

# Oh No! Not Another Data Grid!



elastic

Carly Richmond

@CarlyLRichmond





Mentor

Scrum  
Master

People  
Manager

Axiom  
Developer

.NET  
Developer

Developer  
Advocate

Tech Lead

Frontend  
Developer

Agile  
Coach

7/8

3/8



**WE WANT YOU**

**DASHBOARDS**

Select a customer

Anvil

Show Delivered Orders

Customer Name	Placement Date	Order ID	Product	Status	Purchase Price (£)
M. Martian	Fri, 3 June 2022	579602348	_super Giant Rubber Band	Shipped	84.45
D. Duck	Mon, 16 May 2022	496577906	Giant Rubber Band	Shipped	146.62
B. Bunny	Fri, 13 May 2022	375698743	Anvil	Shipped	84.35
R. Runner	Wed, 11 May 2022	799094661	Anvil	Shipped	114.91
B. Bunny	Mon, 4 April 2022	1796566218	Anvil	Delivered	120.78
D. Duck	Sat, 19 March 2022	1515565216	Gift Card	Delivered	127.59
B. Bunny	Sat, 19 March 2022	1143625701	Anvil	Shipped	132.59
D. Duck	Wed, 9 March 2022	1819106314	_super Giant Rubber Band	Shipped	102.13
B. Bunny	Thu, 17 February 2022	1575645063	Gift Card	Shipped	131.51
W. E. Coyote	Sun, 9 January 2022	1868646898	Gift Card	Shipped	8.1
D. Duck	Fri, 7 January 2022	1765230146	_super Giant Rubber Band	Shipped	119
D. Duck	Fri, 7 January 2022	1852382397	Gift Card	Delivered	48.74
P. Le Pew	Tue, 4 January 2022	562426314	Anvil	Shipped	<input type="checkbox"/> Copy Ctrl+C
M. Martian	Thu, 30 December 2021	1633453698	_super Giant Rubber Band	Shipped	<input type="checkbox"/> Copy with Headers
D. Duck	Sun, 26 December 2021	626772355	Gift Card	Shipped	<input type="checkbox"/> Copy with Group Headers
D. Duck	Mon, 29 November 2021	1839740054	_super Giant Rubber Band	Delivered	Ctrl+V
B. Bunny	Tue, 16 November 2021	254371477	Anvil	Shipped	<input type="checkbox"/> Paste
D. Duck	Tue, 9 November 2021	1237282028	Gift Card	Delivered	<input type="checkbox"/> Export
R. Runner	Sun, 7 November 2021	839368045	_super Giant Rubber Band	Shipped	113.17
W. E. Coyote	Thu, 4 November 2021	350809083	_super Giant Rubber Band	Delivered	26.41
P. Le Pew	Sun, 17 October 2021	324369422	Anvil	Delivered	121.03
W. E. Coyote	Thu, 7 October 2021	1801920898	_super Giant Rubber Band	Shipped	44.67
B. Bunny	Sun, 5 September 2021	1481659071	Giant Rubber Band	Delivered	57.58

# What's the user outcome



A large, golden statue of Lady Justice, also known as Justitia or Iustitia, is positioned on the left side of the frame. She is blindfolded, wearing a traditional crown of justice, and holds a set of scales in her raised right hand. Her left hand rests on a sword sheath at her waist. The statue is highly detailed, showing intricate patterns on her clothing and armor.

Don't Judge So Hastily!



Safari File Edit View History Bookmarks Window Help

DraftWire 2016 NFL Scouting Combine Data

Name College Pos. Conf. Height Weight Class Arms Hands Bench Press 40' Time-10' 40' 20'

Name	College	Pos.	Conf.	Height	Weight	Class	Arms	Hands	Bench Press	40' Time-10'	40' 20'
Westerman, Stephen	Vanderbilt	OT	SEC	6-6	304	Junior	34 3/2	10 3/8	20	5.20	
Beavers, Willie	Wesleyan Michigan	OT	Mid-American	6-6	324	Senior	33 3/2	9 3/2	20	5.37	
Benavoch, Caleb	UCLA	OT	Big 12	6-5	305	Junior	35 1/2	10 5/8	22	5.14	
Clark, Le Rayen	Texas Tech	OT	Big 12	6-5	307	Senior	36 1/8	10 7/8	18	5.14	
Coleman, Shon	Mississippi St.	OT	Big Ten	6-6	303	Junior	35 1/8	10 1/8	20	5.09	
Condit, Jack	USC	OT	Big Ten	6-7	310	Senior	34 3/8	10	20	5.21	
Couper, Ethan	Florida Atlantic	OT	Big Ten	6-6	307	Junior	34 3/8	10	19	5.2	
Huang, Eric	North Dakota St.	OT	Northern Illinois	6-6	307	Senior	33 1/2	9 3/8	14	5.14	
Hawkins, Jerrad	LSU	OT	SEC	6-6	305	Junior	34 1/8	9 7/8	23	5.27	
Johndonne, Tyler	Oregon	OT	Big 12	6-7	303	Senior	34 1/8	10 1/8	20	5.17	
Kirkland, Demetrius	Arkansas	OT	SEC	6-6	311	Junior	34 1/8	10 1/8	20	5.04	
Lewis, Jason	Notre Dame	OT	Big Ten	6-6	312	Senior	34	10	22	5.24	
Miles, Tyler	Wisconsin	OT	Big Ten	6-7	316	Senior	33 1/2	10	19	5.51	
Murphy, Kyle	Stanford	OT	Big 12	6-6	307	Senior	33 1/2	9 3/4	23	5.32	
Neimbot, Stephanie	Colorado	OT	Big 12	6-6	312	Senior	34 1/8	10 1/4	28	5.4	
Nguyen, Dang	Georgia	OT	Georgia	6-5	324	Senior	36	10 3/8	30		
Shull, Brandon	South Carolina	OT	SEC	6-5	324	Senior	34 3/4	10 3/8	22	5.2 (hammy)	
Slater, Pearce	San Diego St.	OT	Mountain West	6-7	313	Senior	34 3/8	10 3/8	17	5.41	
Santiago, Jessie	UT	OT	Big 12	6-6	304	Junior	33 1/8	10 5/8	21	4.92	
Steketee, Benito	Notre Dame	OT	IA Independents	6-6	312	Junior	35 1/8	10 5/8	26		

Combine prospect measurements Quarterbacks Running Backs Wide Receivers Tight Ends Offensive Line





A collage of various antique and modern clocks from different angles and perspectives. The collection includes large grandfather-style clocks, smaller wall and mantel clocks, and a prominent 24-hour clock in the foreground. The clocks are made of wood, metal, and glass, with faces featuring Roman numerals and Arabic numbers. The background is dark, making the light-colored clock faces stand out.

# Time to Market

# What's Their Desired Outcome?

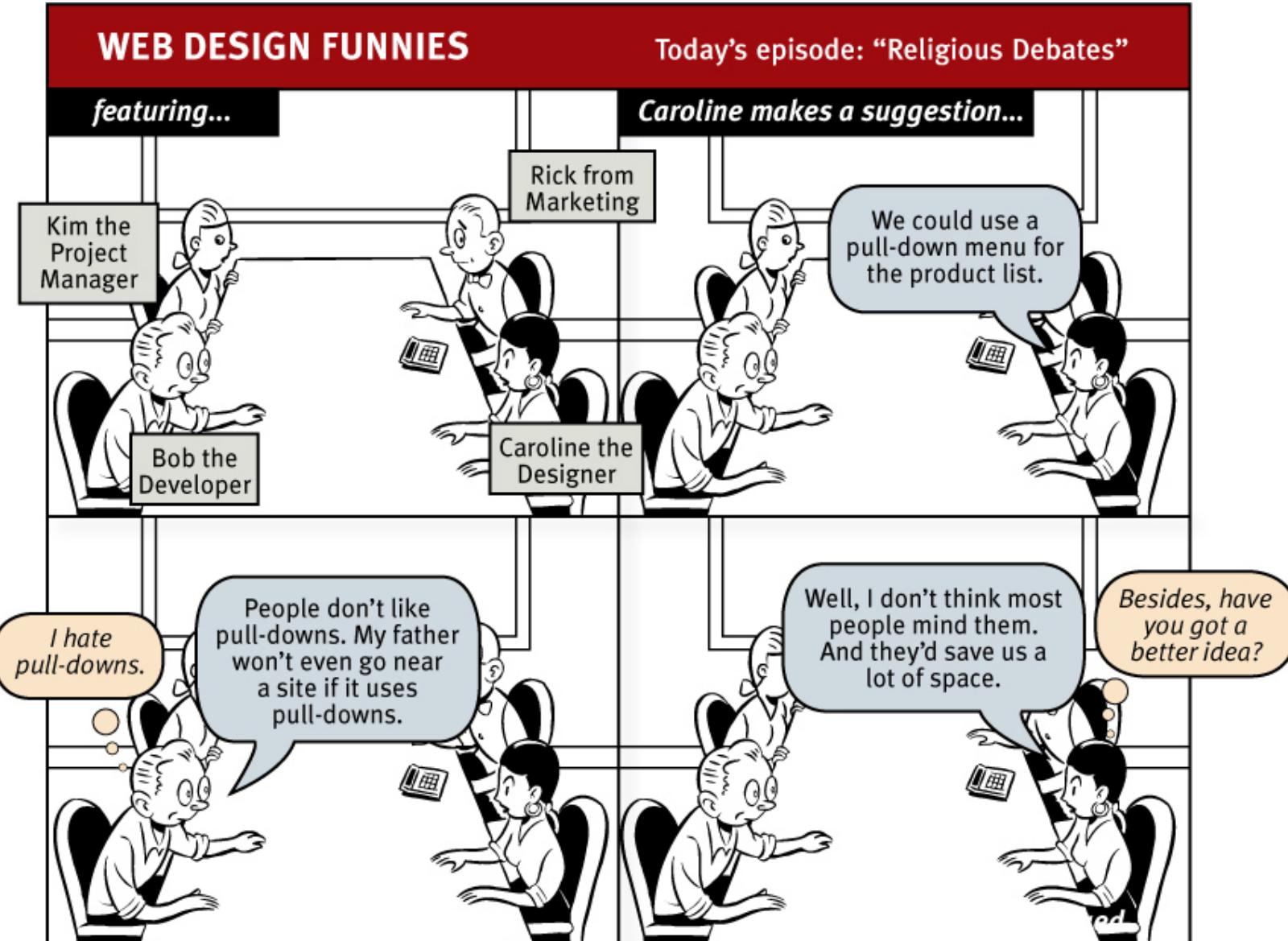


## WEB DESIGN FUNNIES

Today's episode: "Religious Debates"

*featuring...*

*Caroline makes a suggestion...*









# Sorting

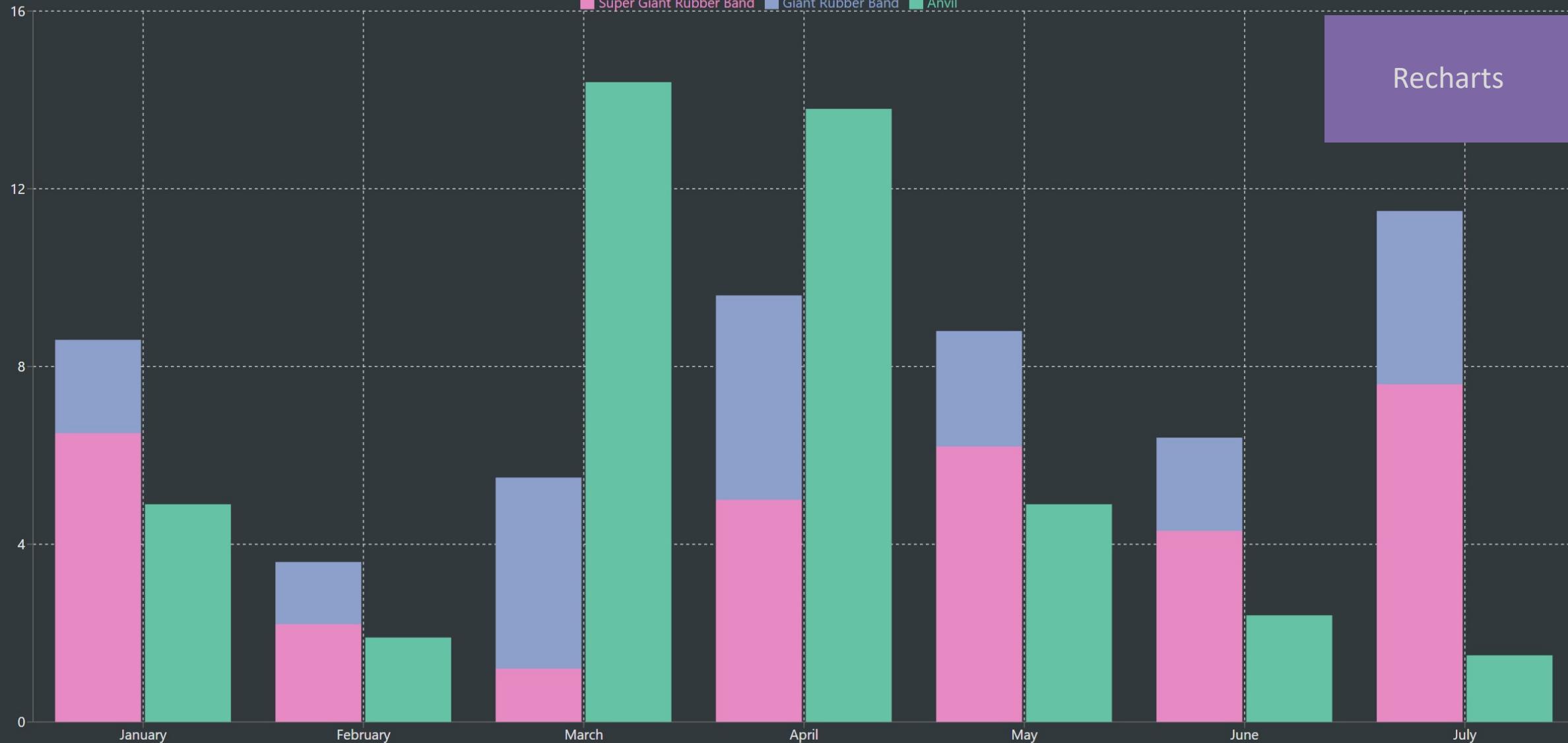
Example available via speaker's [GitHub](#)

## W.E. Coyote

Net Count, January-July 2021

■ Super Giant Rubber Band ■ Giant Rubber Band ■ Anvil

Recharts





## ProductColumnChart.tsx

```
1 import React, { PureComponent } from 'react';
2 import { BarChart, CartesianGrid, Legend, Tooltip, XAxis, YAxis, Bar, ResponsiveContainer } from 'recharts';
3
4 export default class ProductColumnChart extends PureComponent {
5     /*Other methods and properties omitted*/
6
7     render() {
8         return (
9             <ResponsiveContainer minWidth={700} height={height}>
10                <BarChart
11                    data={this.state.data}
12                    margin={{{
13                        top: 20,
14                        right: 30,
15                        left: 30,
16                        bottom: 5,
17                    }}}
18                    <CartesianGrid strokeDasharray="3 3" />
19                    <XAxis dataKey="name"/>
20                    <YAxis/>
21                    <Tooltip/>
22                    <Legend/>
23                    <Bar name="Super Giant Rubber Band" dataKey="superGiantRubberBandCount" stackId="a"/>
24                    <Bar name="Giant Rubber Band" dataKey="giantRubberBandCount" stackId="a"/>
25                    <Bar name="Anvil" dataKey="anvilCount"/>
26                </BarChart>
27            </ResponsiveContainer>
28        )
29    }
30 }
```

```
        )
      )
    )
  )
}

return (
  <ResponsiveContainer minWidth={700} height={height}>
    <BarChart
      data={this.state.data}
      margin={{{
        top: 20,
        right: 30,
        left: 30,
        bottom: 5,
      }}}
    >
      <CartesianGrid strokeDasharray="3 3" />
      <XAxis dataKey="name"/>
      <YAxis/>
      <Tooltip/>
      <Legend/>
    </BarChart>
  </ResponsiveContainer>
)
```

```
    left: 30,  
    bottom: 5,  
  } }>  
  
<CartesianGrid strokeDasharray="3 3" />  
<XAxis dataKey="name"/>  
<YAxis/>  
<Tooltip/>  
<Legend/>  
<Bar name="Super Giant Rubber Band" dataKey="superGiantRubberBandCount"/>  
<Bar name="Giant Rubber Band" dataKey="giantRubberBandCount"/>  
<Bar name="Anvil" dataKey="anvilCount"/>  
  </BarChart>  
</ResponsiveContainer>  
)  
}
```

```
strokeDasharray="3 3" />
    me"/>

    giant Rubber Band" dataKey="superGiantRubberBandCount" stackId="a"/>
    rubber Band" dataKey="giantRubberBandCount" stackId="a"/>
    dataKey="anvilCount"/>
```



## ProductColumnChart.tsx

```
1 import React, { PureComponent } from 'react';
2 import { BarChart, CartesianGrid, Legend, Tooltip, XAxis, YAxis, Bar, ResponsiveContainer } from 'recharts';
3
4 export default class ProductColumnChart extends PureComponent {
5     /*Other methods and properties omitted*/
6
7     render() {
8         return (
9             <ResponsiveContainer minWidth={700} height={height}>
10                <BarChart
11                    data={this.state.data}
12                    margin={{{
13                        top: 20,
14                        right: 30,
15                        left: 30,
16                        bottom: 5,
17                    }}}
18                    <CartesianGrid strokeDasharray="3 3" />
19                    <XAxis dataKey="name"/>
20                    <YAxis/>
21                    <Tooltip/>
22                    <Legend/>
23                    <Bar name="Super Giant Rubber Band" dataKey="superGiantRubberBandCount" stackId="a"/>
24                    <Bar name="Giant Rubber Band" dataKey="giantRubberBandCount" stackId="a"/>
25                    <Bar name="Anvil" dataKey="anvilCount"/>
26                </BarChart>
27            </ResponsiveContainer>
28        )
29    }
30 }
```



## StackedProductBarChart.tsx

```
1 import React from 'react';
2 import Highcharts from 'highcharts';
3 import HighchartsReact from 'highcharts-react-official';
4
5 export default class StackedProductBarChart extends React.Component {
6     /*Other methods and properties omitted*/
7
8     render() {
9         const chartOptions: Highcharts.Options = {
10             xAxis: {
11                 categories: ['January', 'February', 'March']
12             },
13             plotOptions: {
14                 bar: {
15                     stacking: 'normal'
16                 }
17             },
18             series: [
19                 { name: 'Anvil', data: [1, 2, 3] },
20                 { name: 'Giant Rubber Band', data: [1, 2, 3] },
21                 { name: 'Super Giant Rubber Band', data: [1, 2, 3] }
22             ]
23         };
24
25         return (
26             <HighchartsReact
27                 highcharts={Highcharts}
28                 options={chartOptions}
29             />
30         )
31     }
32 }
```

```
20          { name: 'Giant Rubber Band' , data: [1, 2
21          { name: 'Super Giant Rubber Band' , data:
22      ]
23  };
24
25  return (
26      <HighchartsReact
27          highcharts={Highcharts}
28          options={chartOptions}
29      />
30  )
31 }
32 }
```

```
port default class StackedProductBarChart extends Re
/*Other methods and properties omitted*/
render( ) {
  const chartOptions: Highcharts.Options = {
    xAxis: {
      categories: [ 'January', 'February', 'March' ]
    },
    plotOptions: {
      bar: {
        stacking: 'normal'
      }
    },
    series: [
      {
        name: 'Electronics',
        data: [ 10, 15, 20 ]
      },
      {
        name: 'Clothing',
        data: [ 20, 25, 30 ]
      },
      {
        name: 'Food & Beverage',
        data: [ 30, 35, 40 ]
      }
    ]
  };
  return (
    

);
}


```

```
categories: ['January', 'February', 'March']
},
plotOptions: {
    bar: {
        stacking: 'normal'
    }
},
series: [
    { name: 'Anvil', data: [1, 2, 3] },
    { name: 'Giant Rubber Band', data: [1, 2, 3] },
    { name: 'Super Giant Rubber Band', data: [1, 2, 3] }
]
};

return (
    <HighchartsReact
        highcharts={Highcharts}
```

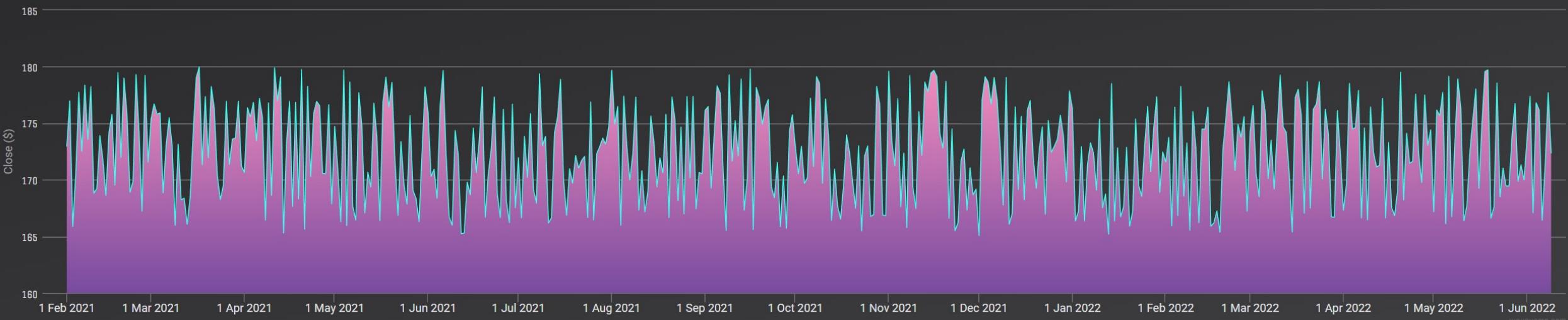


## StackedProductBarChart.tsx

```
1 import React from 'react';
2 import Highcharts from 'highcharts';
3 import HighchartsReact from 'highcharts-react-official';
4
5 export default class StackedProductBarChart extends React.Component {
6     /*Other methods and properties omitted*/
7
8     render() {
9         const chartOptions: Highcharts.Options = {
10             xAxis: {
11                 categories: ['January', 'February', 'March']
12             },
13             plotOptions: {
14                 bar: {
15                     stacking: 'normal'
16                 }
17             },
18             series: [
19                 { name: 'Anvil', data: [1, 2, 3] },
20                 { name: 'Giant Rubber Band', data: [1, 2, 3] },
21                 { name: 'Super Giant Rubber Band', data: [1, 2, 3] }
22             ]
23         };
24
25         return (
26             <HighchartsReact
27                 highcharts={Highcharts}
28                 options={chartOptions}
29             />
30         )
31     }
32 }
```

ACME @ NYSE

ACME CORPORATION



Highcharts



## TimeSeriesLineChart.tsx

```
1 import React from 'react';
2 import Highcharts from 'highcharts';
3 import HighchartsReact from 'highcharts-react-official';
4
5 export default class TimeSeriesLineChart extends React.Component {
6   /*Other methods and properties omitted*/
7
8   render() {
9     const chartOptions: Highcharts.Options = {
10       xAxis: {
11         type: 'datetime'
12       },
13       plotOptions: {
14         area: { /*Area chart specific settings*/ }
15       },
16       series: [
17         {
18           type: 'area',
19           name: 'Daily Close ($)',
20           data: [[1612137600000, 42]]
21         }
22       ];
23
24     return (
25       <HighchartsReact
26         highcharts={Highcharts}
27         options={chartOptions}
28       />
29     )
30   }
31 }
```

```
9 const chartOptions: Highcharts.Options = {  
10     xAxis: {  
11         type: 'datetime'  
12     },  
13     plotOptions: {  
14         area: { /*Area chart specific settings*/ }  
15     },  
16     series: [{  
17         type: 'area',  
18         name: 'Daily Close ($)',  
19         data: [[1612137600000, 42]]  
20     }]  
21 };  
22  
23 //
```

```
15      },
16      series: [{  
17          type: 'area',  
18          name: 'Daily Close ($)',  
19          data: [[161213760000, 42]]  
20      }]  
21  };  
22  
23  return (  
24      <HighchartsReact  
25          highcharts={Highcharts}  
26          options={chartOptions}  
27          />  
28  )
```



## TimeSeriesLineChart.tsx

```
1 import React from 'react';
2 import Highcharts from 'highcharts';
3 import HighchartsReact from 'highcharts-react-official';
4
5 export default class TimeSeriesLineChart extends React.Component {
6   /*Other methods and properties omitted*/
7
8   render() {
9     const chartOptions: Highcharts.Options = {
10       xAxis: {
11         type: 'datetime'
12       },
13       plotOptions: {
14         area: { /*Area chart specific settings*/ }
15       },
16       series: [{{
17         type: 'area',
18         name: 'Daily Close ($)',
19         data: [[1612137600000, 42]]
20       }]
21     };
22
23     return (
24       <HighchartsReact
25         highcharts={Highcharts}
26         options={chartOptions}
27         />
28     )
29   }
30 }
```

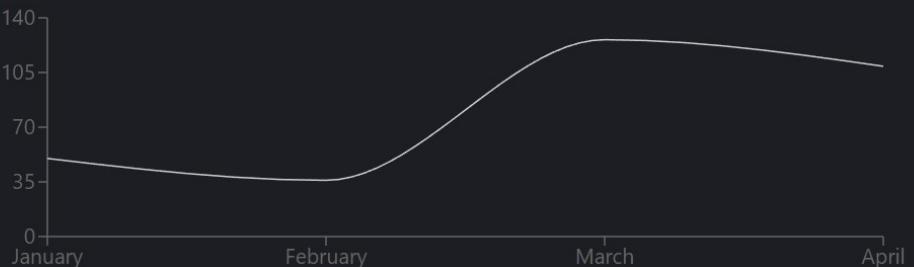
# Hiding Data



## W. E. Coyote

Joined January 2022

Number of Products Purchased per Month



£14,569,807.98

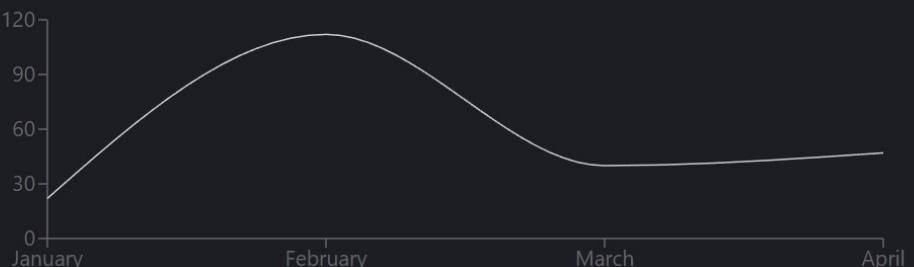
purchased

400 items purchased to date

## R. Runner

Joined January 2022

Number of Products Purchased per Month



£542.5 purchased

10 items purchased to date

MUI Card &  
Recharts



## SummaryCard.tsx

```
1 import React from 'react';
2 import { EuiCard, EuiText } from '@elastic/eui';
3
4 export default class SummaryCard extends React.Component {
5   /*Other methods and properties omitted*/
6
7   render() {
8     return (
9       <EuiCard
10         textAlign="left"
11         title={this.props.customer.name}
12         description="Joined January 2022">
13           <div>
14             <SummaryCardChart />
15             <EuiText size="relative">
16               <ul>
17                 <li>
18                   <label>
19                     {formatBalance(this.props.customer.net_purchases_gbp)}
20                   </label>
21                   <span className="descriptor">purchased</span>
22                 </li>
23                 <li>
24                   <label>
25                     {formatCounts(this.props.customer.item_purchase_count)}
26                   </label>
27                   <span className="descriptor">items purchased to date</span>
28                 </li>
29               </ul>
30             </EuiText>
31           </div>
32         </EuiCard>
33       )
34     }
35 }
```

```
6
7   render( ) {
8     return (
9       <EuiCard
10      textAlign="left"
11      title={this.props.customer.name}
12      description="Joined January 2022">
13      <div>
14        <SummaryCardChart />
15        <EuiText size="relative">
16          <ul>
17            <li>
18              <label>
19                {formatBalance(this.props.customer.balance)}
```

```
<div>
  <SummaryCardChart />
  <EuiText size="relative">
    <ul>
      <li>
        <label>
          {formatBalance(this.props.customer.net_
        </label>
        <span className="descriptor">purchased</s
      </li>
      <li>
        <label>
          {formatCounts(this.props.customer.item_
        </label>
```



## SummaryCard.tsx

```
1 import React from 'react';
2 import { EuiCard, EuiText } from '@elastic/eui';
3
4 export default class SummaryCard extends React.Component {
5   /*Other methods and properties omitted*/
6
7   render() {
8     return (
9       <EuiCard
10         textAlign="left"
11         title={this.props.customer.name}
12         description="Joined January 2022">
13           <div>
14             <SummaryCardChart />
15             <EuiText size="relative">
16               <ul>
17                 <li>
18                   <label>
19                     {formatBalance(this.props.customer.net_purchases_gbp)}
20                   </label>
21                   <span className="descriptor">purchased</span>
22                 </li>
23                 <li>
24                   <label>
25                     {formatCounts(this.props.customer.item_purchase_count)}
26                   </label>
27                   <span className="descriptor">items purchased to date</span>
28                 </li>
29               </ul>
30             </EuiText>
31           </div>
32         </EuiCard>
33       )
34     }
35 }
```

# Relationships

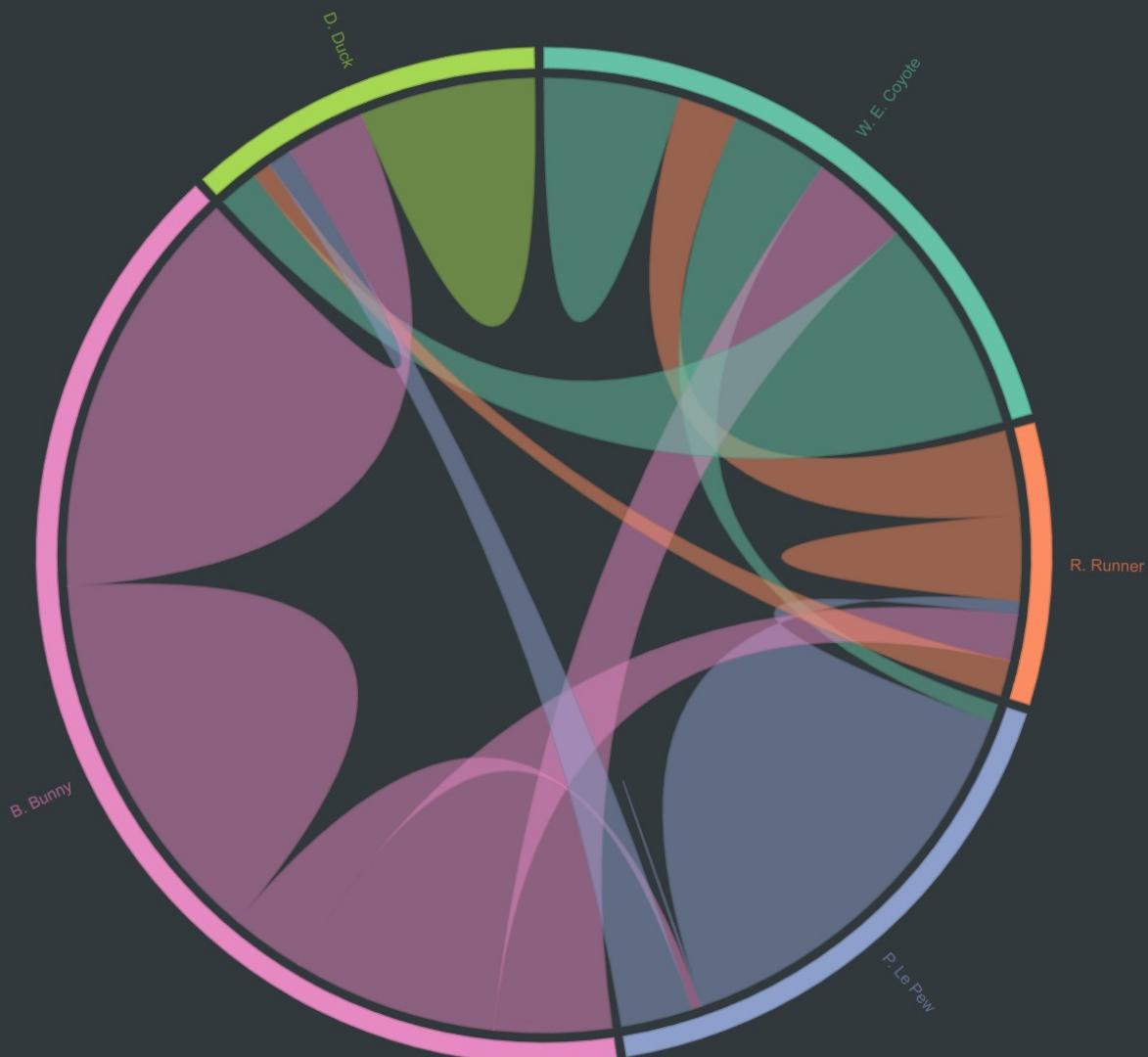


Products

Gift Card

Recipient	Placement Date ↓	Order ID	Product	Status	Purchase Price (£)
	dd/mm/yyyy				
▼ W. E. Coyote (7)					
D. Duck	Tue, 24 May 2022	1857638878	Gift Card	Shipped	84.97
B. Bunny	Thu, 24 June 2021	637359195	Gift Card	Shipped	71
P. Le Pew	Sun, 20 June 2021	1676862910	Gift Card	Shipped	15.22
R. Runner	Mon, 19 April 2021	958906527	Gift Card	Shipped	21.24
B. Bunny	Sat, 30 November 2019	861246392	Gift Card	Delivered	143
D. Duck	Tue, 2 July 2019	1443575445	Gift Card	Shipped	39.57
R. Runner	Fri, 28 June 2019	1863323482	Gift Card	Shipped	98.16
► D. Duck (9)					
▼ P. Le Pew (3)					
D. Duck	Sat, 31 October 2020	156957468	Gift Card	Shipped	53.1
M. Martian	Tue, 8 September 2020	1510899820	Gift Card	Delivered	93.37
D. Duck	Thu, 4 June 2020	1858536998	Gift Card	Shipped	96.42
▼ B. Bunny (5)					
P. Le Pew	Tue, 1 March 2022	585687541	Gift Card	Delivered	76.57
D. Duck	Sun, 9 January 2022	1015069743	Gift Card	Shipped	21.21
D. Duck	Wed, 2 June 2021	159580714	Gift Card	Delivered	100.66
R. Runner	Thu, 26 November 2020	1772591007	Gift Card	Delivered	88.14
M. Martian	Tue, 8 September 2020	1415595699	Gift Card	Delivered	8.51
► M. Martian (6)					
► R. Runner (6)					

AG Grid





## GiftCardChordDiagram.tsx

```
1 import React from 'react';
2 import { ResponsiveChord, RibbonTooltipComponentProps } from '@nivo/chord'
3 import { TableTooltip, Chip } from '@nivo/tooltip'
4
5 export default class GiftCardChordDiagram extends React.Component {
6     /*Other methods and properties omitted*/
7
8     private ribbonTooltip = ({ ribbon }: RibbonTooltipComponentProps) => (
9         <TableTooltip
10             rows={[
11                 [
12                     <Chip key="chip" color={ribbon.source.color} />,
13                     <span key="id">{ribbon.source.id}</span>,
14                     '>',
15                     <span key="id">{ribbon.target.id}</span>,
16                     <strong key="id">{formatBalance(ribbon.source.value)}</strong>,
17                 ]
18             ]}
19         />
20     )
21
22     render() {
23         const customers: string[] = [ 'W. E. Coyote', 'R. Runner', 'P. Le Pew', 'B. Bunny', 'M. Martian' ];
24         const data: number[][] = [
25             [ 650, 285, 465, 470, 1030 ],
26             /*Other data entries omitted*/
27         ];
28
29         return (
30             <ResponsiveChord
31                 data={this.state.data}
32                 ribbonTooltip={this.ribbonTooltip}
33                 keys={customers}
34                 padAngle={0.02}
35                 innerRadiusRatio={0.96}
36                 innerRadiusOffset={0.02}
37                 inactiveArcOpacity={0.25}
38                 onArcClick={this.props.handleDataPointClick}
39                 onRibbonClick={this.props.handleDataPointClick}
40                 activeRibbonOpacity={0.75}
41                 inactiveRibbonOpacity={0.25}
42                 /*Other chord chart configuration goes here*/
43             />
44         );
45     }
46 }
```

```
        ]}
    />
)

render() {
  const customers: string[] = [ 'W. E. Coyote', 'R. Runner', 'P. Le Pew', 'B. Bunny', 'M. Martian' ];
  const data: number[][] = [
    [ 650, 285, 465, 470, 1030 ],
    /*Other data entries omitted*/
  ];

  return (
    <ResponsiveChord
      data={this.state.data}
      ribbonTooltip={this.ribbonTooltip}
      keys={customers}
      padAngle={0.02}
      innerRadiusRatio={0.96}
      innerRadiusOffset={0.02}
      inactiveArcOpacity={0.25}
      onArcClick={this.props.handleDataPointClick}
      onRibbonClick={this.props.handleDataPointClick}
      activeRibbonOpacity={0.75}
      inactiveRibbonOpacity={0.25}
      /*Other chord chart configuration goes here*/
    />
  );
}
```

```
private ribbonTooltip = ({ ribbon }: RibbonTooltipComponentProps) => (
  <TableTooltip
    rows={[
      [
        <Chip key="chip" color={ribbon.source.color} />,
        <span key="id">{ribbon.source.id}</span>,
        ' -> ',
        <span key="id">{ribbon.target.id}</span>,
        <strong key="id">{formatBalance(ribbon.source.value)}</strong>,
      ]
    ]}
  />
)
```

```
render() {
  const customers: string[] = [ 'W. E. Coyote', 'R. Runner', 'P. Le Pew', 'B. Bu
```

```
32     ribbonTooltip={this.ribbonTool...  
33     keys={customers}  
34     padAngle={0.02}  
35     innerRadiusRatio={0.96}  
36     innerRadiusOffset={0.02}  
37     inactiveArcOpacity={0.25}  
38     onArcClick={this.props.handleD...  
39     onRibbonClick={this.props.hand...  
40     activeRibbonOpacity={0.75}  
41     inactiveRibbonOpacity={0.25}  
42     /*Other chord chart configurat...
```



## GiftCardChordDiagram.tsx

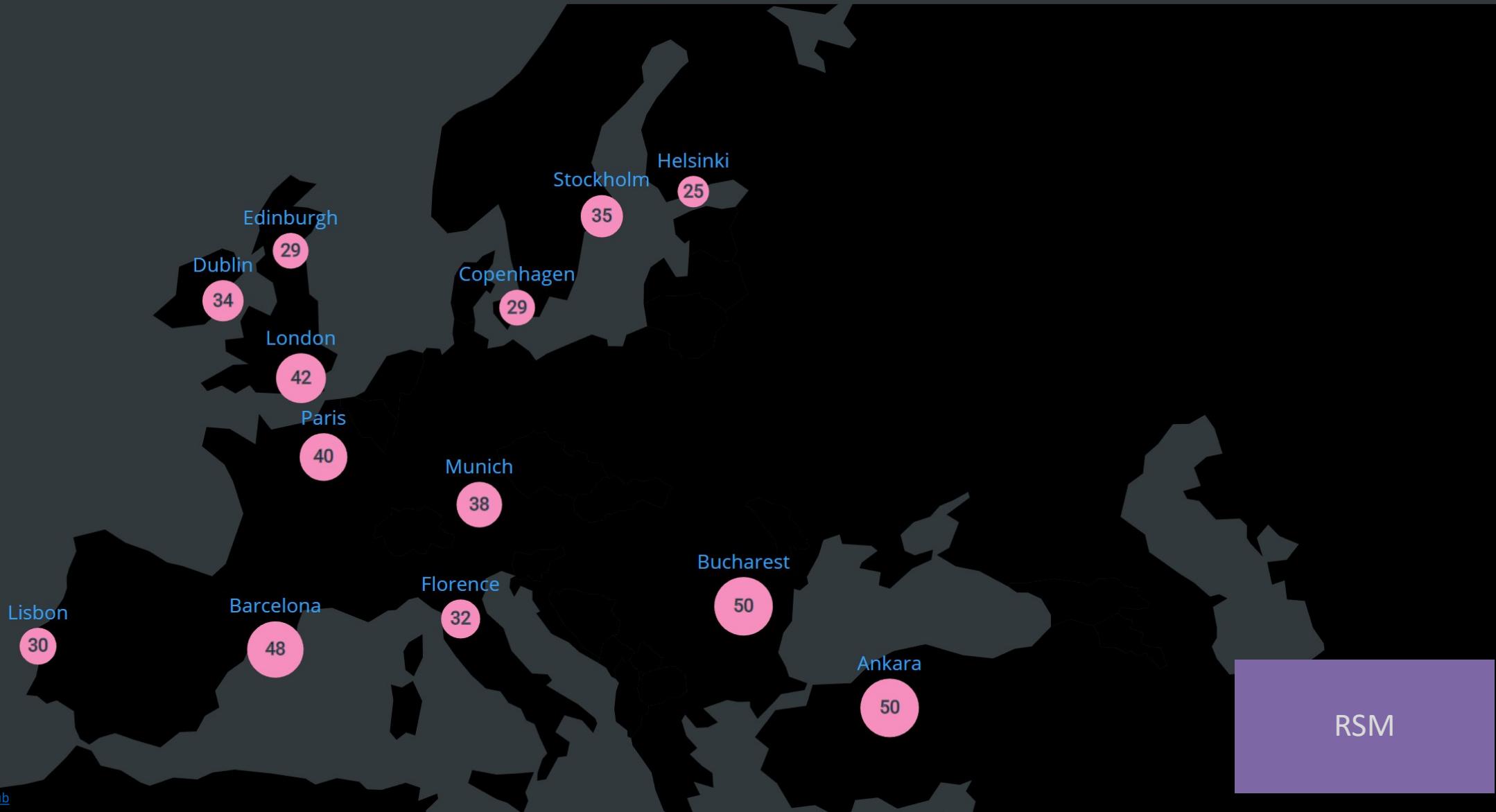
```
1 import React from 'react';
2 import { ResponsiveChord, RibbonTooltipComponentProps } from '@nivo/chord'
3 import { TableTooltip, Chip } from '@nivo/tooltip'
4
5 export default class GiftCardChordDiagram extends React.Component {
6     /*Other methods and properties omitted*/
7
8     private ribbonTooltip = ({ ribbon }: RibbonTooltipComponentProps) => (
9         <TableTooltip
10             rows={[
11                 [
12                     <Chip key="chip" color={ribbon.source.color} />,
13                     <span key="id">{ribbon.source.id}</span>,
14                     '>',
15                     <span key="id">{ribbon.target.id}</span>,
16                     <strong key="id">{formatBalance(ribbon.source.value)}</strong>,
17                 ]
18             ]}
19         />
20     )
21
22     render() {
23         const customers: string[] = [ 'W. E. Coyote', 'R. Runner', 'P. Le Pew', 'B. Bunny', 'M. Martian' ];
24         const data: number[][] = [
25             [ 650, 285, 465, 470, 1030 ],
26             /*Other data entries omitted*/
27         ];
28
29         return (
30             <ResponsiveChord
31                 data={this.state.data}
32                 ribbonTooltip={this.ribbonTooltip}
33                 keys={customers}
34                 padAngle={0.02}
35                 innerRadiusRatio={0.96}
36                 innerRadiusOffset={0.02}
37                 inactiveArcOpacity={0.25}
38                 onArcClick={this.props.handleDataPointClick}
39                 onRibbonClick={this.props.handleDataPointClick}
40                 activeRibbonOpacity={0.75}
41                 inactiveRibbonOpacity={0.25}
42                 /*Other chord chart configuration goes here*/
43             />
44         );
45     }
46 }
```

# Jurisdictions



## Order Delivery Volumes

Number of Orders Shipped per Month





## BubbleMap.tsx

```
1 import React from 'react';
2 import { ComposableMap, Geographies, Geography, Marker } from 'react-simple-maps';
3
4 export default class BubbleMap extends React.Component {
5
6   constructor(props) {
7     super(props);
8
9     this.state = {
10       geoUrl: 'https://raw.githubusercontent.com/zcreativelabs/react-simple-maps/master/topojson-maps/world-110m.json',
11       markerOffset: -5,
12       markers: [
13         { name: 'London', coordinates: [-0.118092, 51.509865], fulfilledPurchaseCount: 42 },
14         /*Other data entries omitted*/
15       ]
16     };
17   }
18
19   render() {
20     return (
21       <ComposableMap
22         projection="geoAzimuthalEqualArea"
23         projectionConfig={{
24           rotate: [-20.0, -52.0, 0],
25           scale: 700
26         }}>
27         <Geographies geography={this.state.geoUrl}>
28           {({ geographies }) =>
29             geographies.map(geo => <Geography key={geo.rsmKey} geography={geo} />)
30           }
31         </Geographies>
32         {this.state.markers.map(({ name, coordinates, fulfilledPurchaseCount }) => (
33           <Marker key={name} coordinates={coordinates}>
34             <circle r={fulfilledPurchaseCount/4} fill="#F68FBEB" stroke="#F68FBEB" strokeWidth={2}/>
35             <text y={0} textAnchor='middle' stroke="#5D5A6D" strokeWidth='0.5px' dy='.3em'
36               style={{ fontFamily: 'Roboto', fontSize: '0.6em' }}>{fulfilledPurchaseCount}</text>
37             <text
38               textAnchor='middle'
39               y={-(fulfilledPurchaseCount/4) + this.state.markerOffset}
40               style={{ fontFamily: 'Open Sans', fontSize: '0.65em', fill: '#36A2EF' }}>
41               {name}
42             </text>
43           </Marker>
44         )));
45       </ComposableMap>
46     );
47   }
48 }
```

```
18
19 render( ) {
20   return (
21     <ComposableMap
22       projection="geoAzimuthalEqualArea"
23       projectionConfig={{
24         rotate: [-20.0, -52.0, 0],
25         scale: 700
26     }}>
27     <Geographies geography={this.state.geoUrl}>
28       {({ geographies }) =>
29         geographies.map(geo => <Geography key={ge
30           }
31         </Geographies>
32         {this.state.markers.map(({ name, coordinates
33           })
34           )
35         )
36       }
37     }
38   )
39 }
```



```
1 import React from 'react';
2 import { ComposableMap, Geographies, Geography, Marker } from 'react-simple-maps';
3
4 export default class BubbleMap extends React.Component {
5
6   constructor(props) {
7     super(props);
8
9     this.state = {
10       geoUrl: 'https://raw.githubusercontent.com/zcreativelabs/react-simple-maps/master/topojson-maps/world-110m.json',
11       markerOffset: -5,
12       markers: [
13         { name: 'London', coordinates: [-0.118092, 51.509865], fulfilledPurchaseCount: 42 },
14         /*Other data entries omitted*/
15       ]
16     };
17   }
18
19   render() {
20     return (
21       <ComposableMap
22         projection="geoAzimuthalEqualArea"
23         projectionConfig={{
24           rotate: [-20.0, -52.0, 0],
25           scale: 700
26         }}>
27         <Geographies geography={this.state.geoUrl}>
28           {({ geographies }) =>
29             geographies.map(geo => <Geography key={geo.rsmKey} geography={geo} />)
30           }
31         </Geographies>
```

```
projectConfig={{
  rotate: [-20.0, -52.0, 0],
  scale: 700
}}>
<Geographies geography={this.state.geoUrl}>
  {({ geographies }) =>
    geographies.map(geo => <Geography key={geo.rsmKey} geography={geo} />)
  }
</Geographies>
{this.state.markers.map(({ name, coordinates, fulfilledPurchaseCount }) =>
<Marker key={name} coordinates={coordinates}>
  <circle r={fulfilledPurchaseCount/4} fill='#F68FBE' stroke='#F68FBE' str
  <text y={0} textAnchor='middle' stroke='#5D5A6D' strokeWidth='0.5px' dy=
  style={{ fontFamily: 'Roboto', fontSize: '0.6em' }}>{fulfilledPurchaseCo
<text
  textAnchor='middle'
  y={-(fulfilledPurchaseCount/4) + this.state.markerOffset}
  style={{ fontFamily: 'Open Sans', fontSize: '0.65em', fill: '#36A2EF' }}
  {name}
</text>
```

```
}}>
<Geographies geography={this.state.geoUrl}>
{({ geographies }) =>
  geographies.map(geo => <Geography key={geo.rsmKey} geography={geo} />)
}
</Geographies>
{this.state.markers.map(({ name, coordinates, fulfilledPurchaseCount }) => (
<Marker key={name} coordinates={coordinates}>
  <circle r={fulfilledPurchaseCount/4} fill="#F68FBE" stroke="#F68FBE" strokeWidth={2}/>
  <text y={0} textAnchor='middle' stroke="#5D5A6D" strokeWidth='0.5px' dy='.3em'
  style={{ fontFamily: 'Roboto', fontSize: '0.6em' }}>{fulfilledPurchaseCount}</text>
  <text
    textAnchor='middle'
    y={-(fulfilledPurchaseCount/4) + this.state.markerOffset}
    style={{ fontFamily: 'Open Sans', fontSize: '0.65em', fill: '#36A2EF' }}>
  {name}
  </text>
</Marker>
))}>
</ComposableMap>
```



## BubbleMap.tsx

```
1 import React from 'react';
2 import { ComposableMap, Geographies, Geography, Marker } from 'react-simple-maps';
3
4 export default class BubbleMap extends React.Component {
5
6   constructor(props) {
7     super(props);
8
9     this.state = {
10       geoUrl: 'https://raw.githubusercontent.com/zcreativelabs/react-simple-maps/master/topojson-maps/world-110m.json',
11       markerOffset: -5,
12       markers: [
13         { name: 'London', coordinates: [-0.118092, 51.509865], fulfilledPurchaseCount: 42 },
14         /*Other data entries omitted*/
15       ]
16     };
17   }
18
19   render() {
20     return (
21       <ComposableMap
22         projection="geoAzimuthalEqualArea"
23         projectionConfig={{
24           rotate: [-20.0, -52.0, 0],
25           scale: 700
26         }}>
27         <Geographies geography={this.state.geoUrl}>
28           {({ geographies }) =>
29             geographies.map(geo => <Geography key={geo.rsmKey} geography={geo} />)
30           }
31         </Geographies>
32         {this.state.markers.map(({ name, coordinates, fulfilledPurchaseCount }) => (
33           <Marker key={name} coordinates={coordinates}>
34             <circle r={fulfilledPurchaseCount/4} fill="#F68FBEB" stroke="#F68FBEB" strokeWidth={2}/>
35             <text y={0} textAnchor='middle' stroke="#5D5A6D" strokeWidth='0.5px' dy='.3em'
36               style={{ fontFamily: 'Roboto', fontSize: '0.6em' }}>{fulfilledPurchaseCount}</text>
37             <text
38               textAnchor='middle'
39               y={-(fulfilledPurchaseCount/4) + this.state.markerOffset}
40               style={{ fontFamily: 'Open Sans', fontSize: '0.65em', fill: '#36A2EF' }}>
41               {name}
42             </text>
43           </Marker>
44         )));
45       </ComposableMap>
46     );
47   }
48 }
```



A 4x4 grid of 16 eyeshadow pans in various colors: red, orange, pink, purple, blue, teal, green, and yellow. The pan in the top-left corner contains the text "Are Grids All Bad?" in white.

# Are Grids All Bad?

Sales Inventory Customer Statistics Purchase History Stock Price Gift Cards Master Gift Card Relationships Analytics Distribution Dashboard

Anvil >		Giant Rubber Band >		Gift Card >		Super Giant Rubber Band >	
Group	count(Purchase Price (£))	Customer Name	Placement Date				
M. Martian (28)	5	9	6	8	7	Order ID	Product
P. Le Pew (25)	6	6	6	7	5	Status	Purchase Price (£)
R. Runner (19)	8	4	3	4	4		
B. Bunny (28)	7	9	7	5	4		
D. Duck (28)	8	6	10	4	6		
W. E. Coyote (22)	5	7	4	6	4		

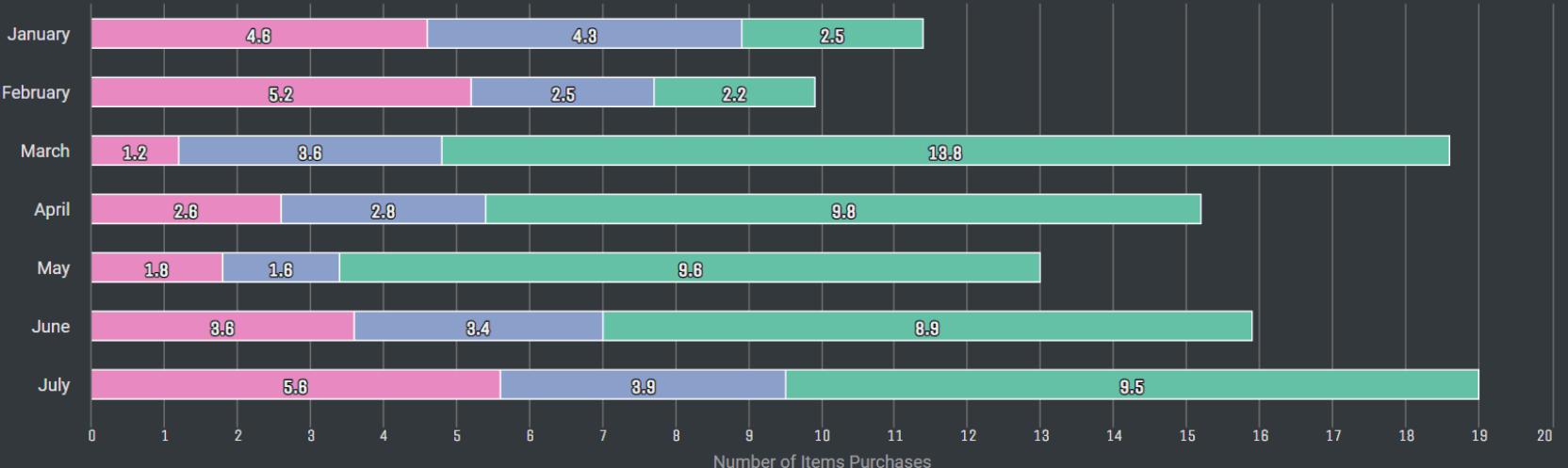
Pivot Mode Search...  
Customer Name Placement Date Order ID Product Status Purchase Price (£)  
Row Groups Customer Name  
Values count(Purchase Price (£))  
Column Labels Product Status

AG Grid

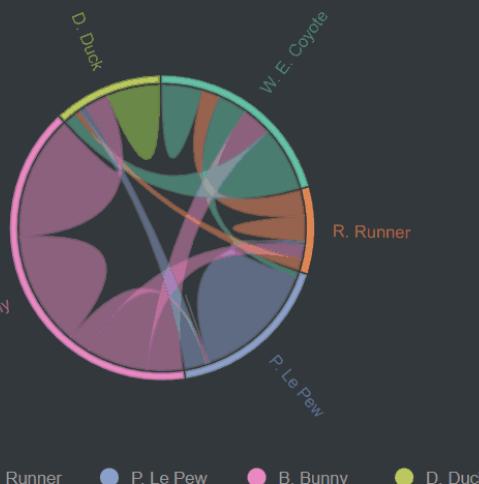
Example available via speaker's [GitHub](#)

## TOP CUSTOMER PRODUCT PURCHASES

W. E. COYOTE



## Gift Card Relationships



## Details

Customer Name	Placement Date	Order ID	Product	Status	Purchase Price (£)
(1) W. E. Coyote	dd/mm/yyyy				
W. E. Coyote	Thu, 5 May 2022	1091572680	>Anvil	Shipped	80.18
W. E. Coyote	Thu, 21 April 2022	347310879	Giant Rubber Band	Delivered	110.5
W. E. Coyote	Mon, 20 December 2021	1823666460	Anvil	Shipped	5.81
W. E. Coyote	Mon, 29 November 2021	862521418	Super Giant Rubber Band	Shipped	118.29
W. E. Coyote	Mon, 27 September 2021	316201296	Giant Rubber Band	Delivered	103.2
W. E. Coyote	Sun, 26 September 2021	777647146	Giant Rubber Band	Shipped	
W. E. Coyote	Sun, 4 July 2021	533927149	Giant Rubber Band	Shipped	

Highcharts, Nivo and  
AG Grid



## GiftCardChordDiagram.tsx

```
1 import React from 'react';
2 import { ResponsiveChord, RibbonTooltipComponentProps } from '@nivo/chord'
3 import { TableTooltip, Chip } from '@nivo/tooltip'
4
5 export default class GiftCardChordDiagram extends React.Component {
6     /*Other methods and properties omitted*/
7
8     private ribbonTooltip = ({ ribbon }: RibbonTooltipComponentProps) => (
9         <TableTooltip
10             rows={[
11                 [
12                     <Chip key="chip" color={ribbon.source.color} />,
13                     <span key="id">{ribbon.source.id}</span>,
14                     '>',
15                     <span key="id">{ribbon.target.id}</span>,
16                     <strong key="id">{formatBalance(ribbon.source.value)}</strong>,
17                 ]
18             ]}
19         />
20     )
21
22     render() {
23         const customers: string[] = [ 'W. E. Coyote', 'R. Runner', 'P. Le Pew', 'B. Bunny', 'M. Martian' ];
24         const data: number[][] = [
25             [ 650, 285, 465, 470, 1030 ],
26             /*Other data entries omitted*/
27         ];
28
29         return (
30             <ResponsiveChord
31                 data={this.state.data}
32                 ribbonTooltip={this.ribbonTooltip}
33                 keys={customers}
34                 padAngle={0.02}
35                 innerRadiusRatio={0.96}
36                 innerRadiusOffset={0.02}
37                 inactiveArcOpacity={0.25}
38                 onArcClick={this.props.handleDataPointClick}
39                 onRibbonClick={this.props.handleDataPointClick}
40                 activeRibbonOpacity={0.75}
41                 inactiveRibbonOpacity={0.25}
42                 /*Other chord chart configuration goes here*/
43             />
44         );
45     }
46 }
```

```
padAngle={0.02}
innerRadiusRatio={0.96}
innerRadiusOffset={0.02}
inactiveArcOpacity={0.25}
onArcClick={this.props.handleDataPointClick}
onRibbonClick={this.props.handleDataPointClick}
activeRibbonOpacity={0.75}
inactiveRibbonOpacity={0.25}
/*Other chord chart configuration goes here*/
```



## GiftCardChordDiagram.tsx

```
1 import React from 'react';
2 import { ResponsiveChord, RibbonTooltipComponentProps } from '@nivo/chord'
3 import { TableTooltip, Chip } from '@nivo/tooltip'
4
5 export default class GiftCardChordDiagram extends React.Component {
6     /*Other methods and properties omitted*/
7
8     private ribbonTooltip = ({ ribbon }: RibbonTooltipComponentProps) => (
9         <TableTooltip
10             rows={[
11                 [
12                     <Chip key="chip" color={ribbon.source.color} />,
13                     <span key="id">{ribbon.source.id}</span>,
14                     '>',
15                     <span key="id">{ribbon.target.id}</span>,
16                     <strong key="id">{formatBalance(ribbon.source.value)}</strong>,
17                 ]
18             ]}
19         />
20     )
21
22     render() {
23         const customers: string[] = [ 'W. E. Coyote', 'R. Runner', 'P. Le Pew', 'B. Bunny', 'M. Martian' ];
24         const data: number[][] = [
25             [ 650, 285, 465, 470, 1030 ],
26             /*Other data entries omitted*/
27         ];
28
29         return (
30             <ResponsiveChord
31                 data={this.state.data}
32                 ribbonTooltip={this.ribbonTooltip}
33                 keys={customers}
34                 padAngle={0.02}
35                 innerRadiusRatio={0.96}
36                 innerRadiusOffset={0.02}
37                 inactiveArcOpacity={0.25}
38                 onArcClick={this.props.handleDataPointClick}
39                 onRibbonClick={this.props.handleDataPointClick}
40                 activeRibbonOpacity={0.75}
41                 inactiveRibbonOpacity={0.25}
42                 /*Other chord chart configuration goes here*/
43             />
44         );
45     }
46 }
```



## ChartDrilldown.tsx

```
1 import React from 'react';
2
3 export default class ChartDrilldown extends React.Component<any, ChartDrilldownState> {
4
5   constructor(props: ChartDrilldownState | Readonly<ChartDrilldownState>) {
6     super(props);
7
8     this.state = {
9       chartSettings: { isDashboardChild: true, handleDataPointClick: this.updateFilterSelection.bind(this) },
10      dataGridSettings: { customerName: 'W. E. Coyote', productCategory: null }
11    }
12  }
13
14  private updateFilterSelection(event: any) {
15    const filterSelection: FilterSelection = { customerName: undefined, productCategory: 'Gift Card' }
16    // Highcharts
17    if (event.point) {
18      filterSelection.productCategory = event.point.series.name
19    }
20    // Nivo Chord Ribbon
21    else if (event.source) {
22      filterSelection.customerName = event.source.id;
23    }
24    // Nivo Chord Arc (must come after Ribbon as id used as combined key)
25    else if (event.id) {
26      filterSelection.customerName = event.id;
27    }
28
29    const newState: ChartDrilldownState = { chartSettings: this.state.chartSettings, dataGridSettings: filterSelection };
30    this.setState(newState);
31  }
32
33  render() {
34    return (
35      <div className="main-dashboard-panel">
36        <div data-testid="DashboardBarChart" className="bar-chart-panel">
37          <StackedProductBarChart {...this.state.chartSettings}/>
38        </div>
39
40        <div data-testid="DashboardChordDiagram" className="chord-diagram-container">
41          <h3 className="dashboard-component-heading">Gift Card Relationships</h3>
42          <GiftCardChordDiagram {...this.state.chartSettings}/>
43        </div>
44
45        <div data-testid="DashboardGrid" className="grid-panel" style={{width: '100%', height: '33vh'}}>
46          <h3 className="dashboard-component-heading">Details</h3>
47          <DataGrid {...this.state.dataGridSettings}/>
48        </div>
49      </div>
50    );
51  }
52 }
```

```
13
14 private updateFilterSelection(event: any) {
15   const filterSelection: FilterSelection = { customerName: undefined, productCategory: 'Gift Card' }
16   // Highcharts
17   if (event.point) {
18     filterSelection.productCategory = event.point.series.name
19   }
20   // Nivo Chord Ribbon
21   else if (event.source) {
22     filterSelection.customerName = event.source.id;
23   }
24   // Nivo Chord Arc (must come after Ribbon as id used as combined key)
25   else if (event.id) {
26     filterSelection.customerName = event.id;
27   }
28
29   const newState: ChartDrilldownState = { chartSettings: this.state.chartSettings, dataGridSettings: filterSelection };
30   this.setState(newState);
31 }
32
33 render() {
34   return (
35     <div className="main-dashboard-panel">
36       <div data-testid="DashboardBarChart" className="bar-chart-panel">
37         <StackedProductBarChart {...this.state.chartSettings}>
38       </div>
39
40       <div data-testid="DashboardChordDiagram" className="chord-diagram-container">
41         <h3 className="dashboard-component-heading">Gift Card Relationships</h3>
42         <GiftCardChordDiagram {...this.state.chartSettings}>
43       </div>
44     </div>
45   )
46 }
```



## ChartDrilldown.tsx

```
1 import React from 'react';
2
3 export default class ChartDrilldown extends React.Component<any, ChartDrilldownState> {
4
5   constructor(props: ChartDrilldownState | Readonly<ChartDrilldownState>) {
6     super(props);
7
8     this.state = {
9       chartSettings: { isDashboardChild: true, handleDataPointClick: this.updateFilterSelection.bind(this) },
10      dataGridSettings: { customerName: 'W. E. Coyote', productCategory: null }
11    }
12  }
13
14  private updateFilterSelection(event: any) {
15    const filterSelection: FilterSelection = { customerName: undefined, productCategory: 'Gift Card' }
16    // Highcharts
17    if (event.point) {
18      filterSelection.productCategory = event.point.series.name
19    }
20    // Nivo Chord Ribbon
21    else if (event.source) {
22      filterSelection.customerName = event.source.id;
23    }
24    // Nivo Chord Arc (must come after Ribbon as id used as combined key)
25    else if (event.id) {
26      filterSelection.customerName = event.id;
27    }
28
29    const newState: ChartDrilldownState = { chartSettings: this.state.chartSettings, dataGridSettings: filterSelection };
30    this.setState(newState);
31  }
32
33  render() {
34    return (
35      <div className="main-dashboard-panel">
36        <div data-testid="DashboardBarChart" className="bar-chart-panel">
37          <StackedProductBarChart {...this.state.chartSettings}/>
38        </div>
39
40        <div data-testid="DashboardChordDiagram" className="chord-diagram-container">
41          <h3 className="dashboard-component-heading">Gift Card Relationships</h3>
42          <GiftCardChordDiagram {...this.state.chartSettings}/>
43        </div>
44
45        <div data-testid="DashboardGrid" className="grid-panel" style={{width: '100%', height: '33vh'}}>
46          <h3 className="dashboard-component-heading">Details</h3>
47          <DataGrid {...this.state.dataGridSettings}/>
48        </div>
49      </div>
50    );
51  }
52 }
```



DataGrid.tsx

```
1 import React from 'react';
2 import { AgGridReact } from 'ag-grid-react';
3 import { ColDef, ColumnApi, GridApi } from 'ag-grid-community';
4 import 'ag-grid-enterprise';
5
6 export default class DataGrid extends React.Component {
7   /*Other methods and properties omitted*/
8
9   static getDerivedStateFromProps(props: any, state: { gridSettings: any; }) {
10     return {
11       gridSettings: state.gridSettings, filterSelection: props
12     };
13   }
14
15   componentDidUpdate(prevProps: FilterSelection) {
16     if (this.props.customerName == prevProps.customerName &&
17       this.props.productCategory == prevProps.productCategory) {
18       return;
19     }
20
21     this.applyFilterSelection();
22   }
23
24   private applyFilterSelection() {
25     const customerInstance = this.gridApi!.getFilterInstance('customerName');
26     const selectedCustomers: string[] | null = this.state.filterSelection?.customerName ?
27       this.state.filterSelection?.customerName?.split(',') : null;
28     customerInstance?.setModel({ values: selectedCustomers });
29
30     const productInstance = this.gridApi!.getFilterInstance('product');
31     const selectedProducts: string[] | null = this.state.filterSelection?.productCategory ?
32       [this.state.filterSelection?.productCategory] : null;
33     productInstance?.setModel({ values: selectedProducts });
34
35     this.gridApi!.onFilterChanged();
36   }
37
38   onGridReady(params: { api: GridApi | null; columnApi: ColumnApi | null; }) {
39     this.gridApi = params.api;
40     this.gridColumnApi = params.columnApi;
41
42     const customerData = generateCustomerPurchaseHistory(100);
43     this.gridApi?.setRowData(customerData);
44
45     this.applyFilterSelection();
46   }
47 }
```

```
23
24  private applyFilterSelection() {
25    const customerInstance = this.gridApi!.getFilterInstance('customerName');
26    const selectedCustomers: string[] | null = this.state.filterSelection?.customerName ?
27      this.state.filterSelection?.customerName?.split(',') : null;
28    customerInstance?.setModel({ values: selectedCustomers });
29
30    const productInstance = this.gridApi!.getFilterInstance('product');
31    const selectedProducts: string[] | null = this.state.filterSelection?.productCategory ?
32      [this.state.filterSelection?.productCategory] : null;
33    productInstance?.setModel({ values: selectedProducts });
34
35    this.gridApi!.onFilterChanged();
36  }
37
38  onGridReady(params: { api: GridApi | null; columnApi: ColumnApi | null; }) {
39    this.gridApi = params.api;
40    this.gridColumnApi = params.columnApi;
41
42    const customerData = generateCustomerPurchaseHistory(100);
43    this.gridApi?.setRowData(customerData);
44
45    this.applyFilterSelection();
46  }
47 }
```



## DataGrid.tsx

```
1 import React from 'react';
2 import { AgGridReact } from 'ag-grid-react';
3 import { ColDef, ColumnApi, GridApi } from 'ag-grid-community';
4 import 'ag-grid-enterprise';
5
6 export default class DataGrid extends React.Component {
7   /*Other methods and properties omitted*/
8
9   static getDerivedStateFromProps(props: any, state: { gridSettings: any; }) {
10     return {
11       gridSettings: state.gridSettings, filterSelection: props
12     };
13   }
14
15   componentDidUpdate(prevProps: FilterSelection) {
16     if (this.props.customerName == prevProps.customerName &&
17       this.props.productCategory == prevProps.productCategory) {
18       return;
19     }
20
21     this.applyFilterSelection();
22   }
23
24   private applyFilterSelection() {
25     const customerInstance = this.gridApi!.getFilterInstance('customerName');
26     const selectedCustomers: string[] | null = this.state.filterSelection?.customerName ?
27       this.state.filterSelection?.customerName?.split(',') : null;
28     customerInstance?.setModel({ values: selectedCustomers });
29
30     const productInstance = this.gridApi!.getFilterInstance('product');
31     const selectedProducts: string[] | null = this.state.filterSelection?.productCategory ?
32       [this.state.filterSelection?.productCategory] : null;
33     productInstance?.setModel({ values: selectedProducts });
34
35     this.gridApi!.onFilterChanged();
36   }
37
38   onGridReady(params: { api: GridApi | null; columnApi: ColumnApi | null; }) {
39     this.gridApi = params.api;
40     this.gridColumnApi = params.columnApi;
41
42     const customerData = generateCustomerPurchaseHistory(100);
43     this.gridApi?.setRowData(customerData);
44
45     this.applyFilterSelection();
46   }
47 }
```

**WE HEAR YOU.**

# Future Inspiration

- Data Sketches by Nadieh Bremer and Shirley Wu
- Information is Beautiful by David McCandless
- D3 Graph Gallery
- Newspaper Infographics
- Websites
- Code Samples



# Thank You!

# Q & A



Carly Richmond

@CarlyLRichmond

