

# Building Your Own Custom Data Visualization for Dashboards

Peter Peterson & Xianlin Hu  
Principal Software Engineers | Splunk

# Forward-Looking Statements



During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Turn Data Into Doing, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2019 Splunk Inc. All rights reserved.



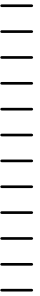
**Peter Peterson**

Principal Software Engineer | Splunk



**Xianlin Hu**

Principal Software Engineer | Splunk





# The State of the World

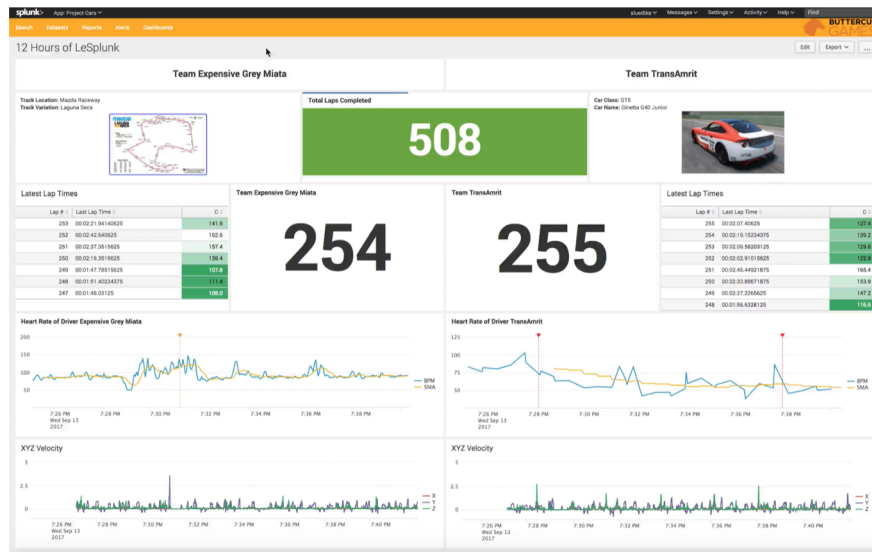
---

Where are Dashboards Today

# Existing Dashboard Tech Stack

## SimpleXML

- Default dashboard in Splunk Enterprise
- Supports JavaScript & CSS extensions



## Glasstable

- Bundled in Splunk premium apps such as ITSI
- Supports advanced editing experience



# Existing Dashboard Tech Stack

Limited to functionality provided by Splunk Web runtime

- Updates to Splunk Web runtime can cause unexpected issues
- Custom JS and CSS, and custom plugins lead to unexpected issues
- Splunk's front-end frameworks are out of date

Limited choices in how to tell a story with your data

- Glasstables are only available in Splunk premium apps
- Glasstable apps don't provide the same level of customization as SimpleXML
- SimpleXML dashboards show a lot of data, but not much of a story

Release cycle limited to Splunk Enterprise updates

# Changing the World

## A new Splunk Dashboard Framework

- Built from the ground up with modern front-end technologies (React/Redux)
- Semantically versioned packages regularly shipped to NPM
- Works equally well in Splunk Enterprise and Splunk Cloud Platform



# Tell Your Story

Combined features of SimpleXML and Glasstables

New framework is modular:

- All dependencies are resolved at build time
- New and improved visualizations
- Easily plug in your own visualizations





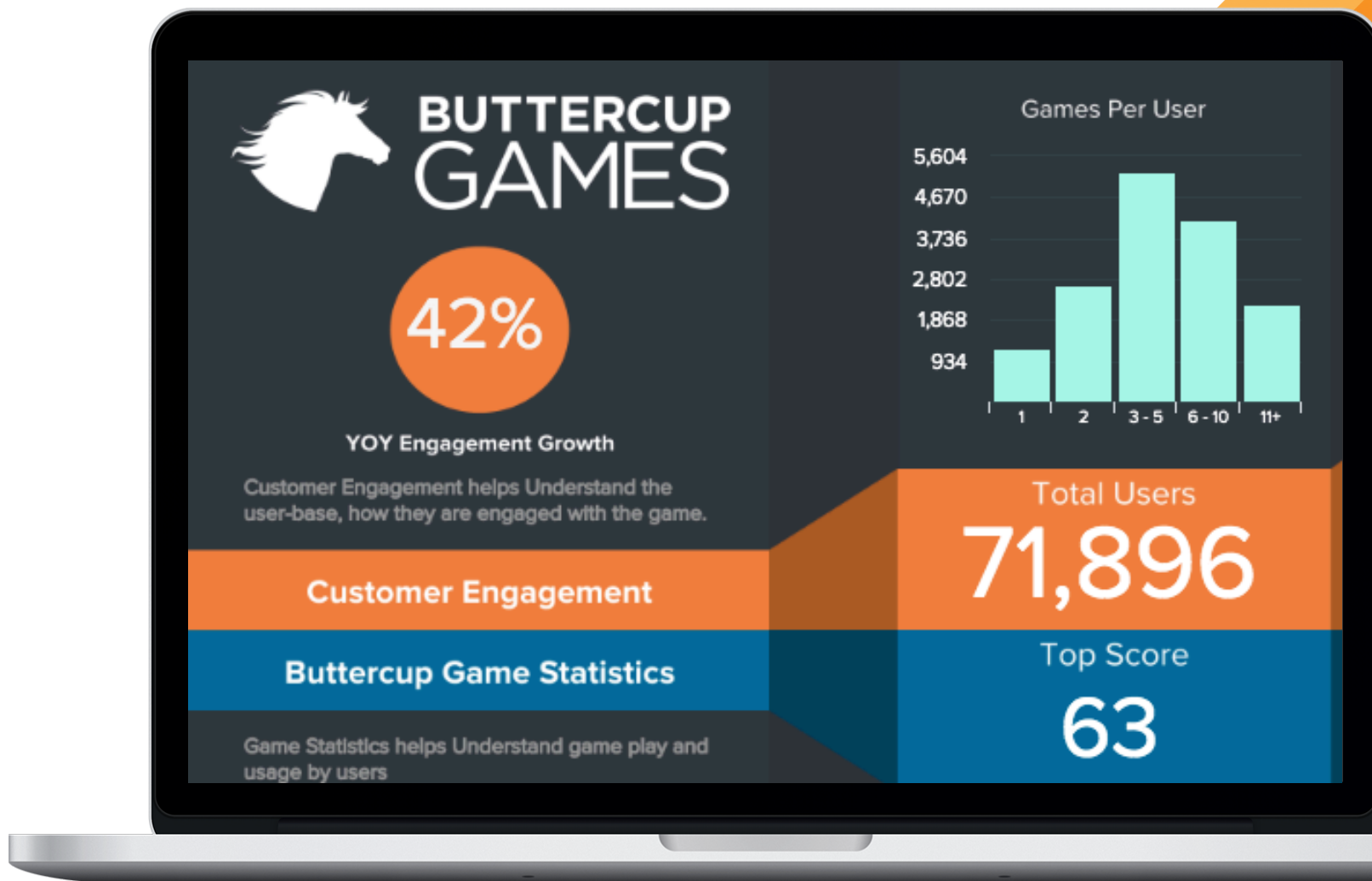


# Creating your own Custom Visualization!

---

# Definition

```
{
  "dataSources": {
    "search1": {
    },
    "visualizations": {
      "games_per_user": {
        "type": "viz.column",
        "options": {},
        "dataSources": {
          "primary": "search1"
        }
      }
    }
  },
  "layout": {
    "type": "absolute",
    "structure": [
      {
        "item": "games_per_user",
        "position": {
          "x": 40,
          "y": 170,
          "w": 610,
          "h": 380
        }
      }
    ]
  }
}
```



# Preset

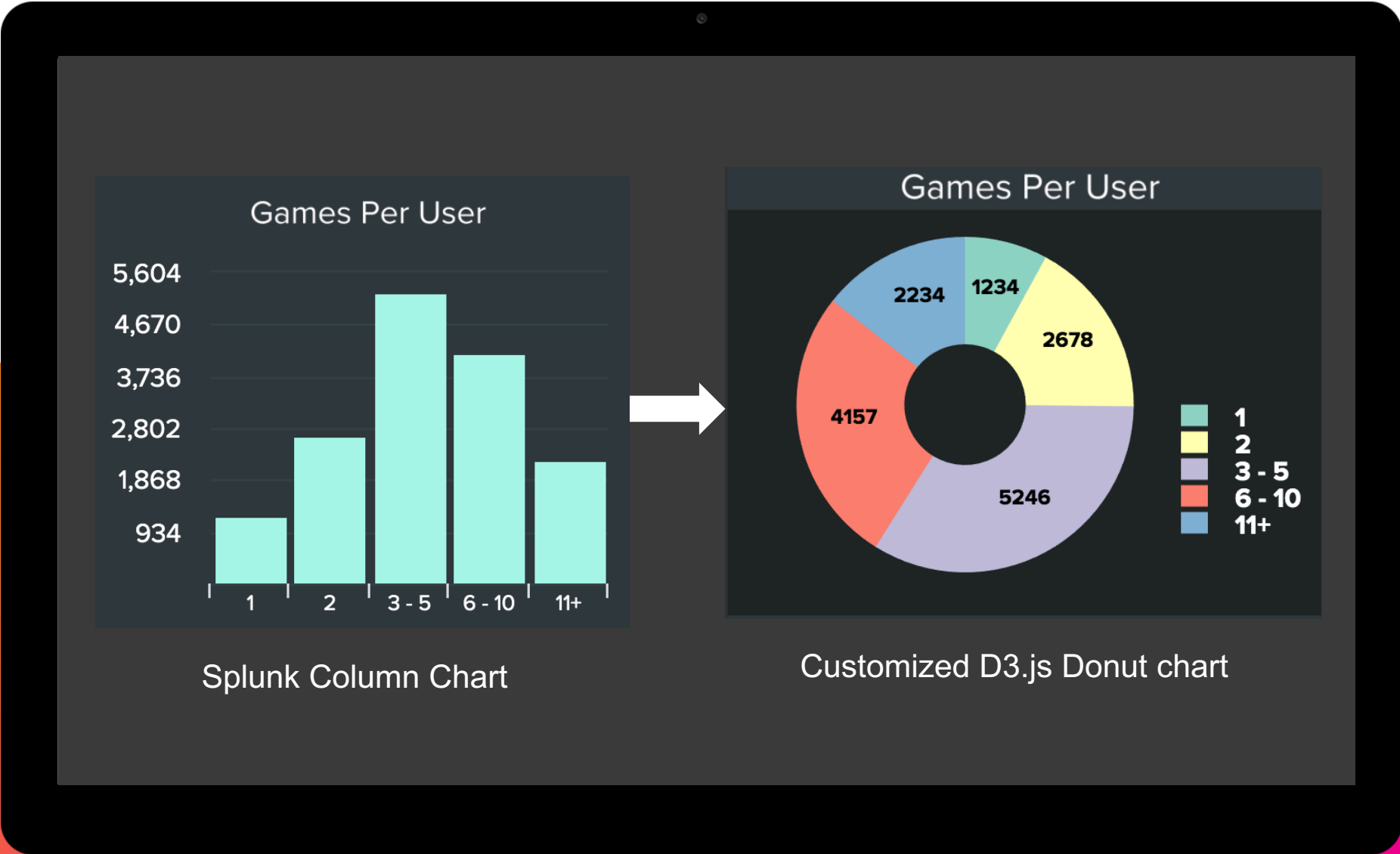
```
{
  "dataSources": {
    "search1": {...
  },
  "visualizations": {
    "games_per_user": {
      "type": "viz.column",
      "options": {},
      "dataSources": {
        "primary": "search1"
      }
    }
  },
  "layout": {
    "type": "absolute",
    "structure": [
      {
        "item": "games_per_user",
        "position": {
          "x": 40,
          "y": 170,
          "w": 610,
          "h": 380
        }
      }
    ]
  }
}
```

```
import Column from '@splunk/dashboard-visualizations/Column';
import AbsoluteLayout from '@splunk/dashboard-layouts/AbsoluteLayoutViewer';

const Preset = {
  visualizations: {
    'viz.column': Column,
  },
  layouts: {
    absolute: AbsoluteLayout
  }
};
```

Let's START!

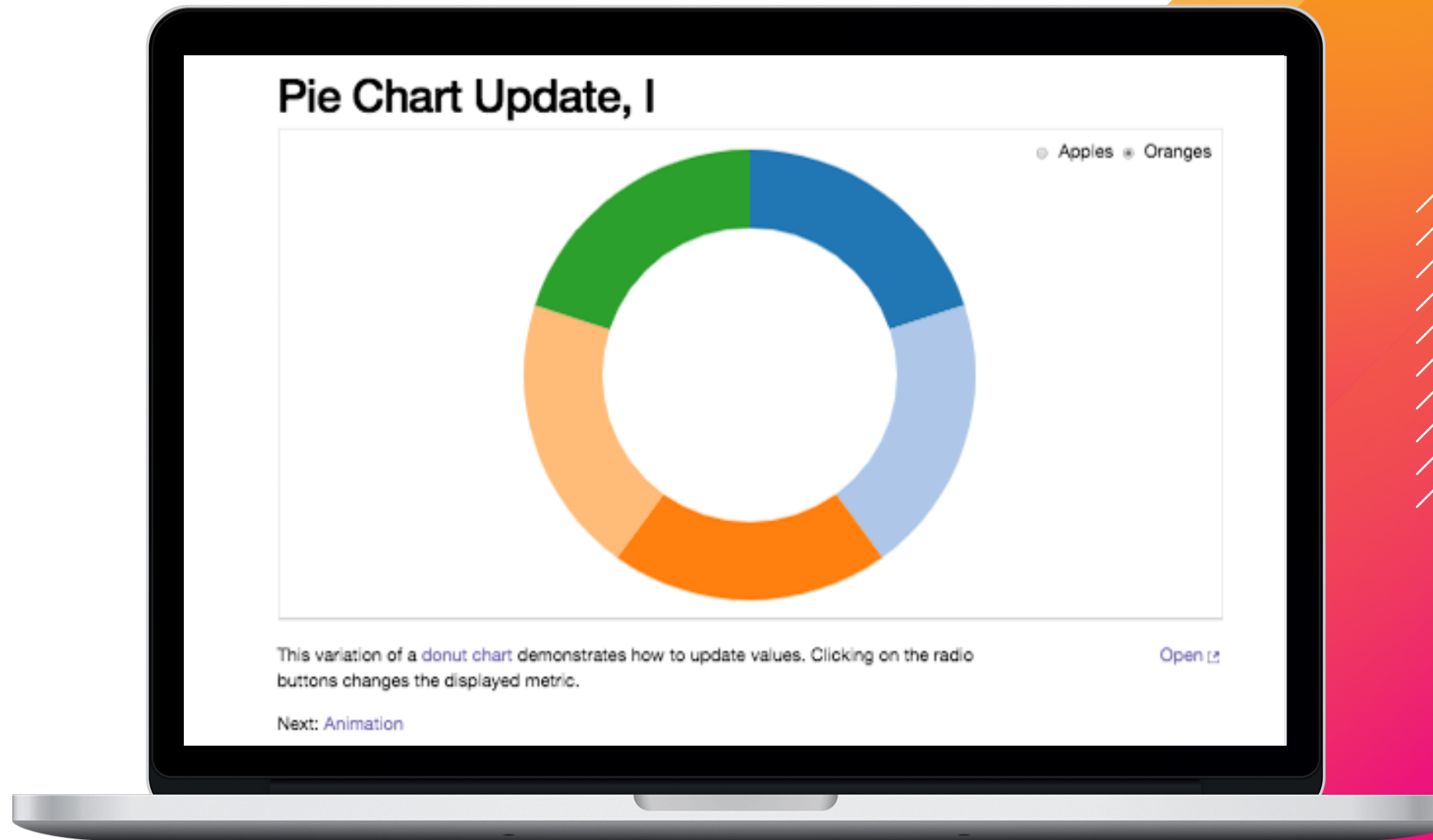




# 3D Donut Chart

Find a 3D donut chart example:

<https://bl.ocks.org/mboostock/1346395>



# React Component

<https://react-d3-library.github.io/>

## Install

- npm install --save react-d3-library

## Using existing d3 code

Have your React component!

```
import rd3 from 'react-d3-library';
import node from 'd3file';
const RD3Component = rd3.Component;

class my_First_React_D3_Library_Component extends React.Component {

  constructor(props) {
    super(props);
    this.state = {d3: ''}
  }

  componentDidMount() {
    this.setState({d3: node});
  }

  render() {
    return (
      <div>
        <RD3Component data={this.state.d3} />
      </div>
    )
  }
};
```



# Add to Dashboard

## Add to Preset

### Add to Preset

```
visualizations: {  
  ...DefaultPreset.visualizations,  
  'viz.column': Column,  
  'viz.donut': Donut,  
},  
layouts: {  
  ...DefaultPreset.layouts,  
  'absolute': AbsoluteLayout,  
}  
};
```

### Update Definition

```
"dataSources": {  
  "search1": {  
  },  
  "visualizations": {  
    "games_per_user": {  
      "type": "viz.donut",  
      "options": {},  
      "dataSources": {  
        "primary": "search1"  
      }  
    }  
  },  
  "layout": {  
    "type": "absolute",
```

# Demo

# How about another Visualization?

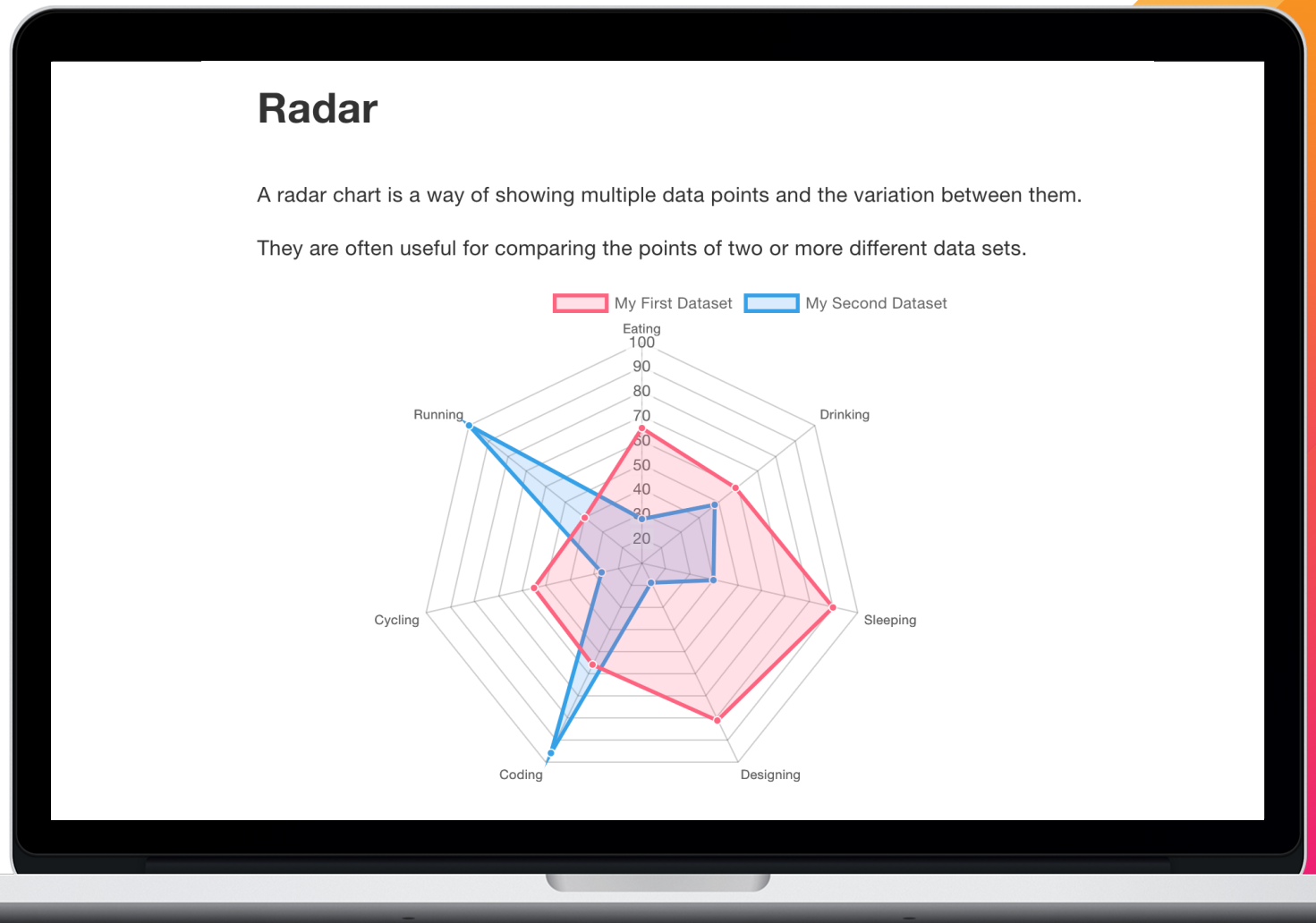
Let's do that again with Chart.js

Lots of interesting sample visualizations

<https://www.chartjs.org/>

There's already a nice React wrapper

<https://www.npmjs.com/package/react-chartjs-2>



# Chart.JS: React Component

<https://www.npmjs.com/package/react-chartjs-2>

## Install

- `npm install --save react-chartjs-2`
- `npm install --save chartjs@^2.5`

Use the Chart.JS Component!

```
1 import React from 'react';
2 import Radar from 'react-chartjs-2';
3
4 > const extractData = function (dataSources) {...
32 }
33
34 const RadarViz = ({ options, width, height, dataSources }) => {
35   const chartData = extractData(dataSources);
36
37   return (
38     <Radar
39       data={chartData}
40       width={width}
41       height={height}
42       options={{}}
43     />
44   );
45 };
```

# Chart.JS: Getting Data In

## Transforming DataSources

- Dashboard DataSources contain fields and associated column data
- Chart.JS needs a set of labels (fields) and an array of datasets

```
"data": {
  "fields": [
    { "name": "Eating" },
    { "name": "Drinking" },
    { "name": "Sleeping" },
    { "name": "Designing" },
    { "name": "Coding" },
    { "name": "Cycling" },
    { "name": "Running" }
  ],
  "columns": [
    [ 65, 28 ], // Eating
    [ 59, 48 ], // Drinking
    [ 90, 40 ], // Sleeping
    [ 81, 25 ], // Designing
    [ 56, 96 ], // Coding
    [ 55, 27 ], // Cycling
    [ 40, 100 ] // Running
  ]
}
```

```
data: {
  labels: ['Eating', 'Drinking', 'Sleeping', 'Designing'],
  datasets: [
    { data: [65, 59, 90, 81, 56, 55, 40] },
    { data: [28, 48, 40, 19, 96, 27, 100] },
  ]
}
```

# Add to Dashboard

- Add to Preset

```
const Preset = {  
  visualizations: {  
    ...DefaultPreset.visualizations,  
    'viz.column': Column,  
    'viz.radar': Radar,  
  },  
  layouts: {  
    ...DefaultPreset.layouts,  
    'absolute': AbsoluteLayout,  
  }  
};
```

- Update Definition

```
{  
  "dataSources": {  
    "search1": {  
    },  
  },  
  "visualizations": {  
    "score_distribution": {  
      "type": "viz.radar",  
      "options": {},  
      "dataSources": {  
        "primary": "search1"  
      }  
    }  
  },  
  "layout": {
```

# Demo



# Summary

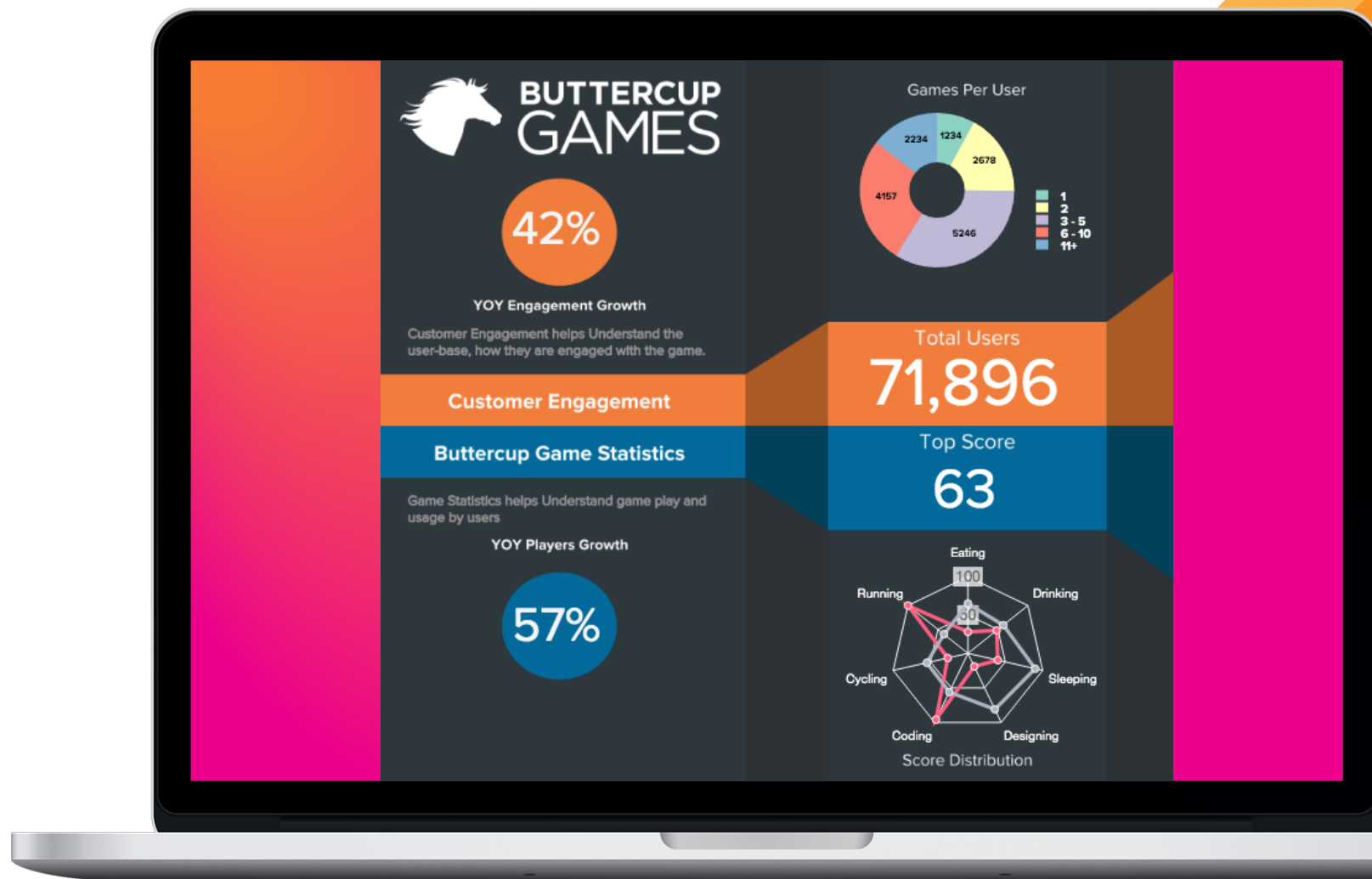
---

Section subtitle goes here



You can make  
your  
visualizations  
as easy or as  
complex as  
you want

Write once, use  
everywhere



# Resources

## Docs

- Enterprise: [https://docs.splunk.com/Documentation/DashApp/1.0.0beta/DashApp/dashEditor#Build\\_Dashboards](https://docs.splunk.com/Documentation/DashApp/1.0.0beta/DashApp/dashEditor#Build_Dashboards)
- Splunk Investigate: <https://docs.splunk.com/Documentation/SplunkInvestigate/Current/Use/DashOverview>
- Splunk Developer Cloud: <https://sdc.splunkbeta.com/docs/dashviz>

## NPM

- <to link once finalized>

## GitHub for conf content

- <https://github.com/splunk/dashboard-conf19-examples>

# Dashboard Sessions

**FN1815** - The New Dashboarding & Content Export Experience in Splunk: A single experience across Enterprise, SCP, ITSI and more!

**DEV 2165** – Deep Dive on The New Dashboarding & Content Export Experience

**DEV1141** – Building Applications with Splunk UI and Splunk React Visualizations

**FN1933** – Unleash your Inner Picasso – Splunk's New Dashboard Framework

**FN1735** – What's next for Geo in Splunk

**DEV2171** – Build your own custom data visualization on dashboard

# We want to hear from you!

Reaching the team

## 1. Dashboard Usage & Feedback Survey

- <to link once finalized>

## 2. Booths

- Foundations & Platform > Splunk Enterprise
- Developer (Dev Zone) > Dashboards
- Developer (Dev Zone) > Visualizations Library

## 3. Email

- Core Dashboard & Splunk Investigate - [dashboards@splunk.com](mailto:dashboards@splunk.com)
- Core Viz & Splunk Investigate - [visualizations@splunk.com](mailto:visualizations@splunk.com)
- Enterprise Experience – [dashboardsbeta@splunk.com](mailto:dashboardsbeta@splunk.com)



splunk>

# Thank

# You!

Go to the .conf19 mobile app to

**RATE THIS SESSION**

