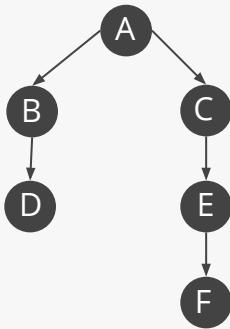


\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.



## Design Structure Matrix (DSM)

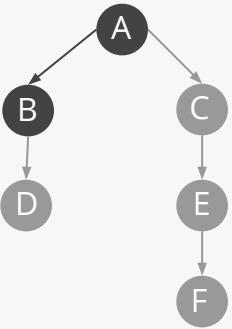
*Also known as an adjacency matrix or a dependency matrix*



$\alpha \longrightarrow \beta$      $\alpha$  depends on  $\beta$

	A	B	C	D	E	F
A	0	<b>1</b>	<b>1</b>	0	0	0
B	0	0	0	<b>1</b>	0	0
C	0	0	0	0	<b>1</b>	0
D	0	0	0	0	0	0
E	0	0	0	0	0	<b>1</b>
F	0	0	0	0	0	0

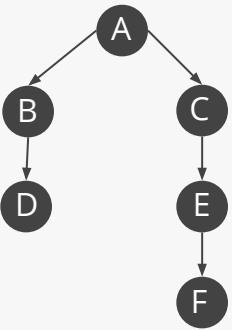
M



$\alpha \longrightarrow \beta$      $\alpha$  depends on  $\beta$

	A	<b>B</b>	C	D	E	F
<b>A</b>	0	1	1	0	0	0
B	0	0	0	1	0	0
C	0	0	0	0	1	0
D	0	0	0	0	0	0
E	0	0	0	0	0	1
F	0	0	0	0	0	0

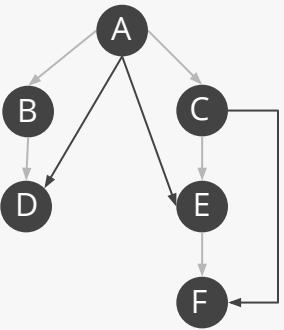
**M**



	A	B	C	D	E	F
A	0	<b>1</b>	<b>1</b>	0	0	0
B	0	0	0	<b>1</b>	0	0
C	0	0	0	0	<b>1</b>	0
D	0	0	0	0	0	0
E	0	0	0	0	0	<b>1</b>
F	0	0	0	0	0	0

**M<sup>1</sup>**

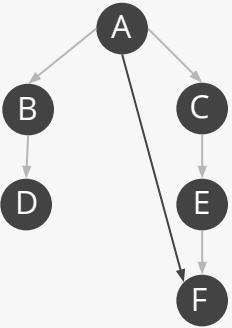
$\alpha \longrightarrow \beta$      $\alpha$  depends on  $\beta$



	A	B	C	D	E	F
A	0	0	0	<b>1</b>	<b>1</b>	0
B	0	0	0	0	0	0
C	0	0	0	0	0	<b>1</b>
D	0	0	0	0	0	0
E	0	0	0	0	0	0
F	0	0	0	0	0	0

$M^2$

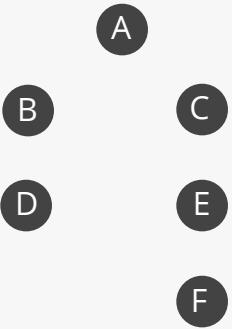
$\alpha \longrightarrow \beta \quad \alpha \text{ depends on } \beta$



	A	B	C	D	E	F
A	0	0	0	0	0	<b>1</b>
B	0	0	0	0	0	0
C	0	0	0	0	0	0
D	0	0	0	0	0	0
E	0	0	0	0	0	0
F	0	0	0	0	0	0

$M^3$

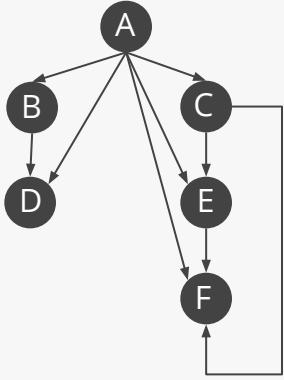
$\alpha \longrightarrow \beta \quad \alpha \text{ depends on } \beta$



	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	0	0	0	0	0	0
D	0	0	0	0	0	0
E	0	0	0	0	0	0
F	0	0	0	0	0	0

$M^4$

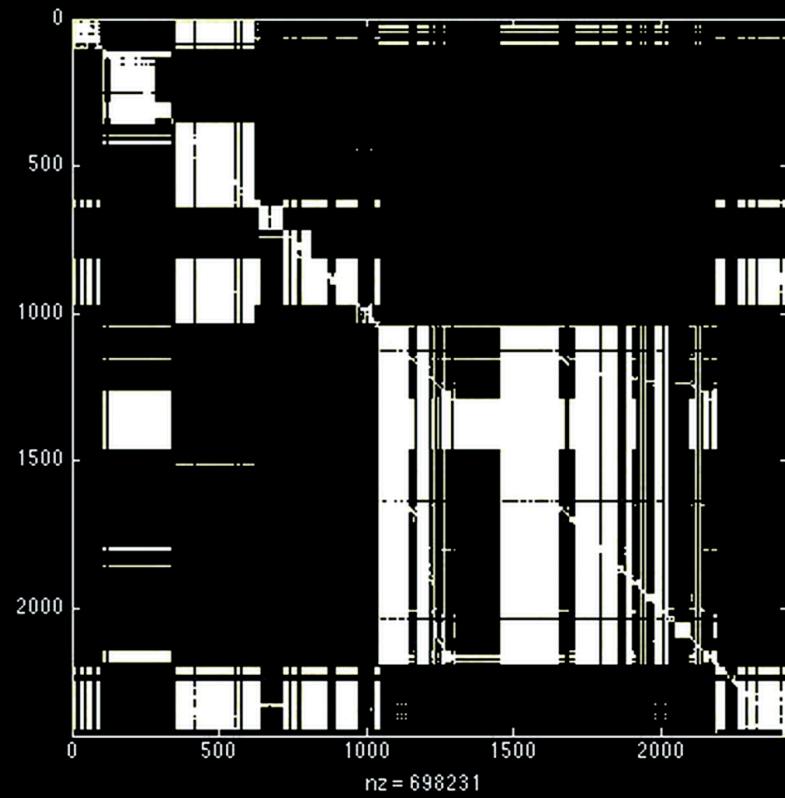
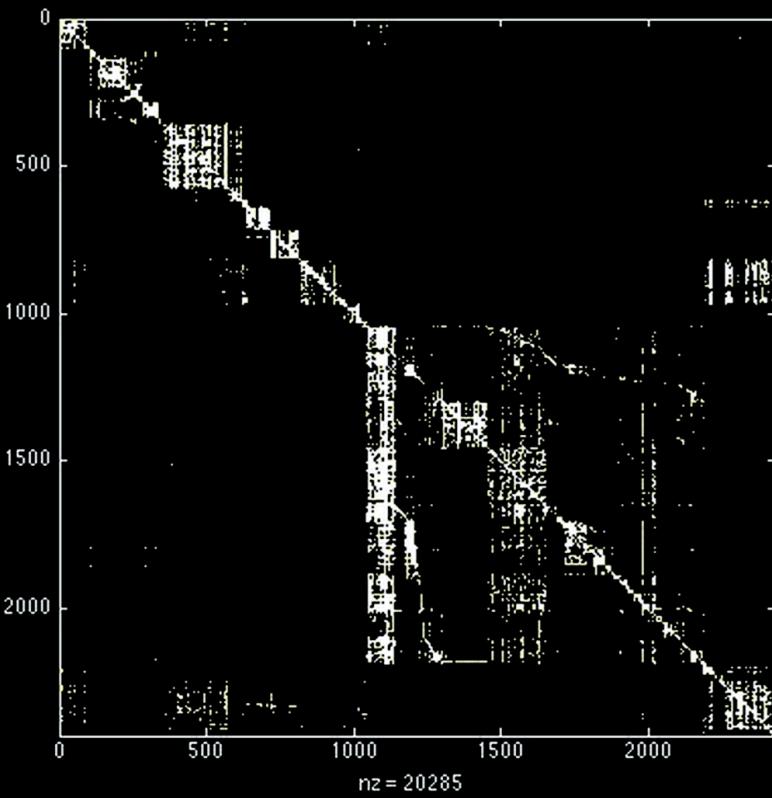
$\alpha \longrightarrow \beta$      $\alpha$  depends on  $\beta$



	A	B	C	D	E	F
A	0	1	1	1	1	1
B	0	0	0	1	0	0
C	0	0	0	0	1	1
D	0	0	0	0	0	0
E	0	0	0	0	0	1
F	0	0	0	0	0	0

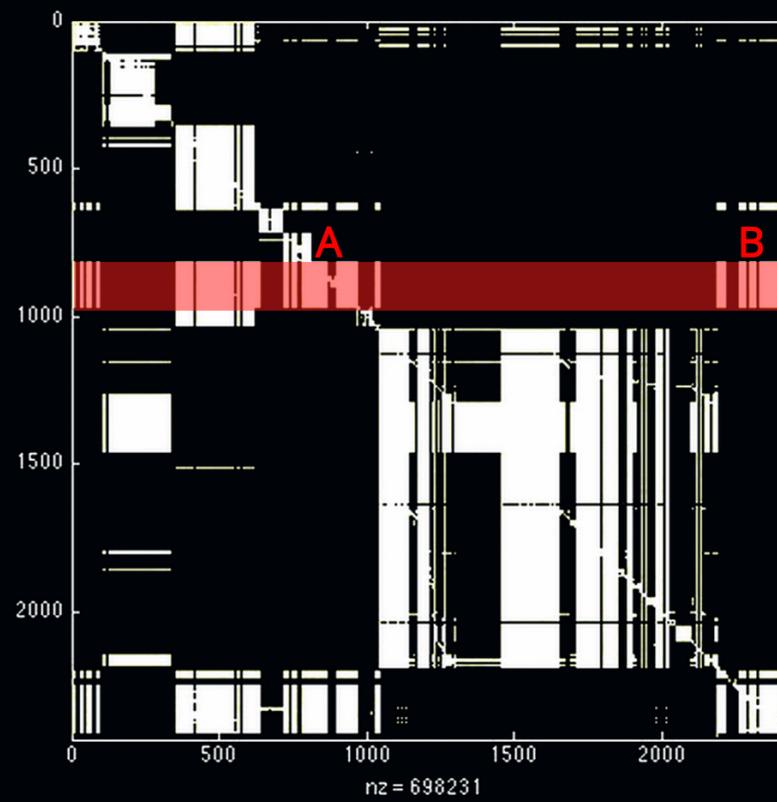
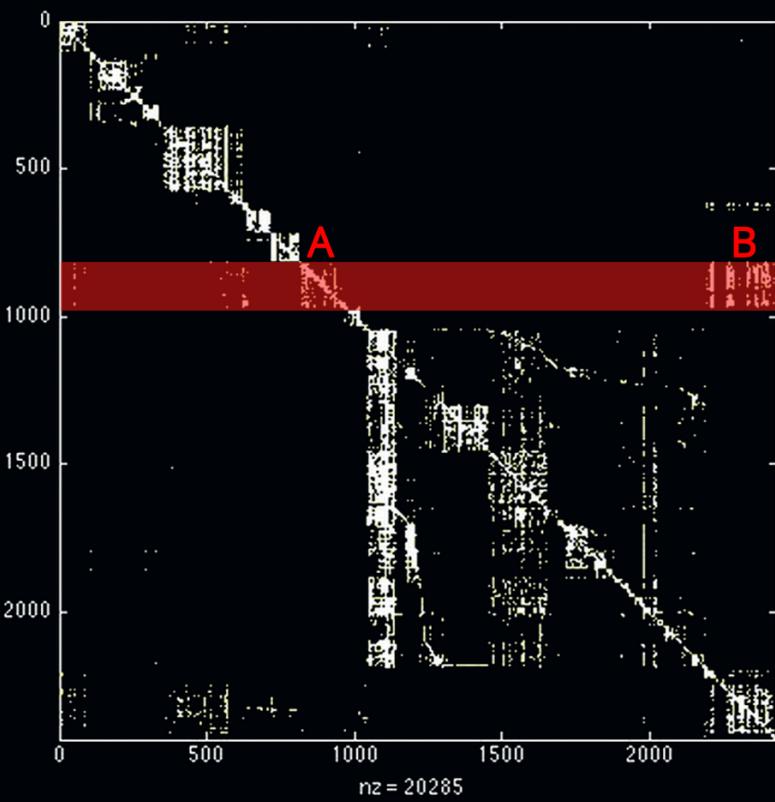
$$R = \sum M^n$$

$\alpha \longrightarrow \beta$      $\alpha$  depends on  $\beta$



The *gfx* module in Firefox 20, direct dependencies (left), direct+indirect dependencies (right)

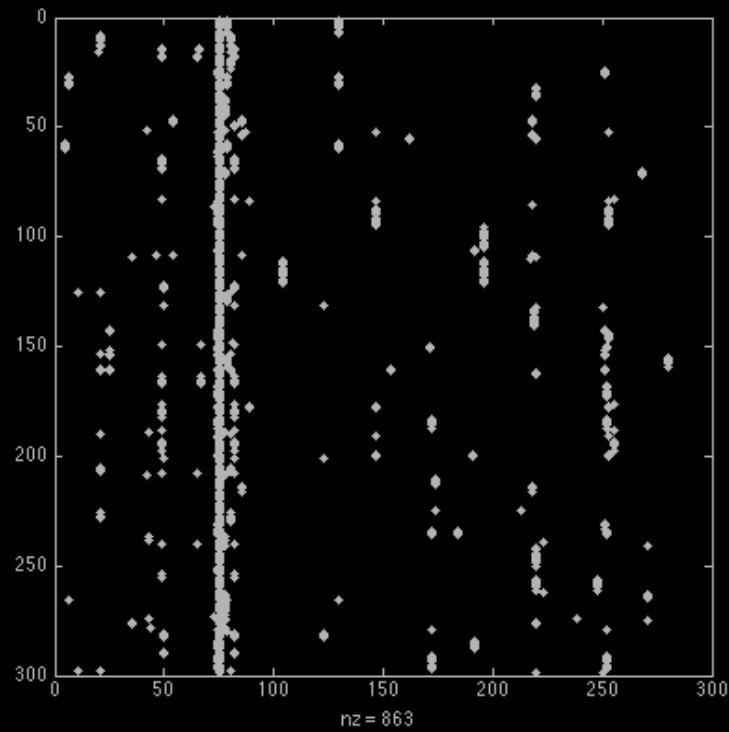
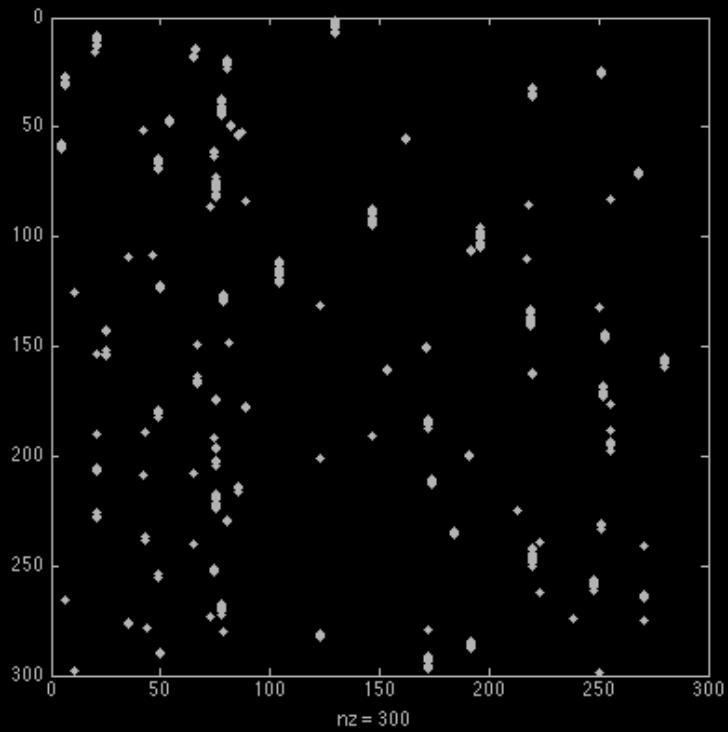
You can sort a DSM based on what's important to you



The *gfx* module in Firefox 20, direct dependencies (left), direct+indirect dependencies (right)

Say we want to see team distributions in an organization

*Sort on timezone then office then manager*



INITECH Staff, direct reports (left), direct+indirect reports (right)

Pick a manager

Ito Jinsai



## Ito Jinsai

Manages Samuel Bold

Team size 42

Staff in each timezone

4 (median), 4 (mean)

Potential lost hours\*

20 hours (between staff)

0 hours (between manager and staff)

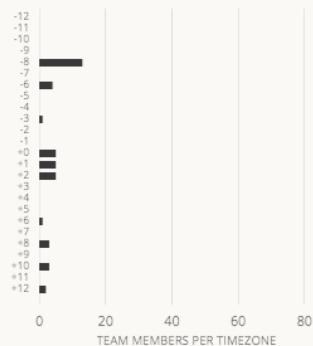
Maximum time difference

20 hours (between staff)

20 hours (between manager and staff)

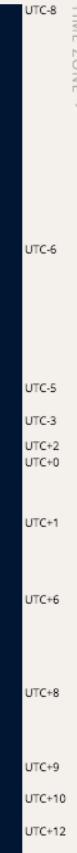
Largest group in the same time zone

13 staff, 31% of team are in UTC-8



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.

Ito Jinsai



# Augustus De Morgan

Manages William Hazlitt

Team size 5

## Staff in each timezone

5 (median), 5 (mean)

## Potential lost hours\*

0 hours (between staff)

10 hours (between manager and staff)

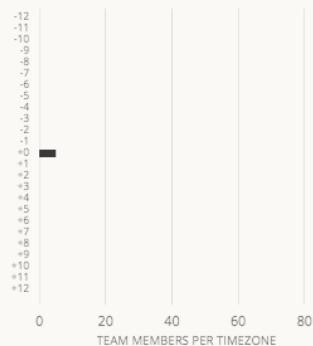
## Maximum time difference

0 hours (between staff)

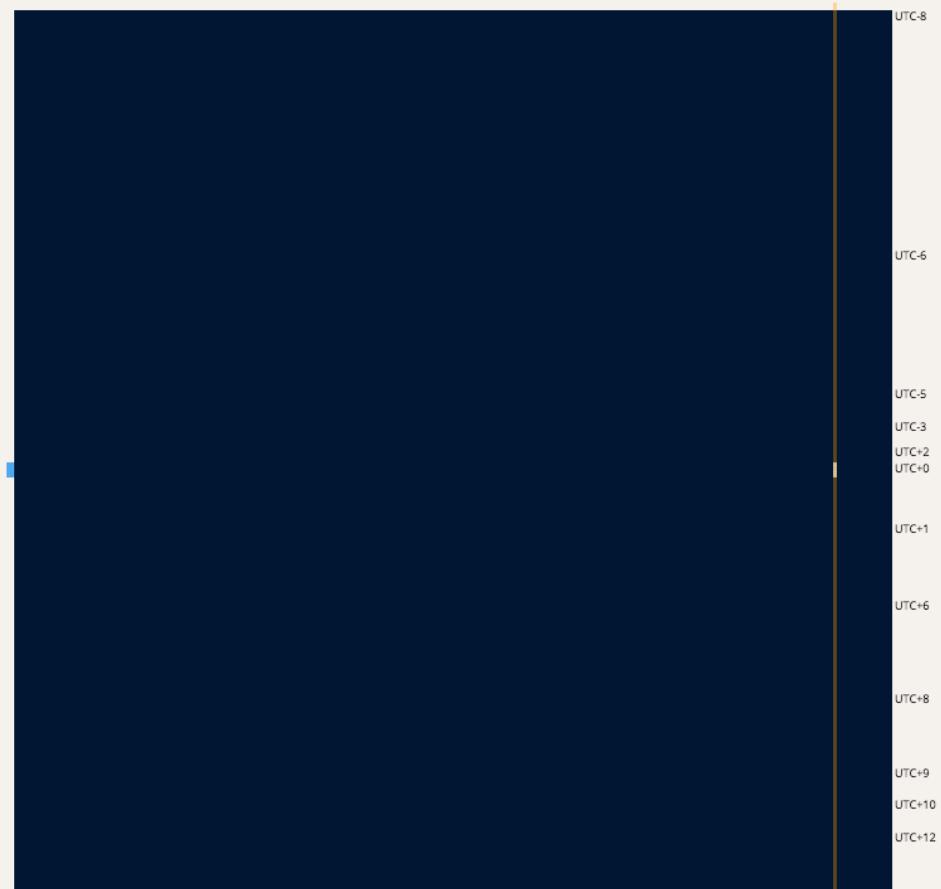
10 hours (between manager and staff)

## Largest group in the same time zone

5 staff, 100% of team are in UTC+0



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.



Pick a manager

Friedrich Schelling

Friedrich Schelling

# Friedrich Schelling

Manages Thomas Arnold

Team size 7

## Staff in each timezone

4 (median), 4 (mean)

## Potential lost hours\*

18 hours (between staff)

10 hours (between manager and staff)

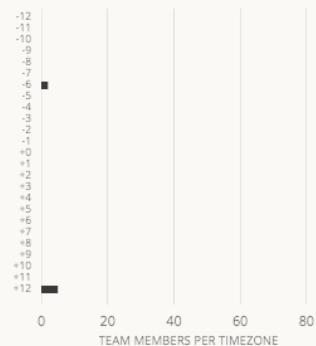
## Maximum time difference

18 hours (between staff)

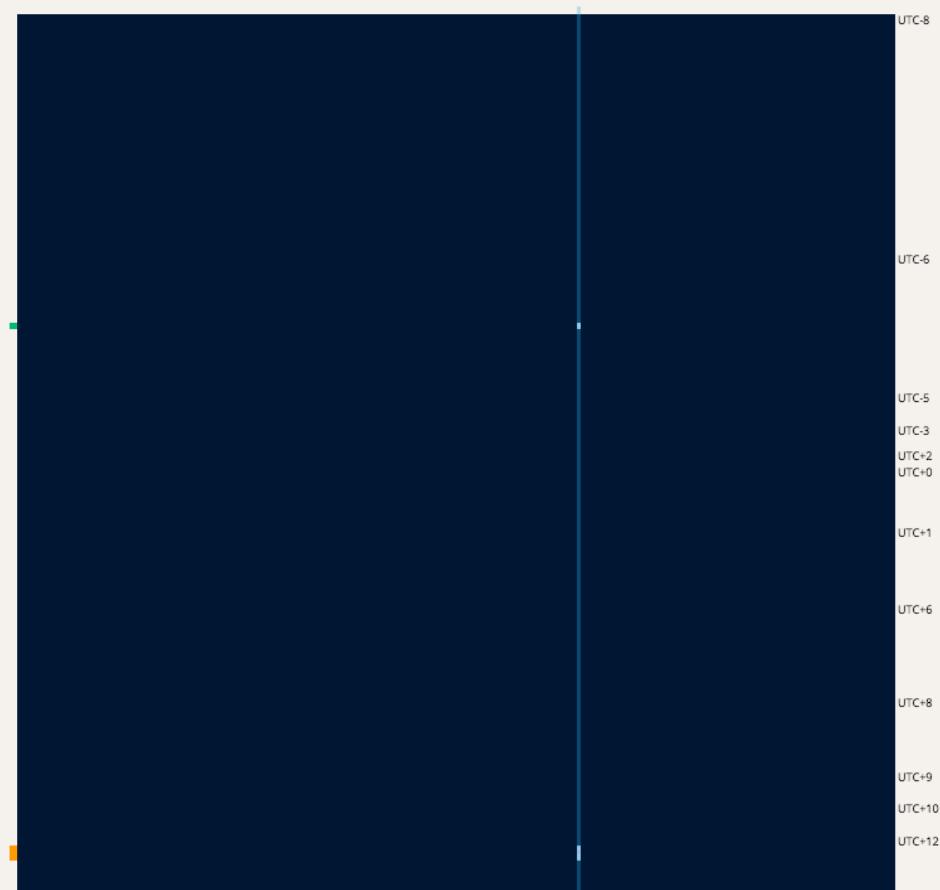
10 hours (between manager and staff)

## Largest group in the same time zone

5 staff, 71% of team are in UTC+12



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.



Pick a manager

Archibald Alison



Archibald Alison

## Archibald Alison

Manages Peter Browne

Team size 17

Staff in each timezone

9 (median), 9 (mean)

Potential lost hours\*

17 hours (between staff)

2 hours (between manager and staff)

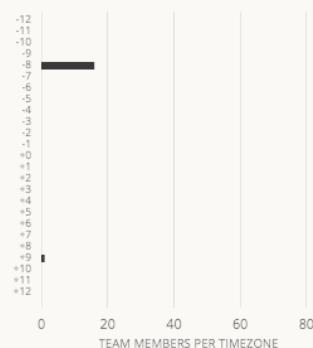
Maximum time difference

17 hours (between staff)

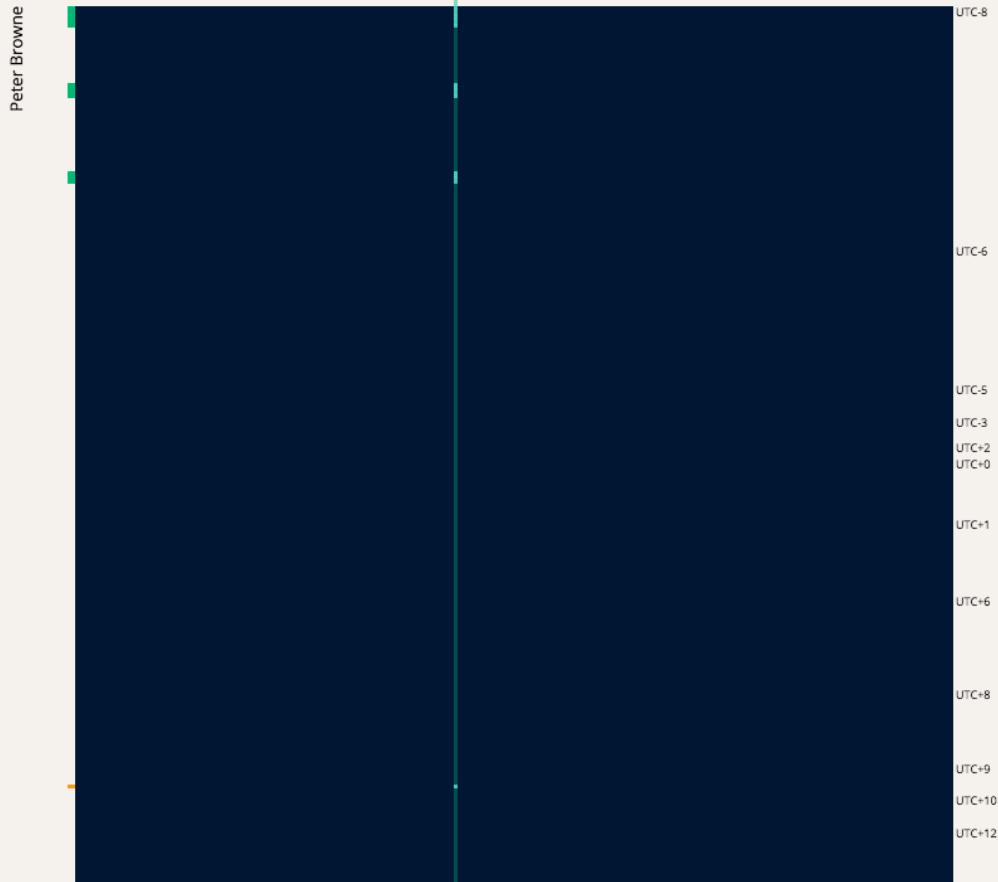
15 hours (between manager and staff)

Largest group in the same time zone

16 staff, 94% of team are in UTC-8



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.



## Ronnie Corbett

Manages Lord Kames

Team size 299

Staff in each timezone

17 (median), 25 (mean)

Potential lost hours\*

20 hours (between staff)

0 hours (between manager and staff)

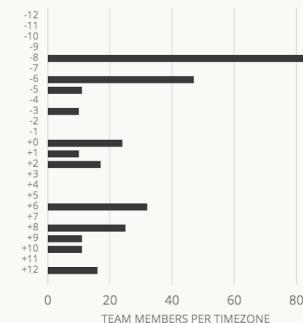
Maximum time difference

20 hours (between staff)

20 hours (between manager and staff)

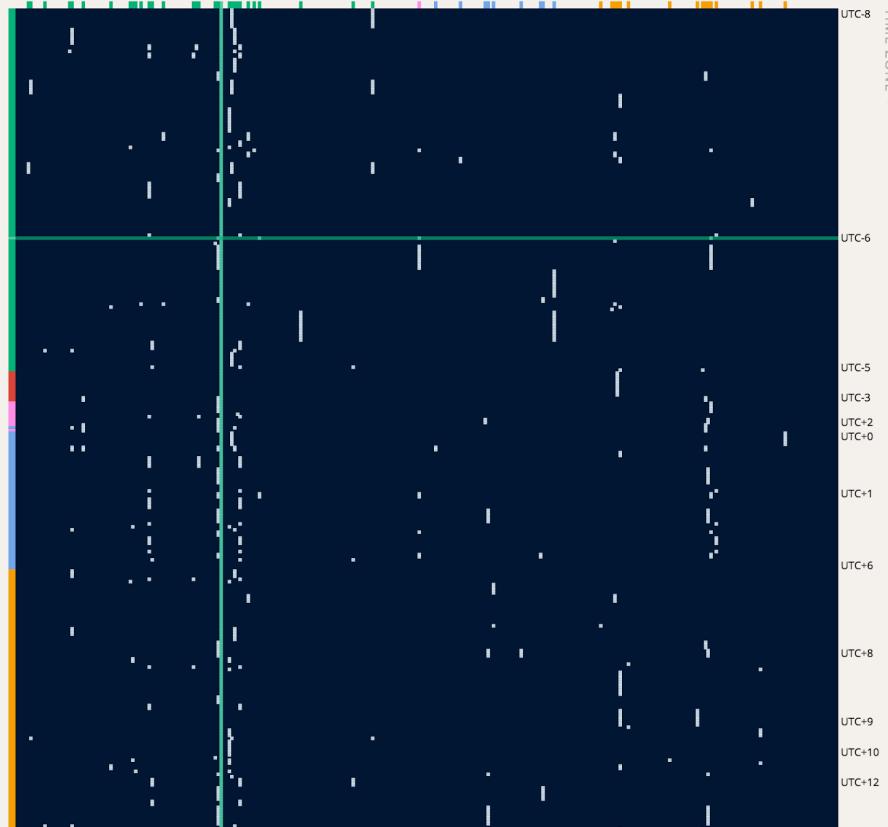
Largest group in the same time zone

85 staff, 28% of team are in UTC-8



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.

Pick a manager



Asia Pacific | Canada | Europe, Middle East and Africa | Latin America | USA

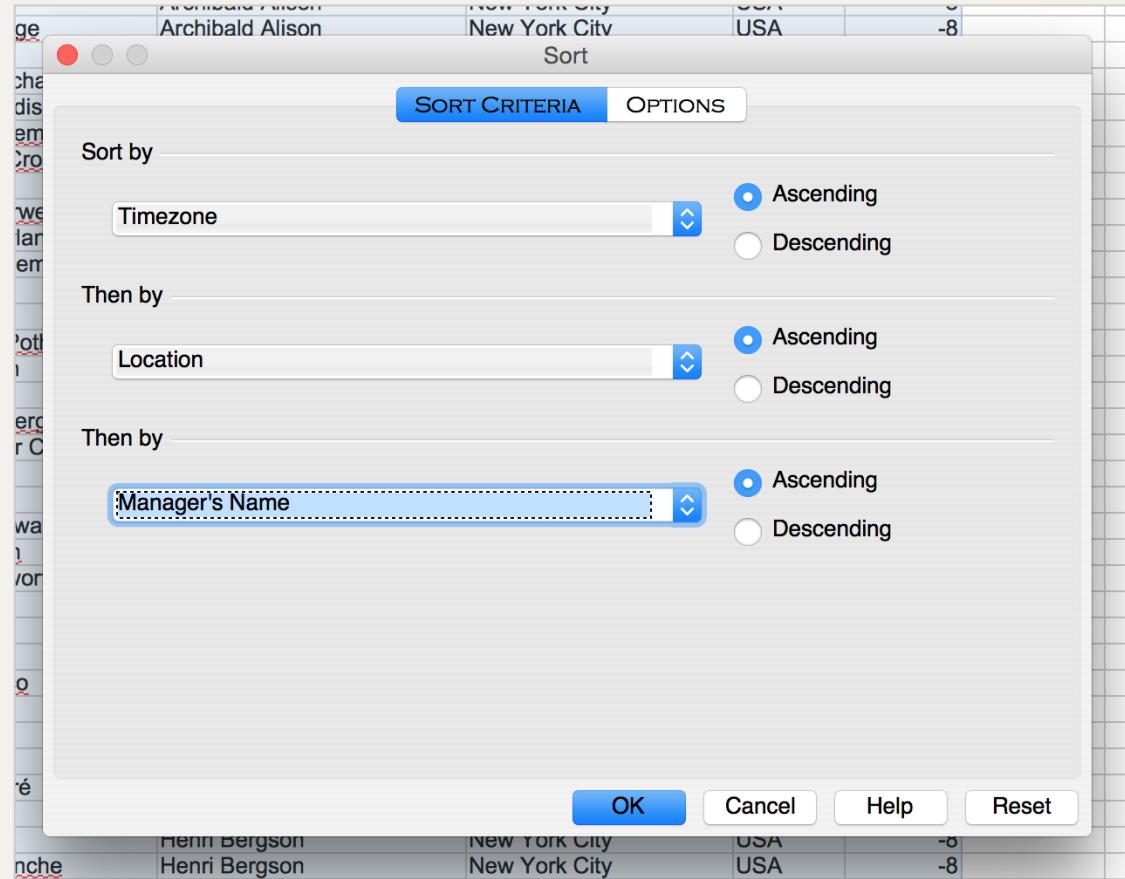
Elements: 863, Density: 0.34%  
 Total staff: 299, Distance to CEO (mean, median): 3  
 Team sizes: 15 (mean), 5 (median), 2 (mode)

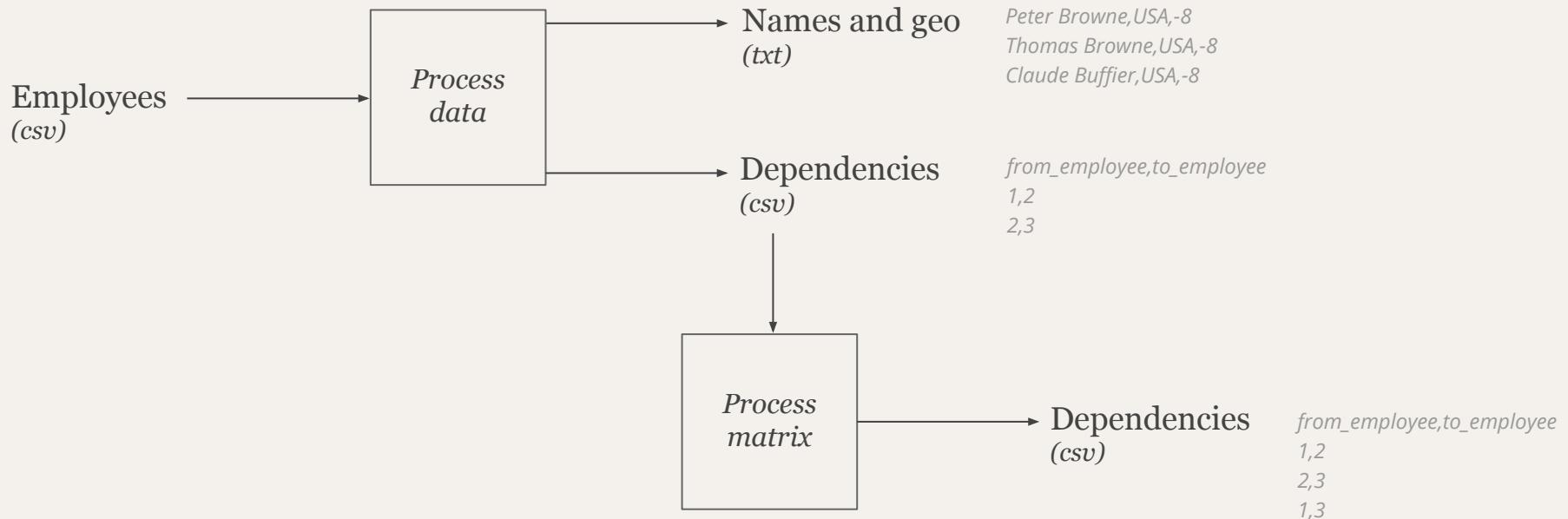
Code and demo  
*github.com/almossawi/d3-matrix*

1. Data
2. Scales
3. Data-join
4. Ticks
5. Voronoi
6. Events
7. Transitions
8. Stats and charts

1. Data
2. Scales
3. Data-join
4. Ticks
5. Voronoi
6. Events
7. Transitions
8. Stats and charts

	A	B	C	D	E	F
1	Name	Manager's Name	Location	Region	Timezone	
2	Peter Browne	Archibald Alison	Portland	USA	-8	
3	Thomas Browne	Archibald Alison	Portland	USA	-8	
4	Claude Buffier	Archibald Alison	Portland	USA	-8	
5	Richard Burthogge	Archibald Alison	Portland	USA	-8	
6	Joseph Butler	Archibald Alison	Portland	USA	-8	
7	Gershom Carmichael	Archibald Alison	Portland	USA	-8	
8	Margaret Cavendish	Archibald Alison	Portland	USA	-8	
9	Geraud de Cordemoy	Arthur Collier	Portland	USA	-8	
10	Jean-Pierre de Crousaz	Arthur Collier	Portland	USA	-8	
11	Ralph Cudworth	Arthur Collier	Portland	USA	-8	
12	Nathaniel Culverwel	Arthur Collier	Portland	USA	-8	
13	Richard Cumberland	Arthur Collier	Portland	USA	-8	
14	Elisabeth of Bohemia	Arthur Collier	Portland	USA	-8	
15	Samuel Bold	Baron de Montesquieu	Portland	USA	-8	
16	Robert Boyle	Baron de Montesquieu	Portland	USA	-8	
17	Robert Joseph Pothier	Catherine Trotter Cockburn	Portland	USA	-8	
18	Samuel Johnson	Chris Hofmann	Portland	USA	-8	
19	Kaibara Ekiken	Chris Hofmann	Portland	USA	-8	
20	Johannes Clauberg	Francis Hutcheson	Portland	USA	-8	
21	Catherine Trotter Cockburn	Francis Hutcheson	Portland	USA	-8	
22	Arthur Collier	Francis Hutcheson	Portland	USA	-8	
23	Anthony Collins	Francis Hutcheson	Portland	USA	-8	
24	Anne Finch Conway	Francis Hutcheson	Portland	USA	-8	
25	Walter Charleton	Franz Brentano	Portland	USA	-8	
26	William Chillingworth	Franz Brentano	Portland	USA	-8	
27	Samuel Clarke	Franz Brentano	Portland	USA	-8	
28	John Toland	Gershom Carmichael	Portland	USA	-8	
29	George Turnbull	Gershom Carmichael	Portland	USA	-8	
30	Giambattista Vico	Gershom Carmichael	Portland	USA	-8	
31	Voltaire	Gershom Carmichael	Portland	USA	-8	
32	Wang Fuzhi	Gershom Carmichael	Portland	USA	-8	
33	Firmin Abauzit	Henri Bergson	Portland	USA	-8	
34	Yves Marie André	Henri Bergson	Portland	USA	-8	
35	Antoine Arnauld	Henri Bergson	Portland	USA	-8	
36	John Locke	Henri Bergson	Portland	USA	-8	
37	Nicolas Malebranche	Henri Bergson	Portland	USA	-8	
38	John Norris	Huang Zongxi	Portland	USA	-8	
39	Ogyü Sorai	Huang Zongxi	Portland	USA	-8	
40	Samuel Pufendorf	Huang Zongxi	Portland	USA	-8	
41	John Ray	Huang Zongxi	Portland	USA	-8	
42	Pierre-Sylvain Regis	Huang Zongxi	Portland	USA	-8	
43	Jacques Rohault	Huang Zongxi	Portland	USA	-8	
44	Anna Maria van Schurman	Huang Zongxi	Portland	USA	-8	
45	John Sergeant	Huang Zongxi	Portland	USA	-8	
46	Baruch Spinoza	Huang Zongxi	Portland	USA	-8	
47	James Dalrymple	Humphry Ditton	Portland	USA	-8	
48	Edward Stillingfleet	Humphry Ditton	Portland	USA	-8	
49	Gabrielle Suchon	Humphry Ditton	Portland	USA	-8	
50	Kumazawa Banzan	Ito Jinsai	Portland	USA	-8	
51	Louis de La Forge	Ito Jinsai	Portland	USA	-8	





$\alpha \rightarrow \beta \rightarrow \gamma$        $\beta$  processes  $\alpha$ , outputs  $\gamma$

*employees.txt*

```
Peter Browne,USA,-8  
Thomas Browne,USA,-8  
Claude Buffier,USA,-8
```

*employees-dependencies.csv*

```
from_employee,to_employee  
1,2  
2,3
```

```
1 d3.csv('data/employees-dependencies.csv', function(data) {  
2   d3.text('data/employees.txt', function(text) {  
3     // do stuff  
4   });  
5 });
```

```
var args = {
    padding: 52,
    matrix_height: 800,
    employee_data: {}
};

};
```

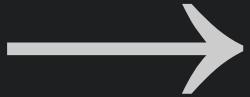
```
1 d3.csv.parseRows(text).forEach(function(row, i) {
2     args.employee_data[++i] = {
3         name: row[0],
4         region: row[1],
5         timezone: row[2]
6     };
7 });

});
```

```
Peter Browne,USA,-8    --->  ["Peter Browne", "USA", "-8"]
Thomas Browne,USA,-8  --->  ["Thomas Browne", "USA", "-8"]
```

```
1 var nested = d3.nest()  
2     .key(function(d) {  
3         return d.name;  
4     })  
5     .entries(data);
```

```
[  
  {  
    "name": "jill",  
    "date": "2015-01-01"  
  },  
  {  
    "name": "jill",  
    "date": "2015-01-02"  
  },  
  {  
    "name": "jack",  
    "date": "2015-01-01"  
  },  
  {  
    "name": "jack",  
    "date": "2015-01-02"  
  }  
]
```



```
[  
  {  
    "key": "jill",  
    "values": [  
      {  
        "name": "jill",  
        "date": "2015-01-01"  
      },  
      {  
        "name": "jill",  
        "date": "2015-01-02"  
      }  
    ]  
  },  
  {  
    "key": "jack",  
    "values": [  
      {  
        "name": "jack",  
        "date": "2015-01-01"  
      },  
      {  
        "name": "jack",  
        "date": "2015-01-02"  
      }  
    ]  
  }  
]
```

```
▼ Object ⓘ
▶ dependencies: Array[863]
  employee_count: 299
▶ employee_data: Object
  in_manager_view: false
▶ managers: Array[57]
▶ managers_team: Object
  matrix_height: 800
  padding: 52
▶ voronoi: function t(n)
▶ x: function i(n)
▶ y: function i(n)
▶ __proto__: Object
```

```
<div class='dsm'></div>
```

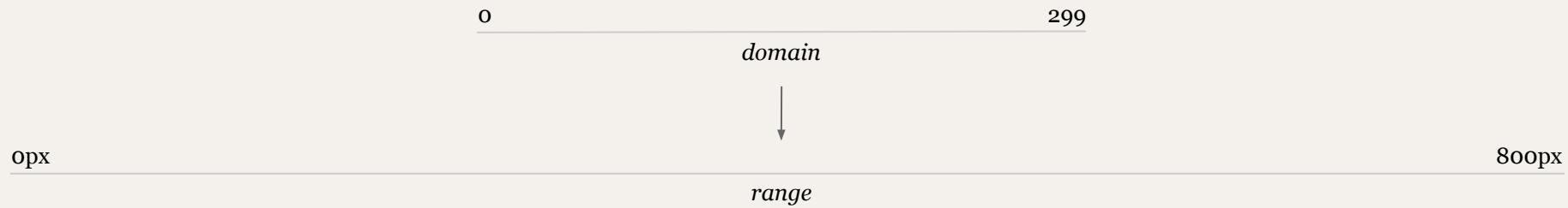
```
1 var svg = d3.select('.dsm')
2     .append('svg')
3         .attr('width', args.matrix_height)
4             .attr('height', args.matrix_height);
5
6 svg.append("rect")
7     .attr("width", args.matrix_height - (args.padding * 2))
8     .attr("height", args.matrix_height - (args.padding * 2))
9     .attr('x', args.padding)
10    .attr('y', args.padding);
```

```
<div class='dsm'></div>
```

```
1 var svg = d3.select('.dsm')
2     .append('svg')
3         .attr('width', args.matrix_height)
4             .attr('height', args.matrix_height);
5
6 svg.append("rect")
7     .attr("width", args.matrix_height - (args.padding * 2))
8     .attr("height", args.matrix_height - (args.padding * 2))
9     .attr('x', args.padding)
10    .attr('y', args.padding);
```



1. Data
- 2. Scales**
3. Data-join
4. Ticks
5. Voronoi
6. Events
7. Transitions
8. Stats and charts



```
1 args.x = args.y = d3.scale.linear()  
2     .domain([1, 299])  
3     .range([0, 800]);  
4  
5 args.x(1); // 0  
6 args.x(299); // 800  
7 args.x(150); // 400
```

```
1 var dots = svg.append('g')
2       .attr('class', 'dot');
```

1. Data
2. Scales
- 3. Data-join**
4. Ticks
5. Voronoi
6. Events
7. Transitions
8. Stats and charts

```
args.dependencies = [
    {from_employee: 1, to_employee: 2},
    {from_employee: 3, to_employee: 2}
]
```

```
1 dots.selectAll('rect')
2     .data(args.dependencies)
3     .enter().append('rect')
4     .attr('class', function(d) {
5         return 'm' + d.to_employee + ' e' + d.from_employee;
6     })
7     .attr('width', 1)
8     .attr('height', 1)
9     .attr('transform', function(d) {
10        return "translate(" + args.x(d.to_employee).toFixed(2)
11                + "," + args.y(d.from_employee).toFixed(2) + ")";
12   }); // toFixed is to avoid DOM diarrhea (ቃ__ቃ)
```

```
args.dependencies = [
    {from_employee: 1, to_employee: 2},
    {from_employee: 3, to_employee: 2}
]
```

```
| 1 dots.selectAll('rect')
| 2     .data(args.dependencies)
| 3     .enter().append('rect')
| 4     .attr('class', function(d) {
| 5         return 'm' + d.to_employee + ' e' + d.from_employee;
| 6     })
| 7     .attr('width', 1)
| 8     .attr('height', 1)
| 9     .attr('transform', function(d) {
| 10        return "translate(" + args.x(d.to_employee).toFixed(2)
| 11            + "," + args.y(d.from_employee).toFixed(2) + ")";
| 12    }); // toFixed is to avoid DOM diarrhea (ቃ__ቃ)
```

```
args.dependencies = [
    {from_employee: 1, to_employee: 2},
    {from_employee: 3, to_employee: 2}
]
```

```
1 dots.selectAll('rect')
| 2     .data(args.dependencies)
3     .enter().append('rect')
4     .attr('class', function(d) {
5         return 'm' + d.to_employee + ' e' + d.from_employee;
6     })
7     .attr('width', 1)
8     .attr('height', 1)
9     .attr('transform', function(d) {
10        return "translate(" + args.x(d.to_employee).toFixed(2)
11            + "," + args.y(d.from_employee).toFixed(2) + ")";
12   }); // toFixed is to avoid DOM diarrhea (ቃ__ቃ)
```

```
args.dependencies = [
    {from_employee: 1, to_employee: 2},
    {from_employee: 3, to_employee: 2}
]
```

```
1 dots.selectAll('rect')
2     .data(args.dependencies)
| 3 .enter().append('rect')
4     .attr('class', function(d) {
5         return 'm' + d.to_employee + ' e' + d.from_employee;
6     })
7     .attr('width', 1)
8     .attr('height', 1)
9     .attr('transform', function(d) {
10        return "translate(" + args.x(d.to_employee).toFixed(2)
11            + "," + args.y(d.from_employee).toFixed(2) + ")";
12   }); // toFixed is to avoid DOM diarrhea (ቃ__ቃ)
```

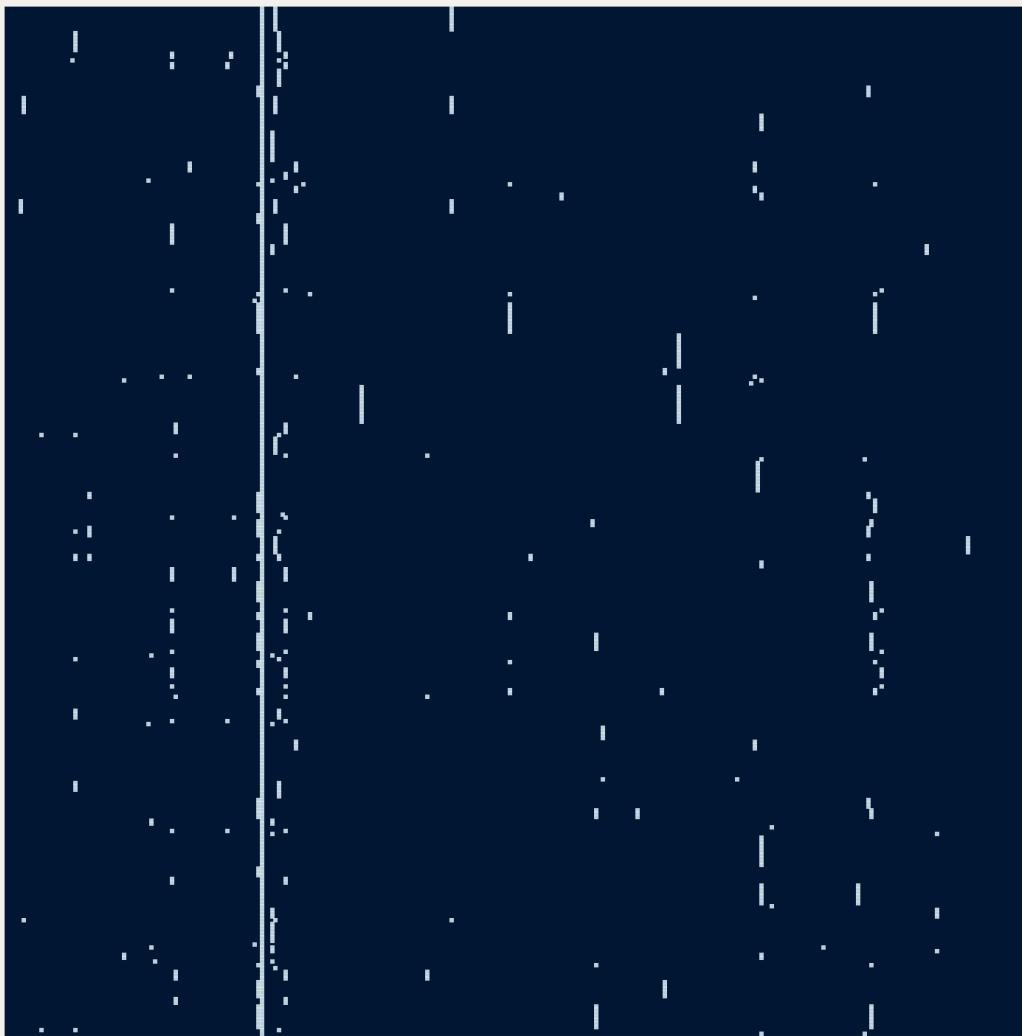
```
args.dependencies = [
    {from_employee: 1, to_employee: 2},
    {from_employee: 3, to_employee: 2}
]
```

```
1 dots.selectAll('rect')
2     .data(args.dependencies)
3     .enter().append('rect')
4     .attr('class', function(d) {
5         return 'm' + d.to_employee + ' e' + d.from_employee;
6     })
7     .attr('width', 1)
8     .attr('height', 1)
9     .attr('transform', function(d) {
10        return "translate(" + args.x(d.to_employee).toFixed(2)
11                + "," + args.y(d.from_employee).toFixed(2) + ")";
12   }); // toFixed is to avoid DOM diarrhea (ቃ__ቃ)
```

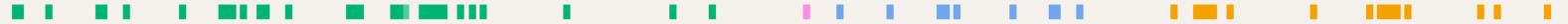
```
args.dependencies = [
    {from_employee: 1, to_employee: 2},
    {from_employee: 3, to_employee: 2}
]
```

```
1 dots.selectAll('rect')
2     .data(args.dependencies)
3     .enter().append('rect')
4     .attr('class', function(d) {
5         return 'm' + d.to_employee + ' e' + d.from_employee;
6     })
7     .attr('width', 1)
8     .attr('height', 1)
9     .attr('transform', function(d) {
10        return "translate(" + args.x(d.to_employee).toFixed(2)
11                + "," + args.y(d.from_employee).toFixed(2) + ")";
12   }); // toFixed is to avoid DOM diarrhea (ಠ_ಠ)
```

```
.dot rect {  
    fill: #d2e4ed;  
    opacity: 0.9;  
}
```



1. Data
2. Scales
3. Data-join
4. Ticks
5. Voronoi
6. Events
7. Transitions
8. Stats and charts



Asia Pacific

Canada

Europe, Middle East and Africa

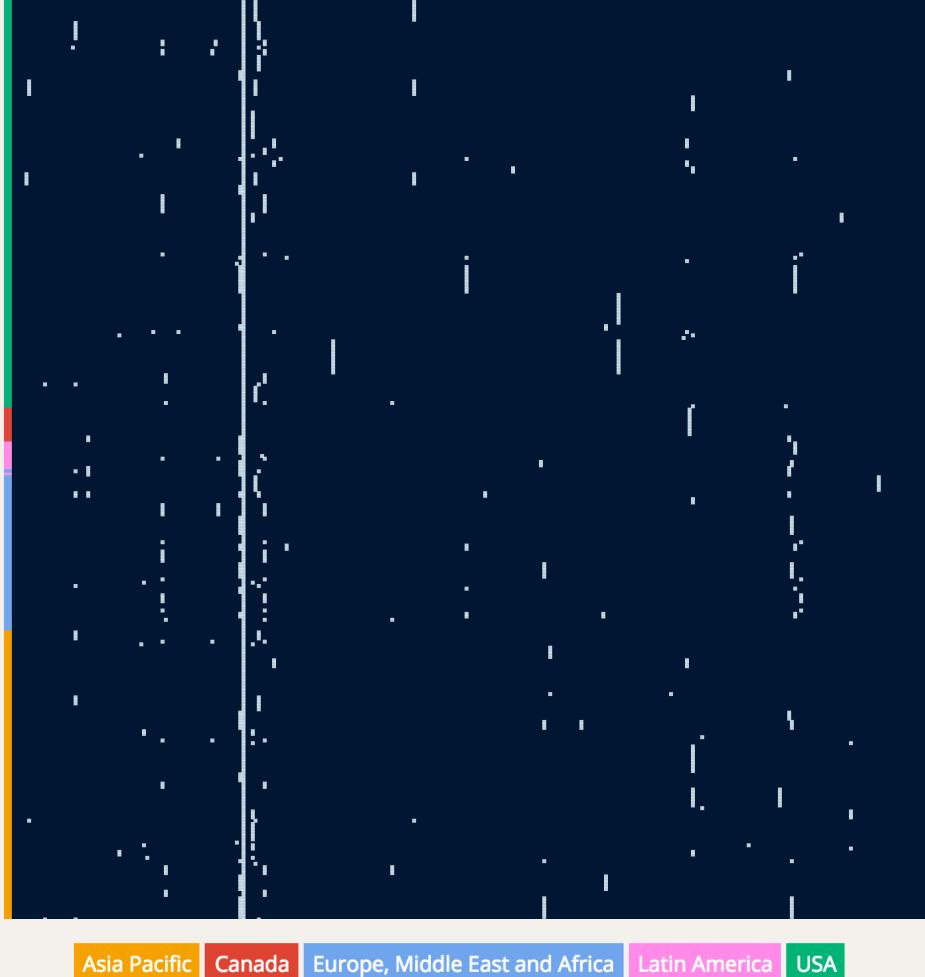
Latin America

USA

```
.EMEA {  
    color: #71a6ea;  
    fill: #71a6ea;  
}  
  
.USA {  
    color: #05b378;  
    fill: #05b378;  
}  
  
.CANADA {  
    color: #db4437;  
    fill: #db4437;  
}
```

```
1 var timezone_x = svg.append('g')
2     .attr('class', 'timezone-x');
3
4 timezone_x.selectAll("rect")
| 5     .data(dedup(args.dependencies, 'to_employee'))
6     .enter().append("rect")
7     .attr('class', function(d) {
8         return args.employee_data[d.to_employee].region
9             + ' m' + d.to_employee;
10    })
11    .attr('width', 2)
12    .attr('height', 6)
13    .attr("transform", function(d) {
14        return "translate(" + args.x(d.to_employee).toFixed(2)
15            + "," + (args.padding - 6).toFixed(2) + ")";
16    });

```



TIME ZONE →

UTC-8

UTC-6

UTC-5

UTC-3

UTC+2

UTC+0

UTC+1

UTC+6

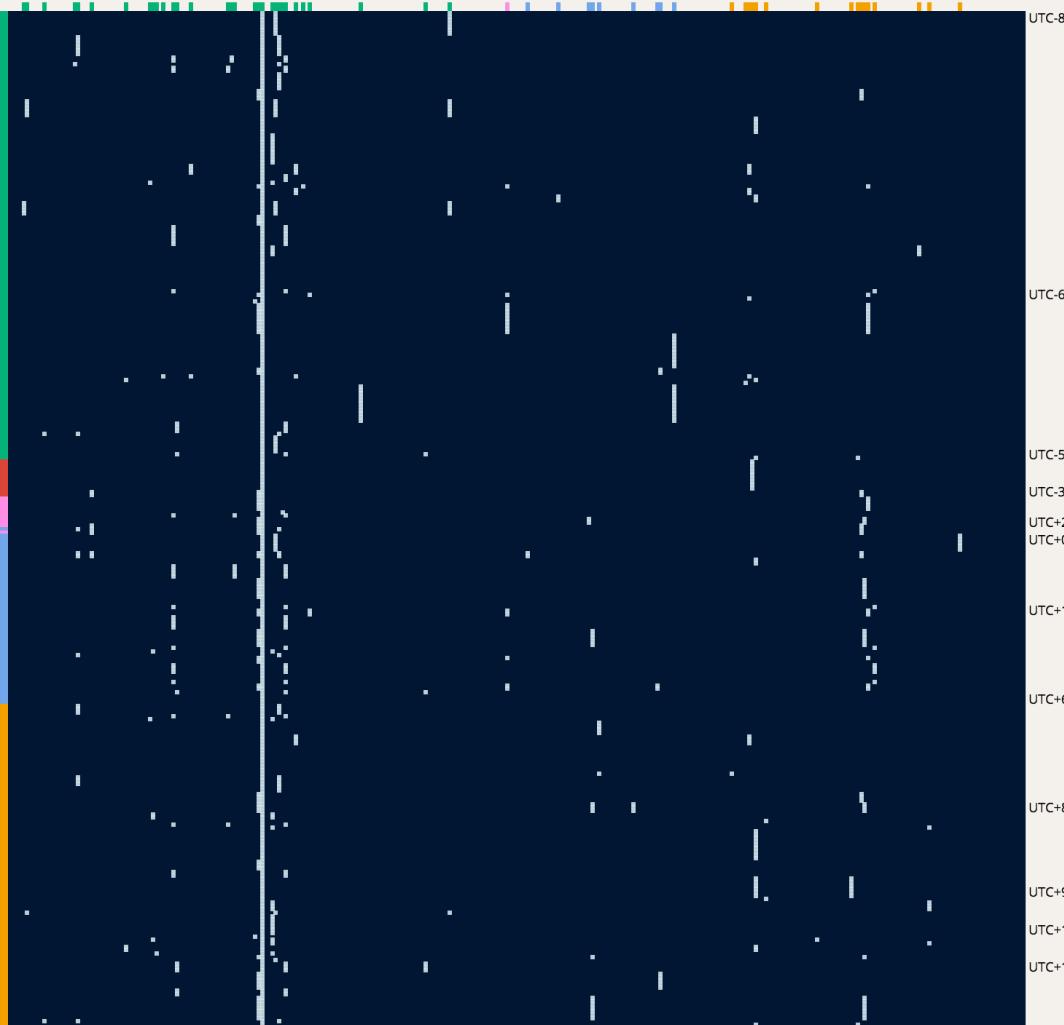
UTC+8

UTC+9

UTC+10

UTC+12





1. Data
2. Scales
3. Data-join
4. Ticks
5. Voronoi
6. Events
7. Transitions
8. Stats and charts

# Percy Williams Bridgman

Manages Mary Wollstonecraft

Team size 24

## Staff in each timezone

4 (median), 4 (mean)

## Potential lost hours\*

15 hours (between staff)

4 hours (between manager and staff)

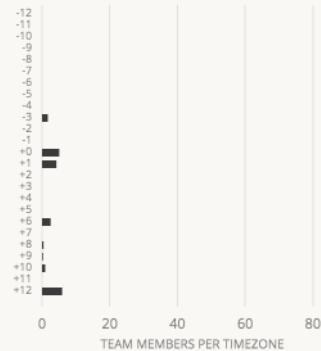
## Maximum time difference

15 hours (between staff)

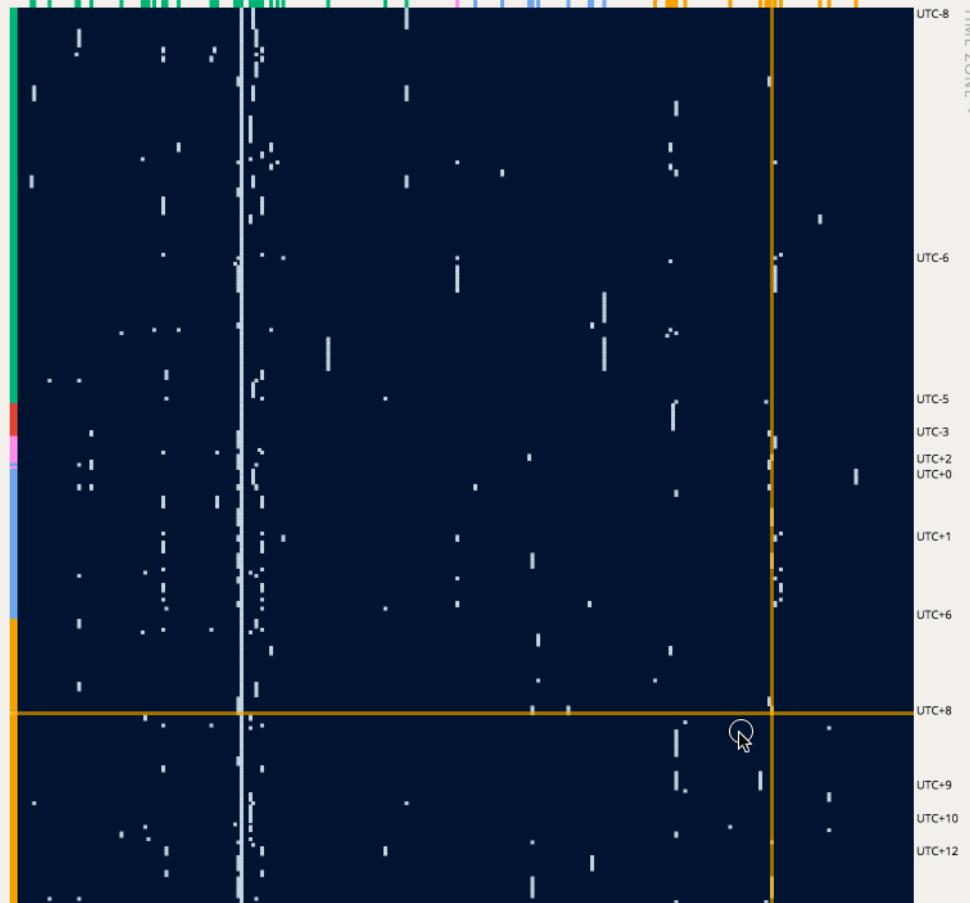
11 hours (between manager and staff)

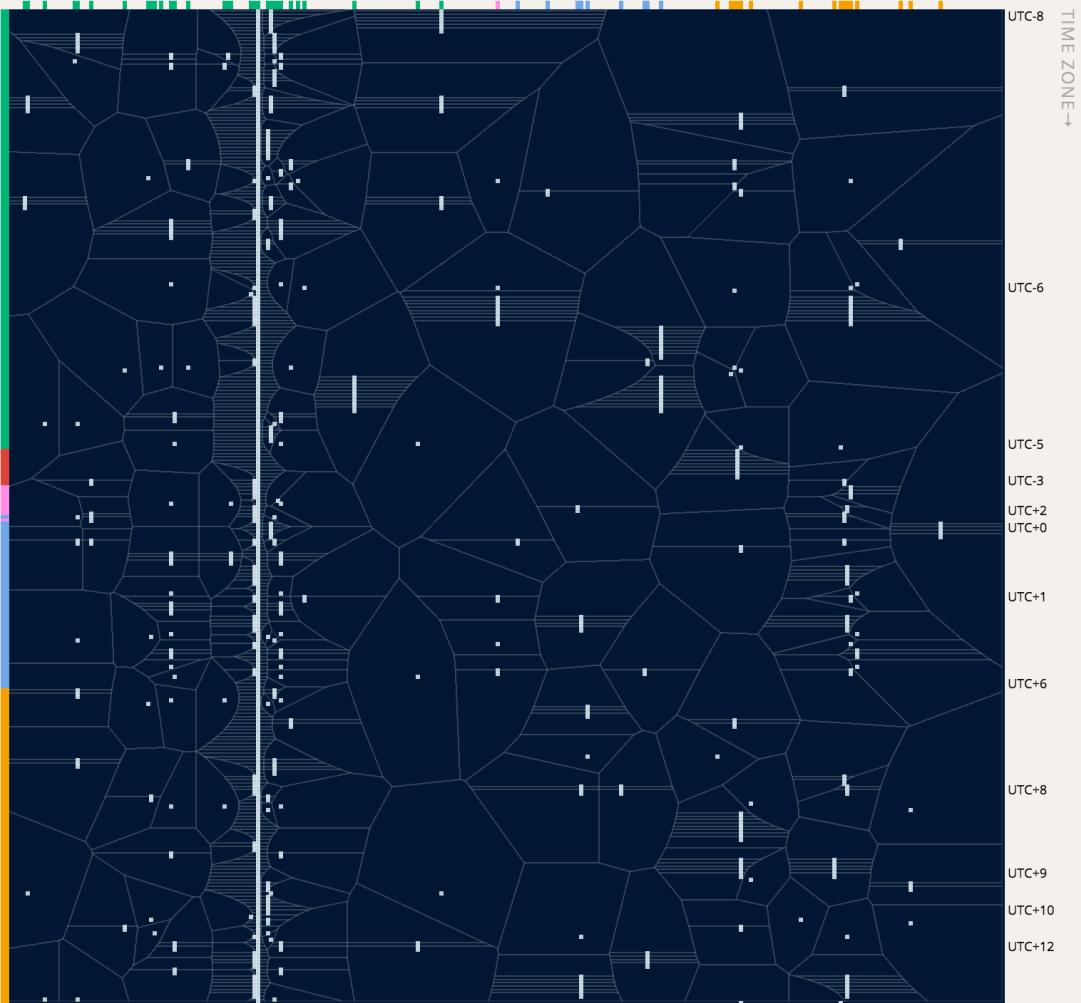
## Largest group in the same time zone

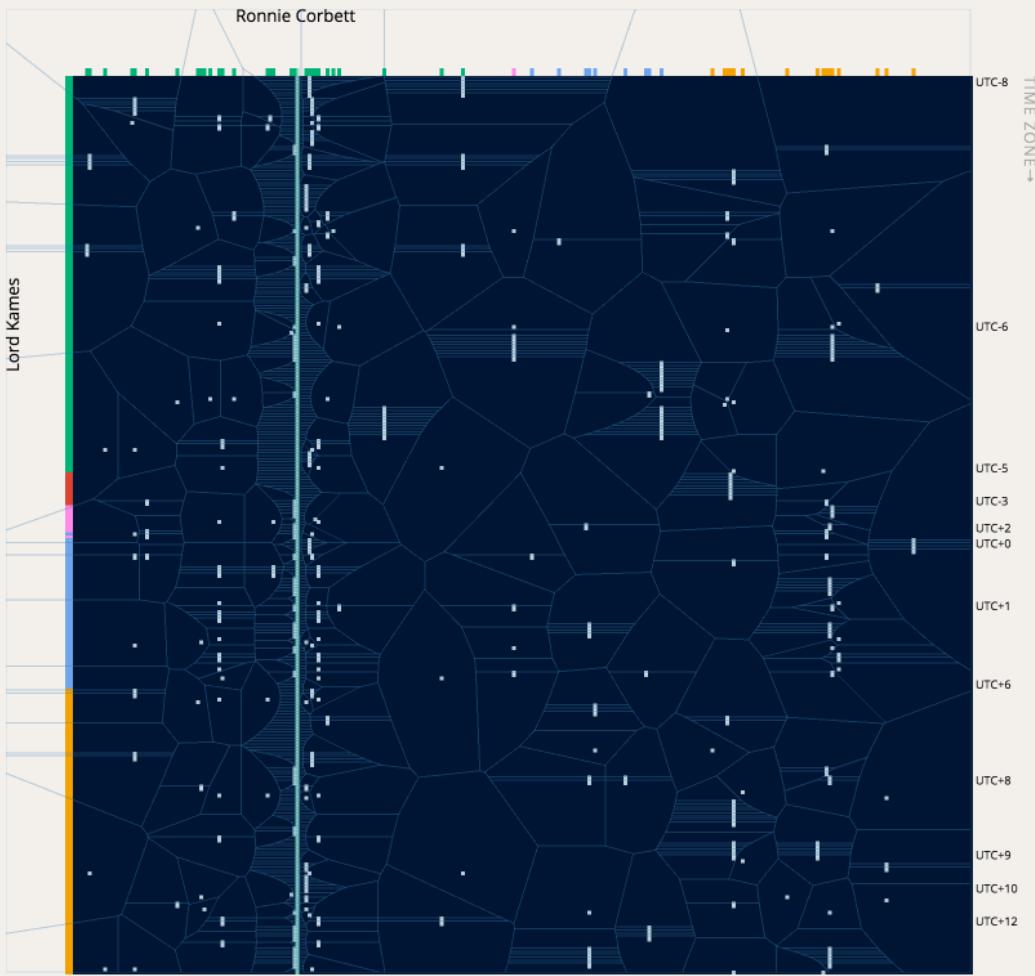
7 staff, 29% of team are in UTC+12



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.

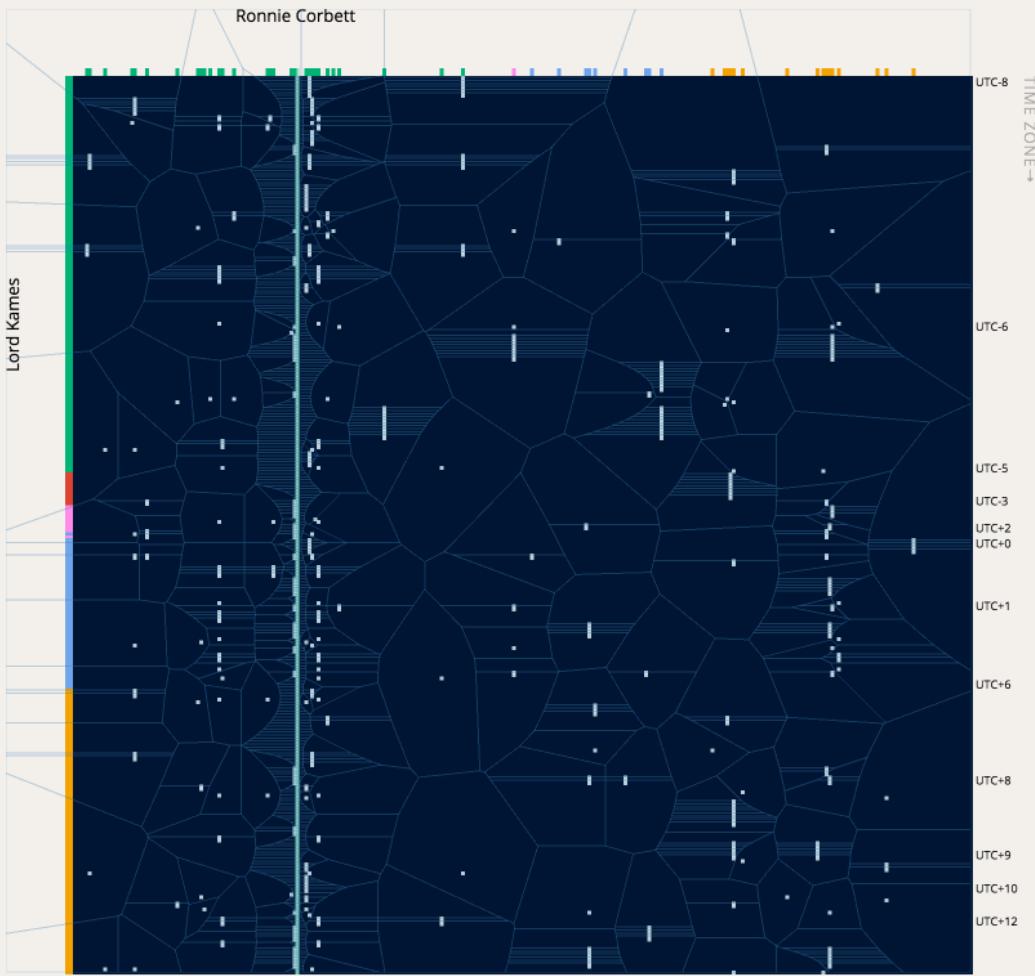






```
1 var dots_rollover = svg.append('g')
2     .attr('class', 'dot-rollover');
3
4 args.voronoi = d3.geom.voronoi()
5     .x(function(d) {
6         return args.x(d.to_employee);
7     })
8     .y(function(d) {
9         return args.y(d.from_employee);
10    })
11    .clipExtent([[0, 0],
12        [args.matrix_height - args.padding,
13            args.matrix_height - args.padding]])
14 );
```

```
1 var dots_rollover = svg.append('g')
2     .attr('class', 'dot-rollover');
3
4 args.voronoi = d3.geom.voronoi()
5     .x(function(d) {
6         return args.x(d.to_employee);
7     })
8     .y(function(d) {
9         return args.y(d.from_employee);
10    })
11    .clipExtent([[0, 0],
12                [args.matrix_height - args.padding,
13                 args.matrix_height - args.padding]])
14 );
```



```
1 dots_rollover.selectAll("path")
2     .data(args.voronoi(args.dependencies))
3     .enter().append('path')
4     .attr("d", function(d) {
5         return "M" + d.join("L") + "Z";
6     })
7     .attr("class", function(d) {
8         return "m" + d.point.to_employee
9             + " e" + d.point.from_employee;
10    })
11    .on('mouseover', dotMouseOver(args))
12    .on('click', dotClick(args));
```

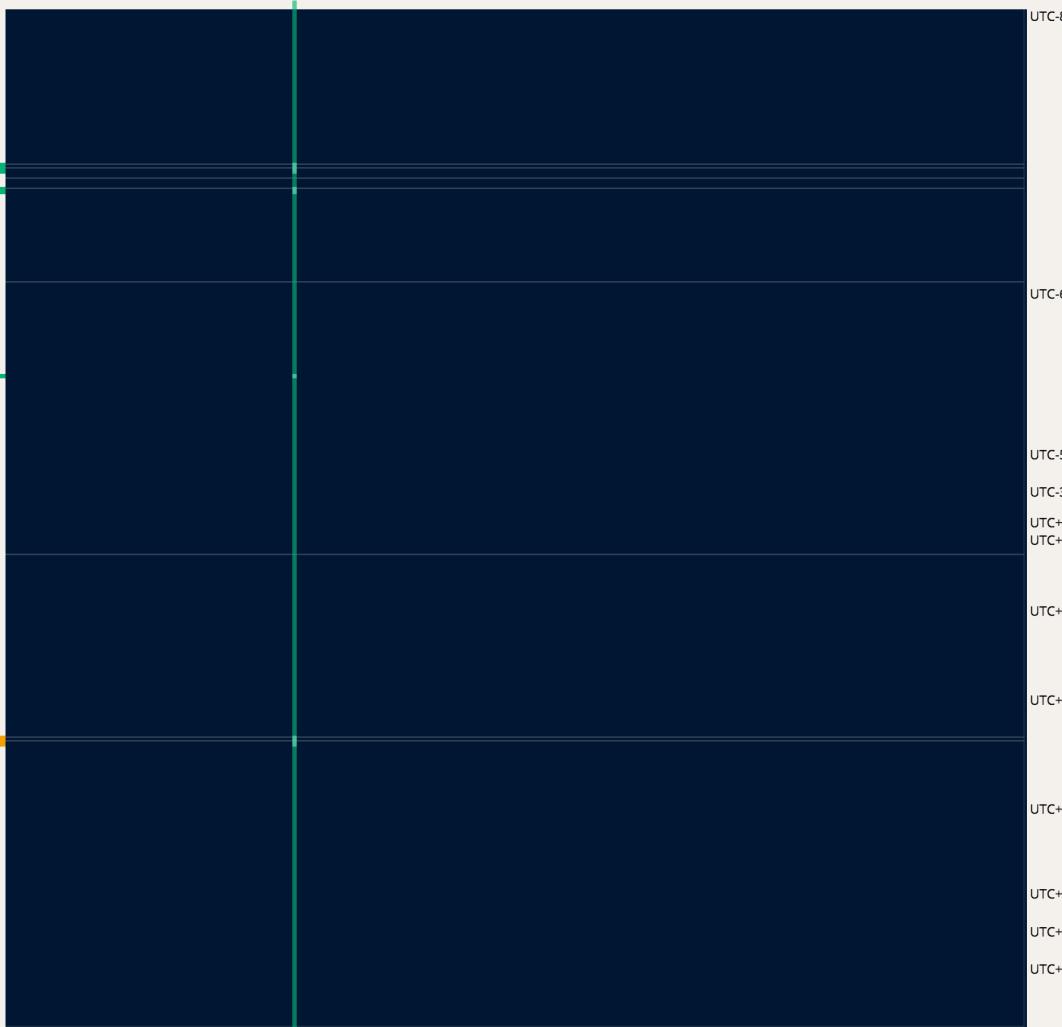
```
1 dots_rollover.selectAll("path")
2     .data(args.voronoi(args.dependencies))
3     .enter().append('path')
4     .attr("d", function(d) {
5         return "M" + d.join("L") + "Z";
6     })
7     .attr("class", function(d) {
8         return "m" + d.point.to_employee
9             + " e" + d.point.from_employee;
10    })
11    .on('mouseover', dotMouseOver(args))
12    .on('click', dotClick(args));
```

```
1 dots_rollover.selectAll("path")
|   .data(args.voronoi(args.dependencies))
2   .enter().append('path')
3   .attr("d", function(d) {
4       return "M" + d.join("L") + "Z";
5   })
6   .attr("class", function(d) {
7       return "m" + d.point.to_employee
8           + " e" + d.point.from_employee;
9   })
10  .on('mouseover', dotMouseOver(args))
11  .on('click', dotClick(args));
```

```
1 dots_rollover.selectAll("path")
2     .data(args.voronoi(args.dependencies))
3     .enter().append('path')
4     .attr("d", function(d) {
5         return "M" + d.join("L") + "Z";
6     })
7     .attr("class", function(d) {
8         return "m" + d.point.to_employee
9             + " e" + d.point.from_employee;
10    })
11    .on('mouseover', dotMouseOver(args))
12    .on('click', dotClick(args));
```

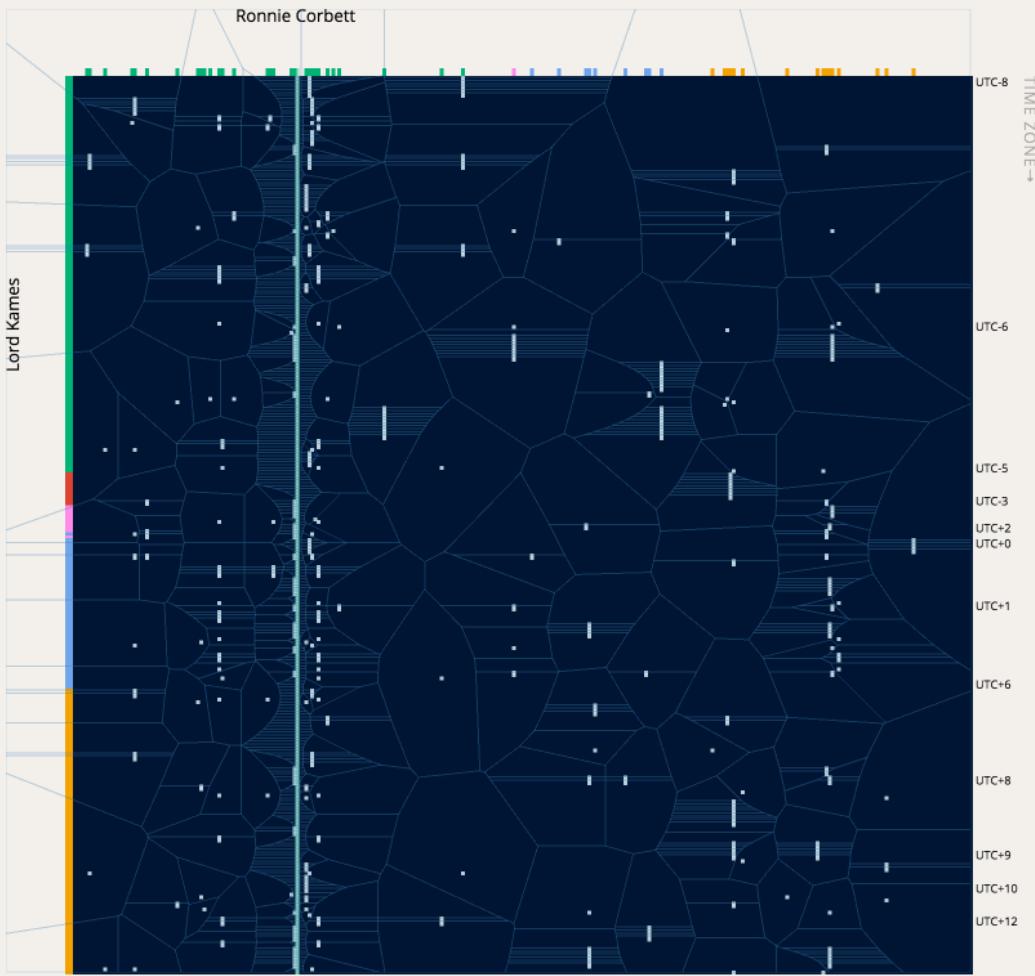
```
▼ Array[4] i
  ▼ 0: Array[2]
    0: 389.82358530029154
    1: 742.161074
    length: 2
    ► __proto__: Array[0]
  ▼ 1: Array[2]
    0: 478.9422818
    1: 742.161074
    length: 2
    ► __proto__: Array[0]
  ▼ 2: Array[2]
    0: 478.0080536
    1: 739.8255035
    length: 2
    ► __proto__: Array[0]
  ▼ 3: Array[2]
    0: 390.4432264716785
    1: 739.8255035
    length: 2
    ► __proto__: Array[0]
  length: 4
  ► __proto__: Object
  point: Object
    from_employee: "296"
    to_employee: "172"
    ► __proto__: Object
```

```
> $('path.e296.m172')
< [ <path d="M389.82358530029154,742.161074L478.9422818,742.161074L478.0080536,739.8255035L390.4432264716785,739.8255035Z" class="m172 e296"></path>]
```



```
// manager comes from the dropdown or from the clicked voronoi cell
1 var team = args.dependencies.filter(function(employee, i) {
2     return employee.to_employee === manager;
3 });
4
5 regenerateVoronoi(team);
```

```
▼ <html hola_ext_inject="disabled">
  ▶ <head>...</head>
  ▼ <body>
    ▼ <div class="container">
      ::before
    ▼ <div class="row">
      ::before
      ▶ <div class="col-md-3">...</div>
      ▼ <div class="col-md-8 matrix text-center">
        ▼ <div class="dsm hidden-sm hidden-xs">
          ▼ <svg width="800" height="800" viewBox="0 0 800 800" preserveAspectRatio="xMinYMin meet">
            <text class="to-employee" text-anchor="start" x="75.35570469798658" y="10">Nathaniel Culverwel</text>
            <text class="from-employee" text-anchor="middle" x="10" y="341.6107382550336" transform="rotate(-90 10 341.6107382550336)">...</text>
            <rect width="698" height="698" x="52" y="52"/>
            ▶ <g class="dot">...</g>
            ▶ <g class="timezone-labels">...</g>
              <text class="timezone-axis-description" text-anchor="start" x="790" y="52" transform="rotate(90 790,52)">Time zone--</text>
            ▶ <g class="timezone-y">...</g>
            ▶ <g class="timezone-x">...</g>
            ▶ <g class="dot-rollover">...</g>
          </svg>
        </div>
      ▶ <div class="legend hidden-sm hidden-xs">...</div>
      ▶ <div class="matrix-details">...</div>
      ▶ <div class="region-table">...</div>
    </div>
    <div class="col-md-1"></div>
```



1. Data
2. Scales
3. Data-join
4. Ticks
5. Voronoi
- 6. Events**
7. Transitions
8. Stats and charts

```
1 function dotMouseOver(args) {  
| 2     return function(d) {  
| 3         d = d.point;  
| 4         ...  
| 5     }  
| 6 }
```

```
1 function dotMouseOver(args) {  
2     ...  
3     d3.select('.to-employee')  
4         .attr('x', args.x(d.to_employee))  
5         .attr('y', 10)  
| 6     .text(args.employee_data[d.to_employee].name)  
7     .attr('text-anchor', function() {  
8         if (d.to_employee < 20) {  
9             return 'start';  
10        } else if (d.to_employee > args.employee_count - 20) {  
11            return 'end';  
12        } else {  
13            return 'middle';  
14        }  
15    });  
16}  
17...  
18}
```

```
1  function dotMouseOver(args) {
2      return function(d) {
| 3          console.log(d3.select(this));
4          d = d.point;
5          ...
6      }
7  }
```

Elements Network Sources Timeline Profiles Resources Audits Console

✖ <top frame> ▾  Preserve log

▼ [Array[1]] 1

  ▼ 0: Array[1]

    ▼ 0: path

      ▼ \_\_data\_\_: Array[5]

        ► 0: Array[2]

        ► 1: Array[2]

        ► 2: Array[2]

        ► 3: Array[2]

        ► 4: Array[2]

        length: 5

      ▼ point: Object

        from\_employee: "228"

        to\_employee: "21"

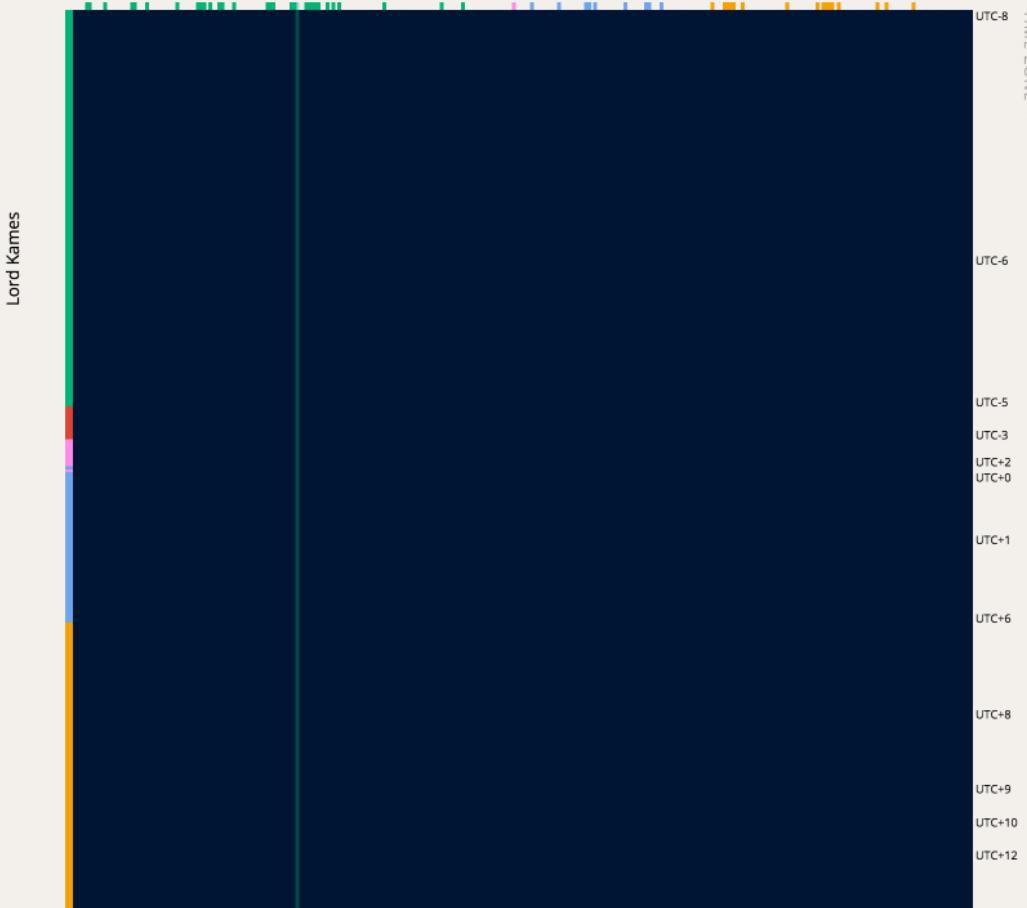
        ► \_\_proto\_\_: Object

        ► \_\_proto\_\_: Array[0]

        ► onclick: function (e)

        ► onmouseover: function (e)

1. Data
2. Scales
3. Data-join
4. Ticks
5. Voronoi
6. Events
7. **Transitions**
8. Stats and charts



```
1 dots.attr('width', 0)
2     .attr('height', 0)
3     .transition()
4         .delay(1000)
5         .duration(function(d, i) {
6             return i * Math.random() % 2000;
7         })
8         .attr('width', 1)
9         .attr('height', 1);
```

## Data

```
{  
  "year": "1975",  
  "sightings": 279  
},  
{  
  "year": "1976",  
  "sightings": 246  
},  
{  
  "year": "1977",  
  "sightings": 239  
},  
{  
  "year": "1978",  
  "sightings": 301  
},  
{  
  "year": "1979",  
  "sightings": 221  
},  
{  
  "year": "1980",  
  "sightings": 211  
},  
{
```

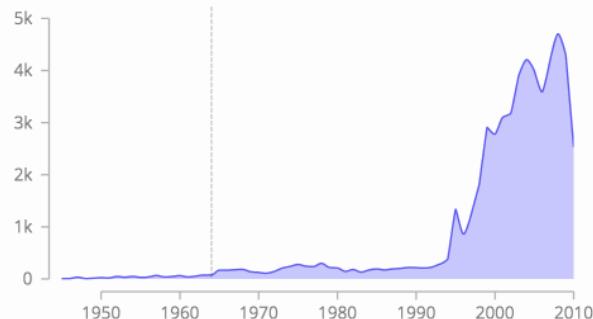
## JavaScript

[See Full List of Options](#)[Run](#)

```
1 MG.data_graphic({  
2   title: "UFO Sightings",  
3   description: "Yearly UFO sightings from 1945 to 2010.",  
4   data: JSON.parse($('.data textarea').val()),  
5   markers: [{year: 1964, label: '"The Creeping Terror" released'},  
6     width: 400,  
7     height: 250,  
8     target: ".result",  
9     x_accessor: "year",  
10    y_accessor: "sightings",  
11    interpolate: "monotone"  
12  }],  
13});
```

UFO Sightings ?

"The Creeping Terror" released



```
var line = d3.svg.line()  
  .x(args.scalefns.xf)  
  .y(args.scalefns.yf)  
  .interpolate(args.interpolate)  
  .tension(args.interpolate_tension);
```

```
1 existing_line  
2   .transition()  
3   .duration(args.transition_duration)  
4   .attr('d', line(args.data));
```

1. Data
2. Scales
3. Data-join
4. Ticks
5. Voronoi
6. Events
7. Transitions
8. Stats and charts

# Ronnie Corbett

Manages Lord Kames

Team size 299

## Staff in each timezone

17 (median), 25 (mean)

## Potential lost hours\*

20 hours (between staff)

0 hours (between manager and staff)

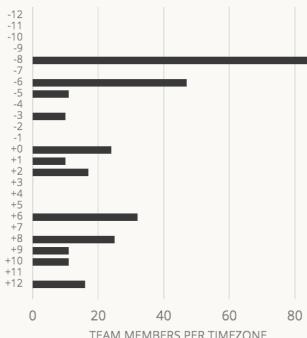
## Maximum time difference

20 hours (between staff)

20 hours (between manager and staff)

## Largest group in the same time zone

85 staff, 28% of team are in UTC-8



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.

# Ronnie Corbett

Manages Lord Kames

Team size 299

## Staff in each timezone

17 (median), 25 (mean)

## Potential lost hours\*

20 hours (between staff)

0 hours (between manager and staff)

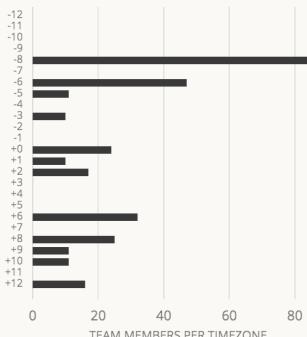
## Maximum time difference

20 hours (between staff)

20 hours (between manager and staff)

## Largest group in the same time zone

85 staff, 28% of team are in UTC-8



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.

## Ronnie Corbett

Manages Lord Kames

Team size 299

### Staff in each timezone

17 (median), 25 (mean)

### Potential lost hours\*

20 hours (between staff)

0 hours (between manager and staff)

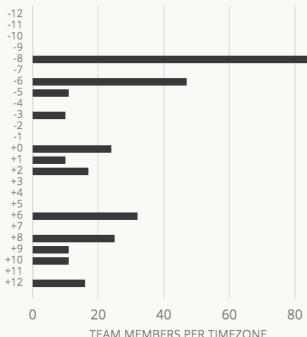
### Maximum time difference

20 hours (between staff)

20 hours (between manager and staff)

### Largest group in the same time zone

85 staff, 28% of team are in UTC-8



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.

# Ronnie Corbett

Manages Lord Kames

Team size 299

## Staff in each timezone

17 (median), 25 (mean)

## Potential lost hours\*

20 hours (between staff)

0 hours (between manager and staff)

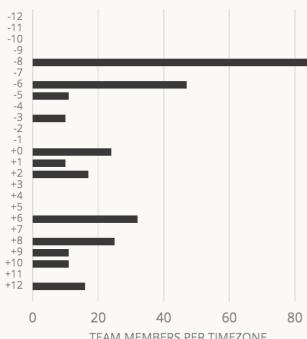
## Maximum time difference

20 hours (between staff)

20 hours (between manager and staff)

## Largest group in the same time zone

85 staff, 28% of team are in UTC-8



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.

```
d3.min  
d3.max  
d3.mean  
d3.median  
d3.quantile
```

# Arthur Collier

Manages Friedrich Engels

Team size 18

## Staff in each timezone

2 (median), 3 (mean)

## Potential lost hours\*

14 hours (between staff)

14 hours (between manager and staff)

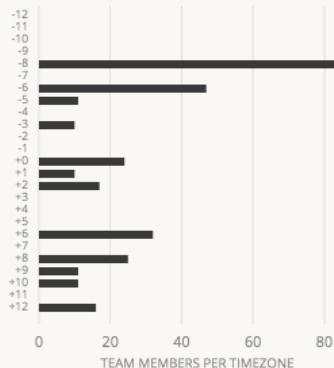
## Maximum time difference

20 hours (between staff)

20 hours (between manager and staff)

## Largest group in the same time zone

6 staff, 33% of team are in UTC+6



\* The difference between the timezone with the largest group of staff (mode) and the farthest timezone from the mode.

```
// show team members across timezones for this manager
1 MG.data_graphic({
2     data: args.managers_team[d.to_employee].timezones,
3     target: '.managers-team-across-timezones',
4     x_accessor: 'value',
5     y_accessor: 'key',
6     chart_type: 'bar',
6     max_x: 85
8});
```

Elements: 863, Density: 0.34%  
 Total staff: 299, Distance to CEO (mean, median): 3  
 Team sizes: 15 (mean), 5 (median), 2 (mode)

	Asia Pacific	Canada	EMEA	Latin America	USA
Staff by region:	32%	4%	17%	3%	44%
Managers by region:	28%	0%	16%	2%	54%

This is an interactive Design Structure Matrix (DSM) where a dot indicates that the employee along the left side reports to the employee along the top. The DSM is a visibility matrix, which means that in addition to showing direct reports, we also show indirect reports. Hence, an employee is shown as reporting to his or her direct manager, as well as to his or her manager's manager and so on. We achieve this through matrix multiplication.

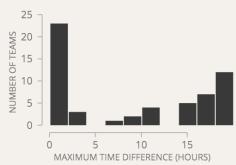
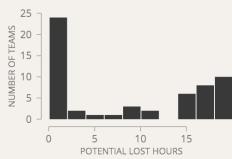
The term for the number of managers that an employee reports to, directly or indirectly, is *fan-out* and the term for the number of staff that report to a particular manager is *fan-in*. The manager with the largest team is hence the manager with the largest fan-in, and taking an average of the fan-out for all staff gives us a sense of the average distance to the CEO.

Depending on how the matrix is sorted, different clusters of dots indicate different things. When we sort the matrix by timezone, then location then manager—the current case—vertical lines show staff who are either physically close to each other or in the same timezone. The more spread out a vertical line is, the more spread out a manager's team members are.

**You can mouse-over the matrix to reveal details about its various parts or view details about a particular manager's team by clicking anywhere on the matrix or by choosing a manager from the dropdown.**

## How spread out are teams?

Of the 57 teams at INITECH, 23 teams are entirely in the same timezone or within two hours of the same timezone; 12 teams have a maximum time difference of 18 to 20 hours.



Here is a look at the most spread out of INITECH's 57 teams:

	Manager	Team size	Maximum time difference
1.	Ronnie Corbett	299	20 hours
2.	Joseph Glanvill	71	20 hours
3.	Ito Jinsai	42	20 hours
4.	Henri Bergson	29	20 hours
5.	Francis Hutcheson	26	20 hours
6.	Arthur Collier	18	20 hours
7.	Huang Zongxi	30	18 hours
8.	Pierre Daniel Huet	28	18 hours
9.	Louis de La Forge	10	18 hours
10.	Friedrich Schelling	7	18 hours
11.	Rudolf Bultmann	2	18 hours
12.	Nathaniel Culverwel	2	18 hours
13.	Archibald Alison	17	17 hours
14.	Gershom Carmichael	6	17 hours
15.	Kumazawa Banzan	30	16 hours
16.	Pierre Azais	5	16 hours
17.	Bertrand Russell	4	16 hours
18.	John Locke	3	16 hours
19.	Arnold Geulincx	2	16 hours
20.	Percy Williams Bridgman	24	15 hours

Code and demo

*github.com/almossawi/d3-matrix*

D3 is a useful low-level toolkit

Portable

*SVG deployed to the Web*

Interactive

*You can show your data in layers*

Free

*No licenses*

Extensible

*HTML, JavaScript, SVG, CSS*

What do we want to visualize?

How do we intend to visualize it?

What do we want to visualize?

How do we intend to visualize it?

How might we think about organizational structure?

*Our broad question*

## **How might we think about organizational structure?**

1. How are employees distributed throughout the organization?
2. How flat is the organization?
3. Which teams are the most spread-out?
4. How big are teams?
5. To what extent are offices used?

“We found that one-third of people, regardless of profession, do their job because they see a **meaning** in it.”

Have you heard about the janitors at that coma ward who change the wall art in patients’ rooms?

“[Top teams] achieve superior levels of participation, cooperation, and collaboration because their members **trust** one another... and have confidence in their effectiveness as a team.”

“[That’s] the biggest predictor of team success.”

“We identified three aspects of communication that affect team performance.”

“The most valuable form of communication is **face-to-face**. The next most valuable is by phone or videoconference, but with a caveat...”

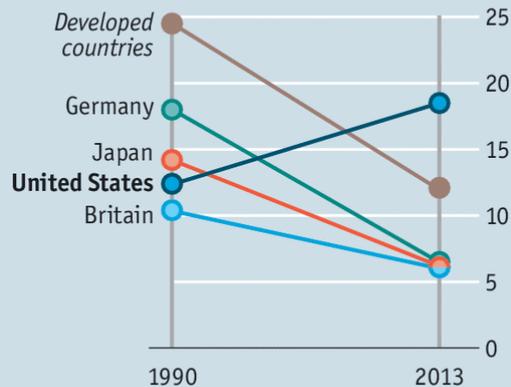
“Products tend to mirror the architectures of the organizations  
in which they are developed.”

What do we want to visualize?

How do we intend to visualize it?

## Odd one out

Maternal-mortality rate, per 100,000 live births



Source: Kassebaum *et al*, *Lancet*

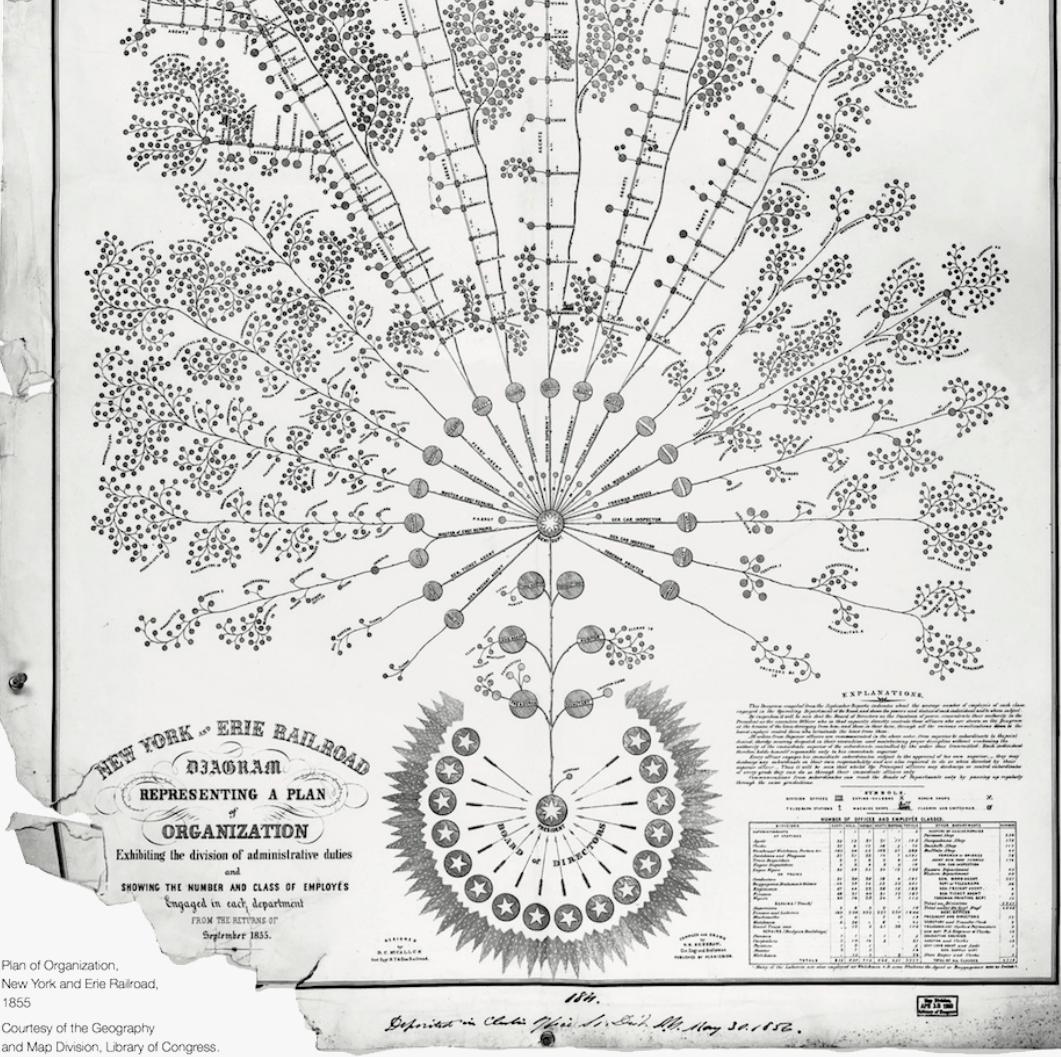
Economist.com

A graphic is often meant to maximize **understandability**

Ideally, we want the reader to grasp our work with minimal investment

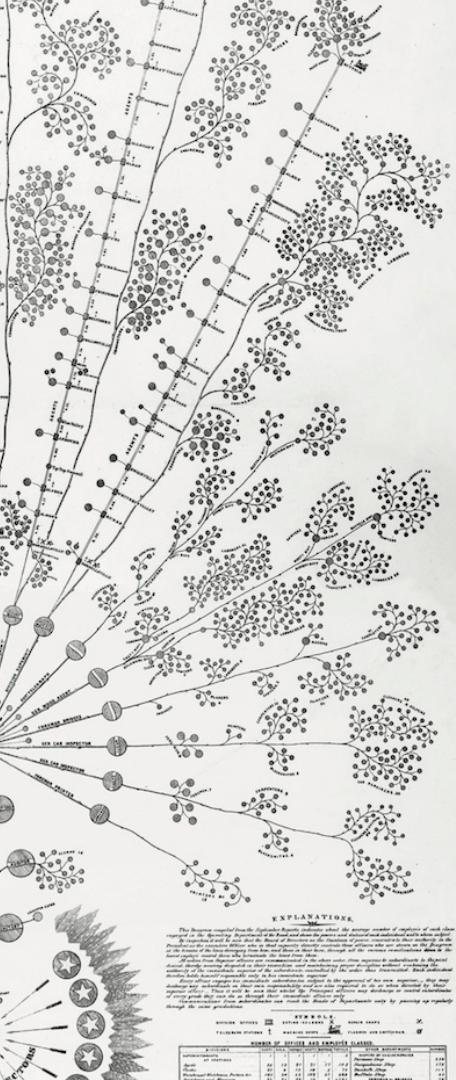
Any extra acts of cognition that users are made to perform  
incur a **debt** that we must then relieve in the form of increased **value**  
*More sophisticated insights or the affordance of particular qualities*

The path from the familiar to the novel ought to be  
fraught with **caution** and **reflection**



Plan of Organization,  
New York and Erie Railroad,  
1855

Courtesy of the Geography  
and Map Division, Library of Congress.



It looks nice

It fits with our mental model of hierarchy

At a glance, it gives us a sense of how big and how deep the organization is

It's not compact

It's difficult to compare multiple charts

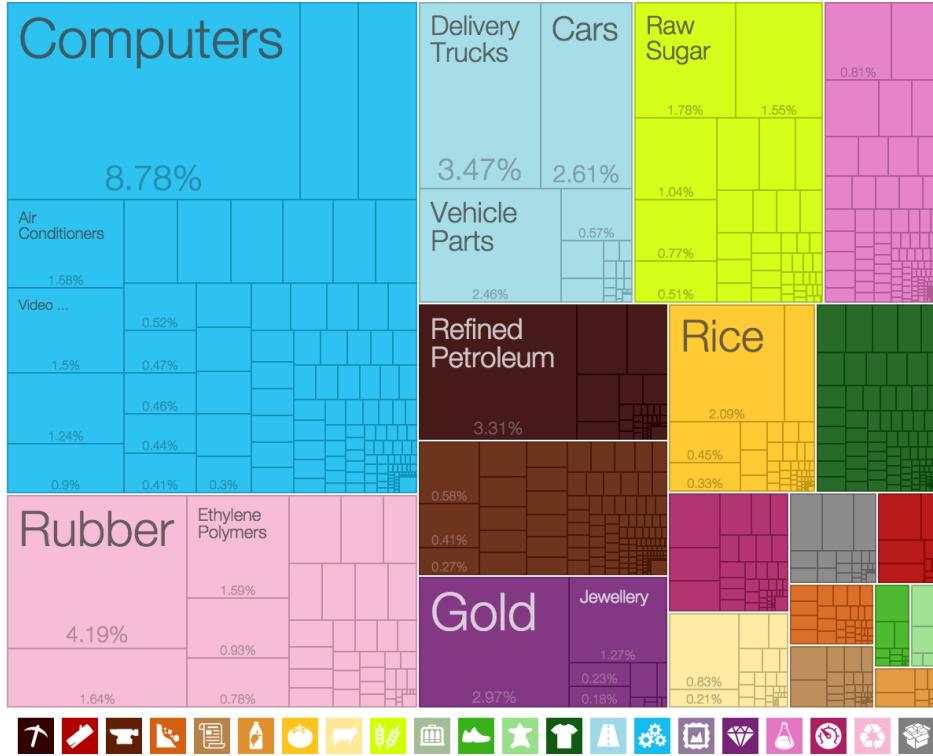
It doesn't really help us answer our questions



Mike Bostock

# Computers

8.78%



Alexander Simoes

	<i>Tree</i>	<i>Sunburst</i>	<i>Treemap</i>
<i>Answers our questions?</i>	3 out of 5	2 out of 5	2 out of 5
<i>Compact?</i>	Not really	Kind of	Yes
<i>Comparability?</i>	Not really	Kind of	Kind of

- 
1. How are employees distributed throughout the organization?
  2. **How flat is the organization?**
  3. **Which teams are the most spread-out?**
  4. **How big are teams?**
  5. To what extent are offices used?

An organization is a system

*Albeit, its fundamental unit is a person rather than a component*



Complexity

Internal dynamics

External Influences

Potential to degrade with time

	<i>Tree</i>	<i>Sunburst</i>	<i>Treemap</i>	<i>DSM</i>
<i>Answers our questions?</i>	3 out of 5	2 out of 5	2 out of 5	4 out of 5
<i>Compact?</i>	Not really	Kind of	Yes	Yes
<i>Comparability?</i>	Not really	Kind of	Kind of	Yes

- 
1. How are employees distributed throughout the organization?
  2. How flat is the organization?
  3. Which teams are the most spread-out?
  4. How big are teams?
  5. To what extent are offices used?

How are employees distributed throughout the organization?

*sort order*

How flat is the organization?

*average horizontal cardinality (fan-out)*

Which teams are the most spread-out?

*vertical spread*

How big are teams?

*vertical cardinality (fan-in)*

To what extent are offices used?

—

Final thought

“The #1 neglected topic in statistics is measurement...  
the connection between the data you gather and the  
underlying object of your study”

Andrew Gelman, What's the most important thing in statistics that's not in the textbooks?

Measuring  $\in$  Thinking

It turns out that thinking is quite important :)

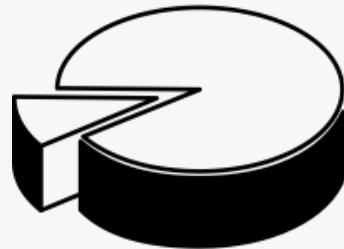
## Encodings

*Position, area, color, chart types*

Maybe I shouldn't show everything at once

*Maybe I shouldn't show everything*

*Who is this for?*



- ♪ *How could this happen to me?*
- ♪ *I've made my mistakes...*

almossawi.com/sfhtml5



@alialmossawi