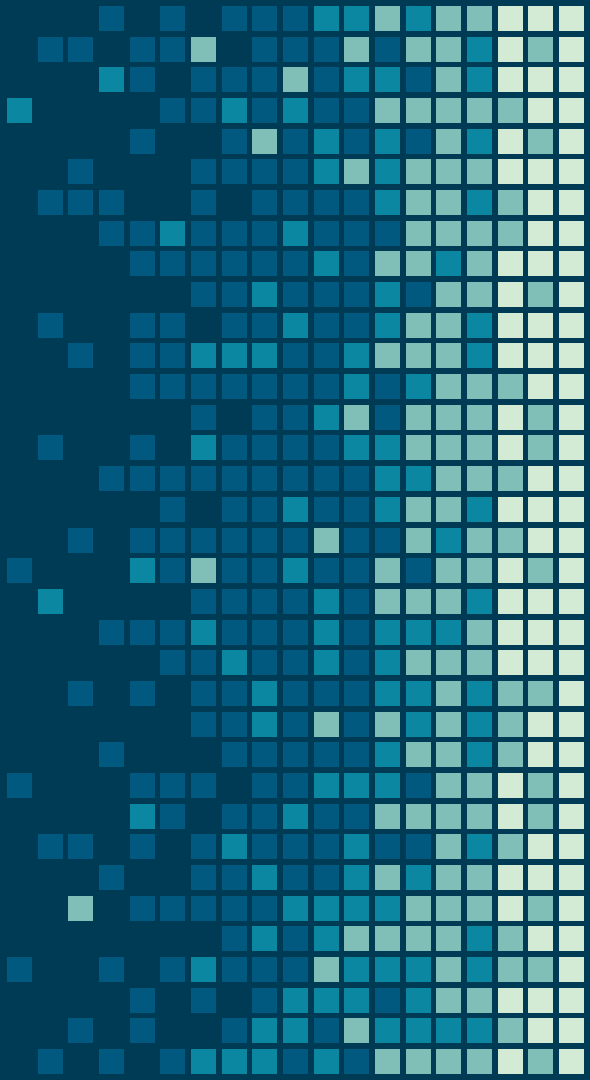
Statistics visualization for the CERN Document Server

Invenio-Stats-Js

Ioannis Androulidakis

School of Electrical & Computer Engineering,
National Technical University of Athens

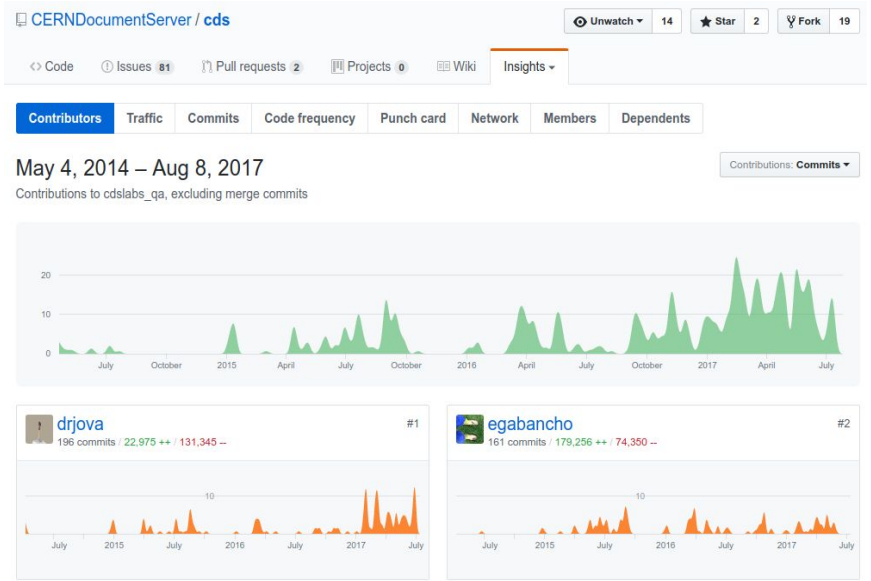
August 21, 2017, CERN

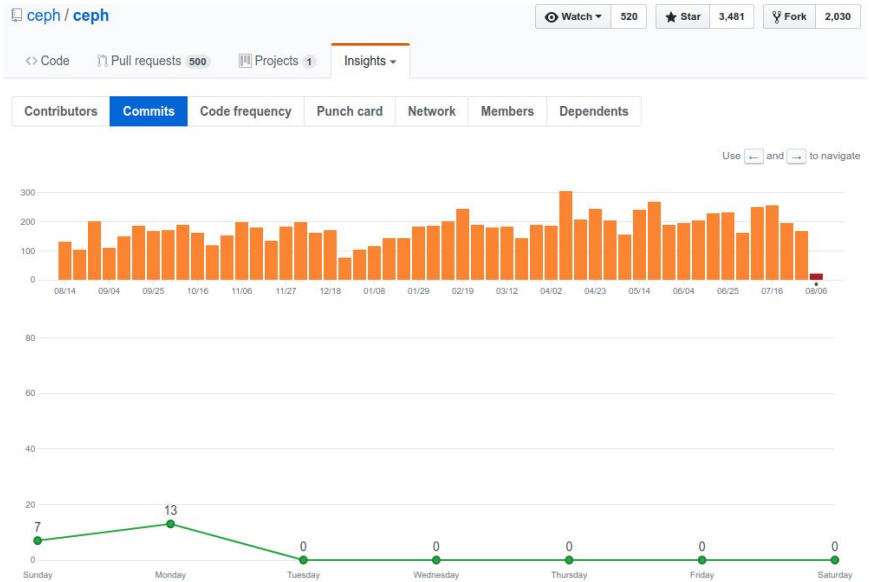


DATA FOR HUMANS

Why even visualize ?

Github insights





Google Maps popular hours

Youtube statistics

Storytelling

Visualization a simple, intuitive and powerful representation of data

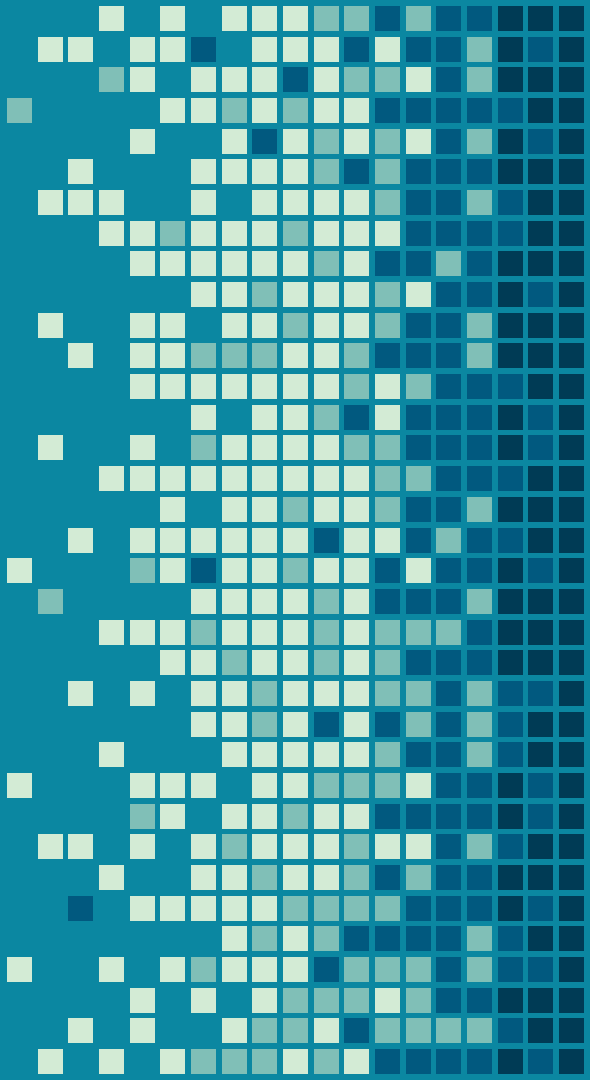
Extract all core information of a dataset

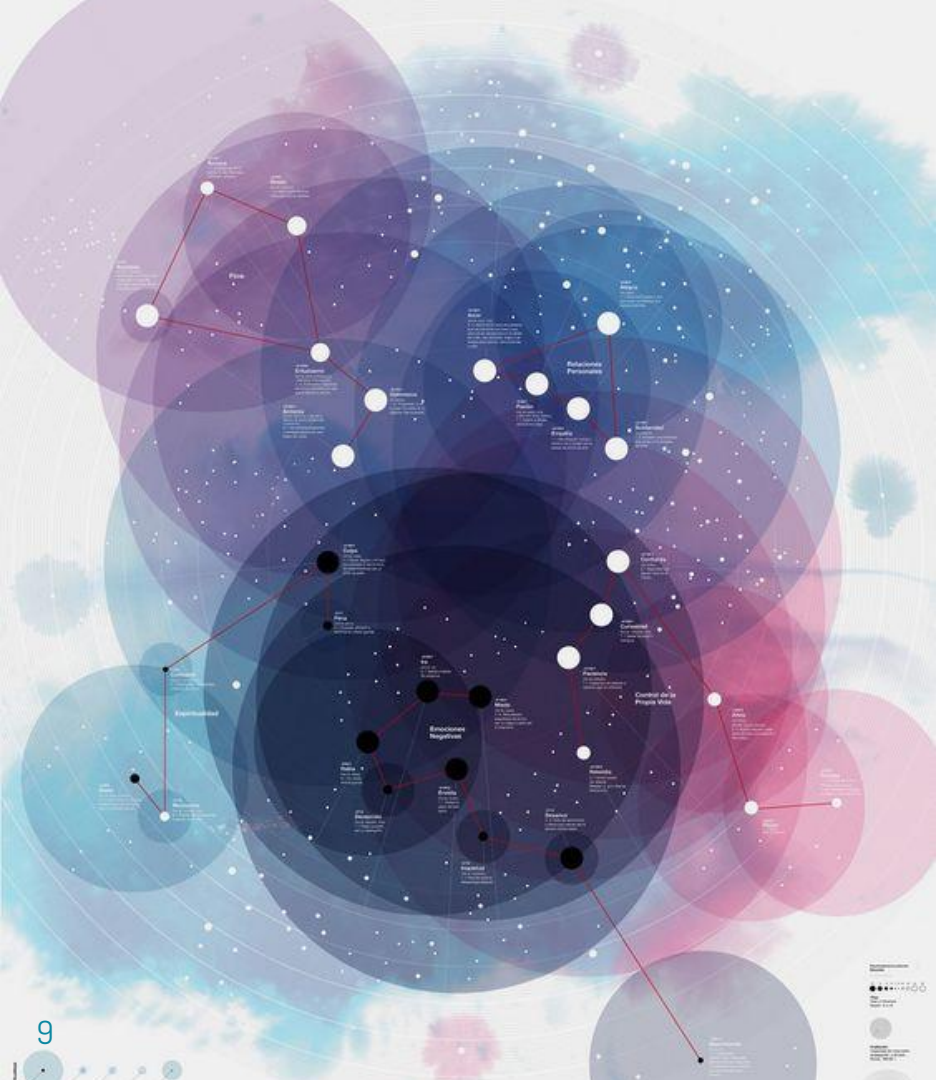
- Discover new insights
- Observation and understanding of behavior
- Trend analysis / pattern recognition
- Data reasoning and correlation
- Decision making and planning

Motivation

Kibana Dashboard

> *Design a flexible, standalone visualization module for Invenio, that delivers highly-configurable, interactive and responsive charts*





Best practices

- Strip away the noise
- Keep it clean and concise
- Identify needs and customize
- Convey a substantive message
- Keep it open-source 

CURRENT PICTURE

Invenio + CDS

CERN projects involved

Invenio 3.0*

A digital library framework that enables building and running large scale digital repositories

Collection of modules that provide functionality

http://inveniosoftware.org/

https://github.com/inveniosoftware/

CERN Document Server

Enables access to articles, reports and multimedia content in High Energy Physics

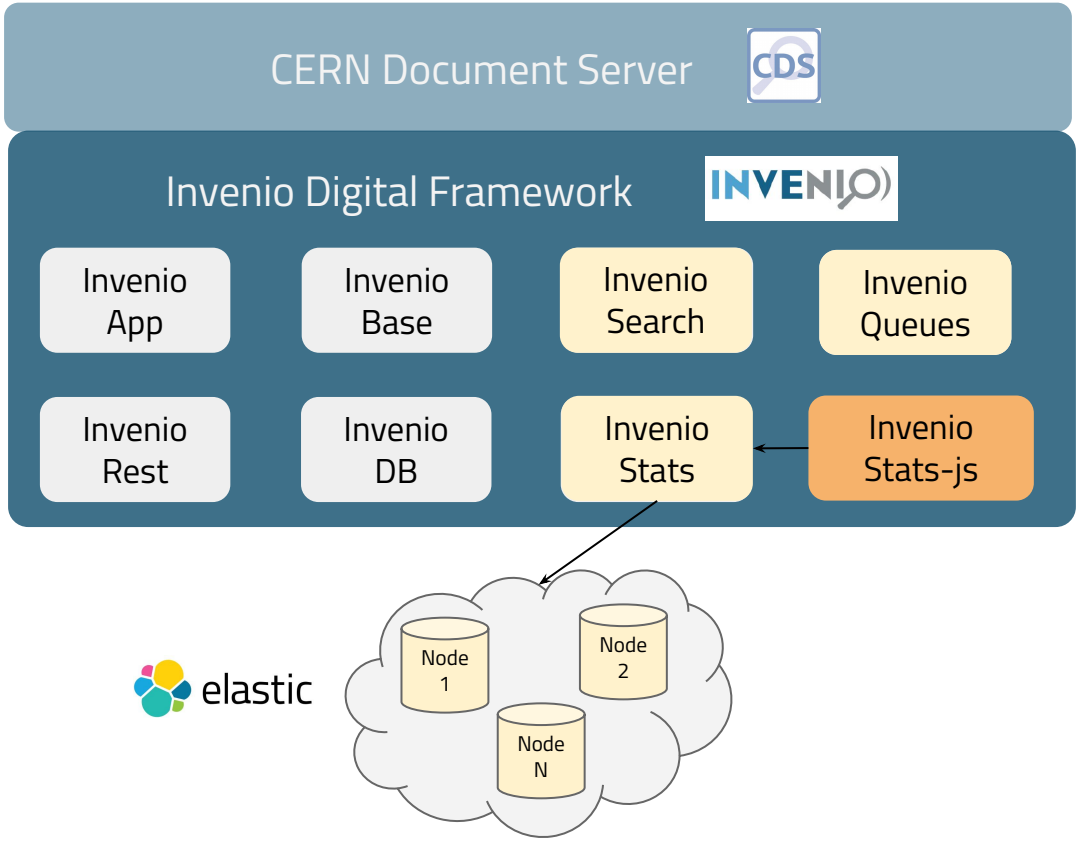
An Invenio instance

https://cds.cern.ch/

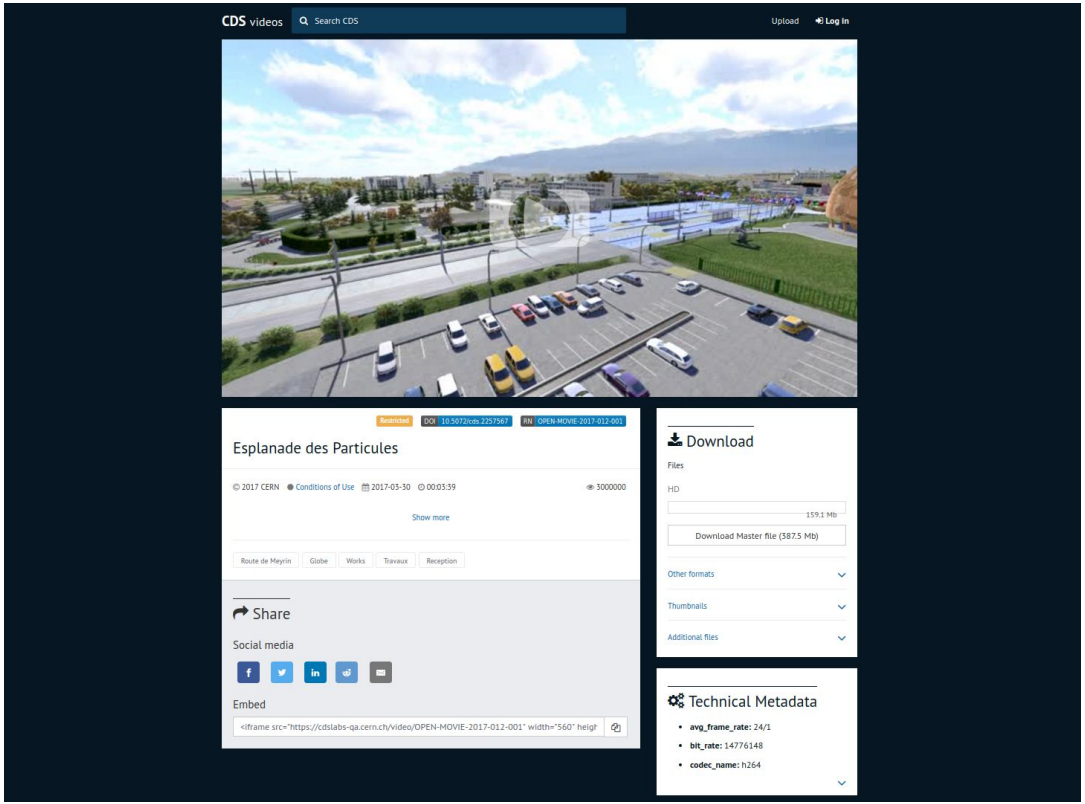
https://github.com/CERNDocumentServer/

*Currently under development

Architecture



Upcoming CDS videos



SOFTWARE STACK

Our toolset

Core Components

D3.js v4.10

Data-driven documents

De facto visualization framework

Powerful **functional-style programming**

Selections, bindings, maps, filters, transitions, interactions

Low level concepts in **SVG**, **HTML** & **DOM**

NodeJS v4.8

Runtime environment based on the V8 Javascript engine

Event-driven architecture

Automation of **package installation** and code compilation

Ecmascript v6

The latest version of **Javascript**, designed to support async/await, arrow functions, **classes**, modules, template literals

Webpack v3.5

A module **bundler** for modern Javascript applications

Build assets, scripts, styles in a **dependency graph**

Code

```
npm install invenio-stats-js
```

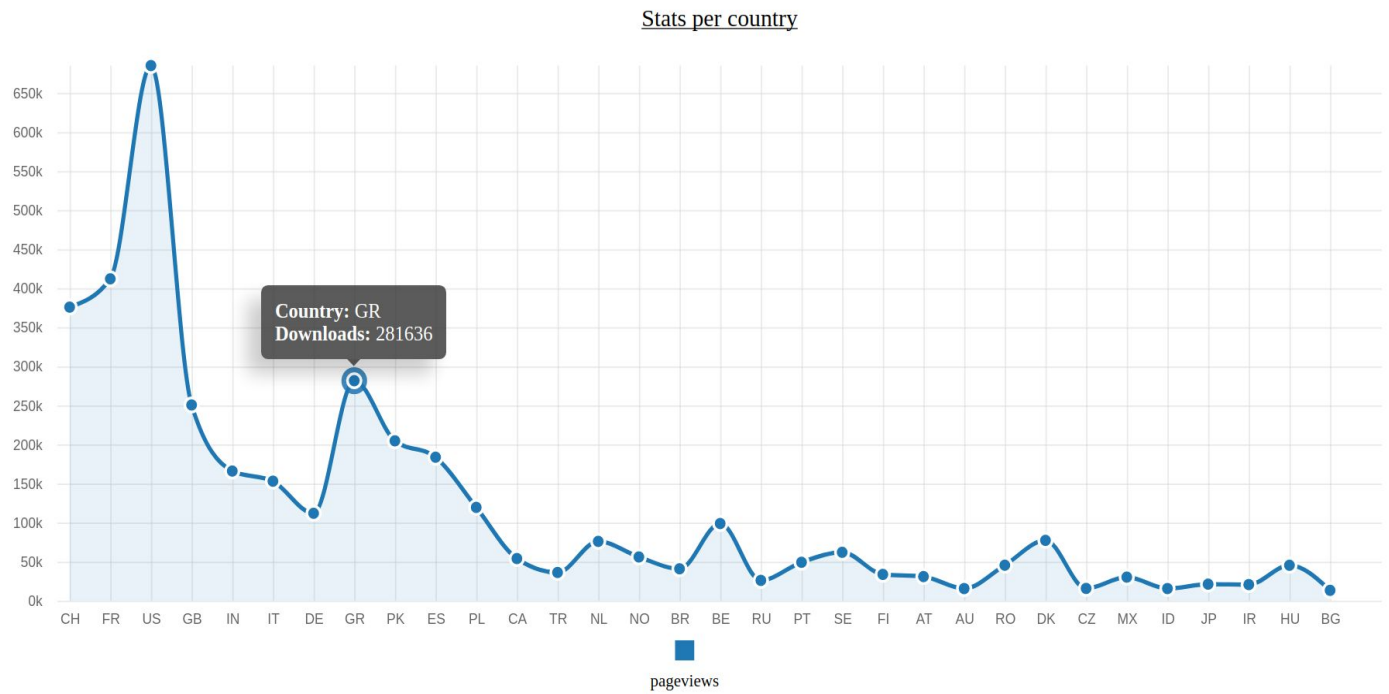
(soon available on )

```
// Import as ES2015 module
import { LineGraph, BarGraph } from 'invenio-stats-js';

// Create a line chart
new LineGraph(data, 'pageviews').render();

// Create a bar chart
new BarGraph(data, 'downloads').render();
```

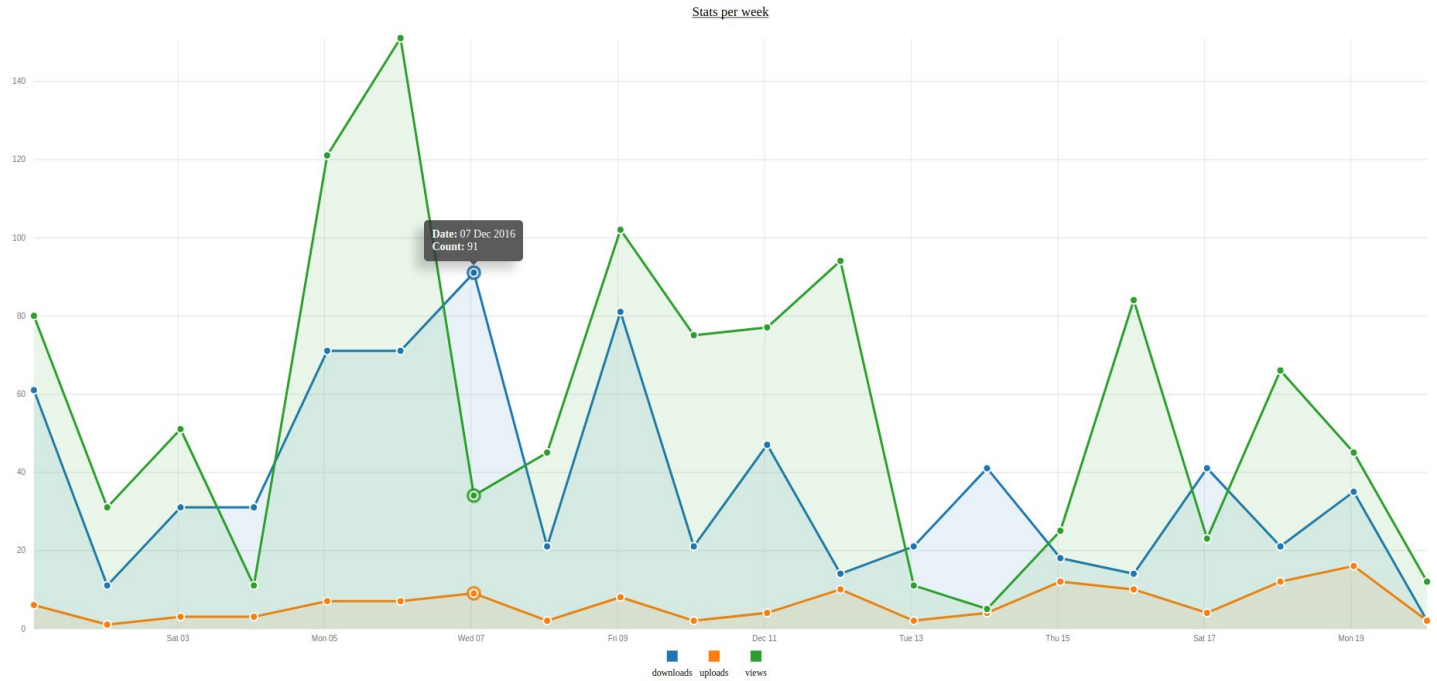
EARLY RESULTS

Alpha version of charts

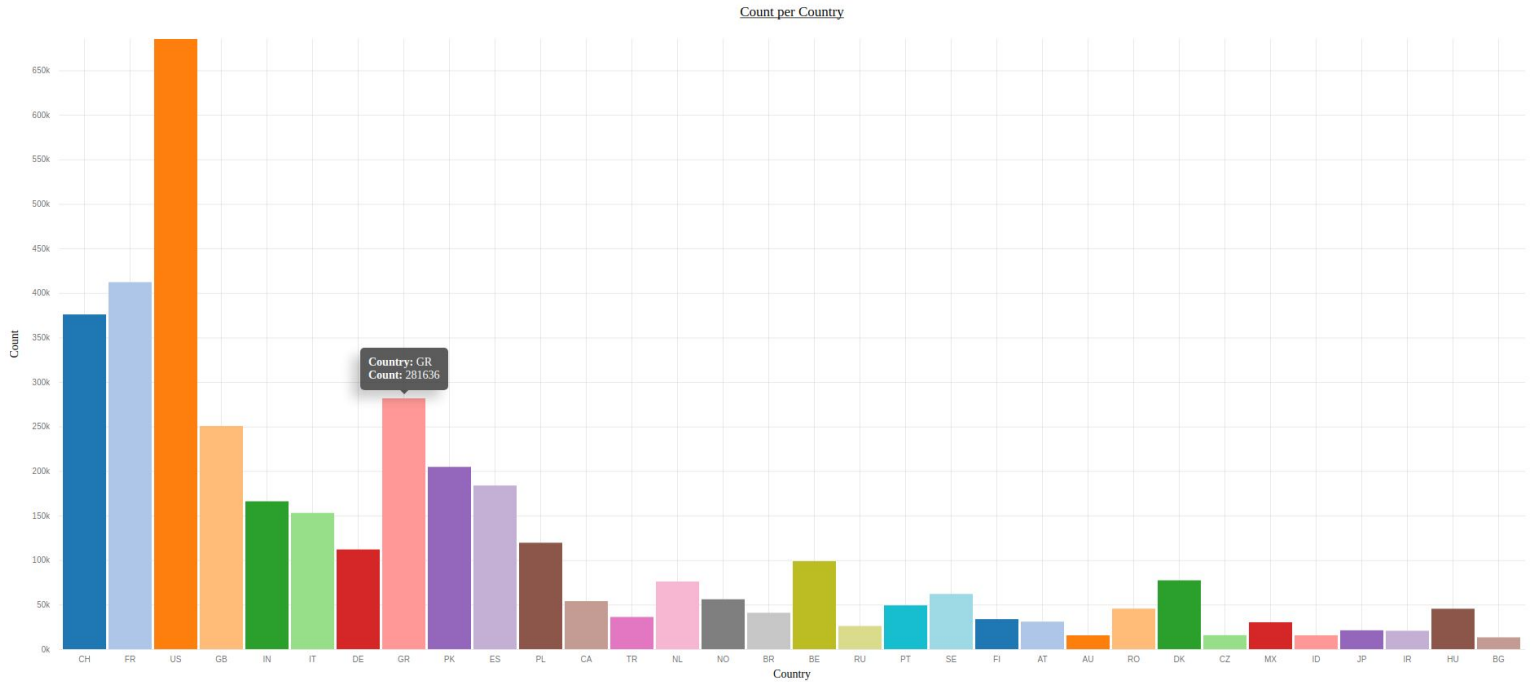
Simple Line/Area Chart



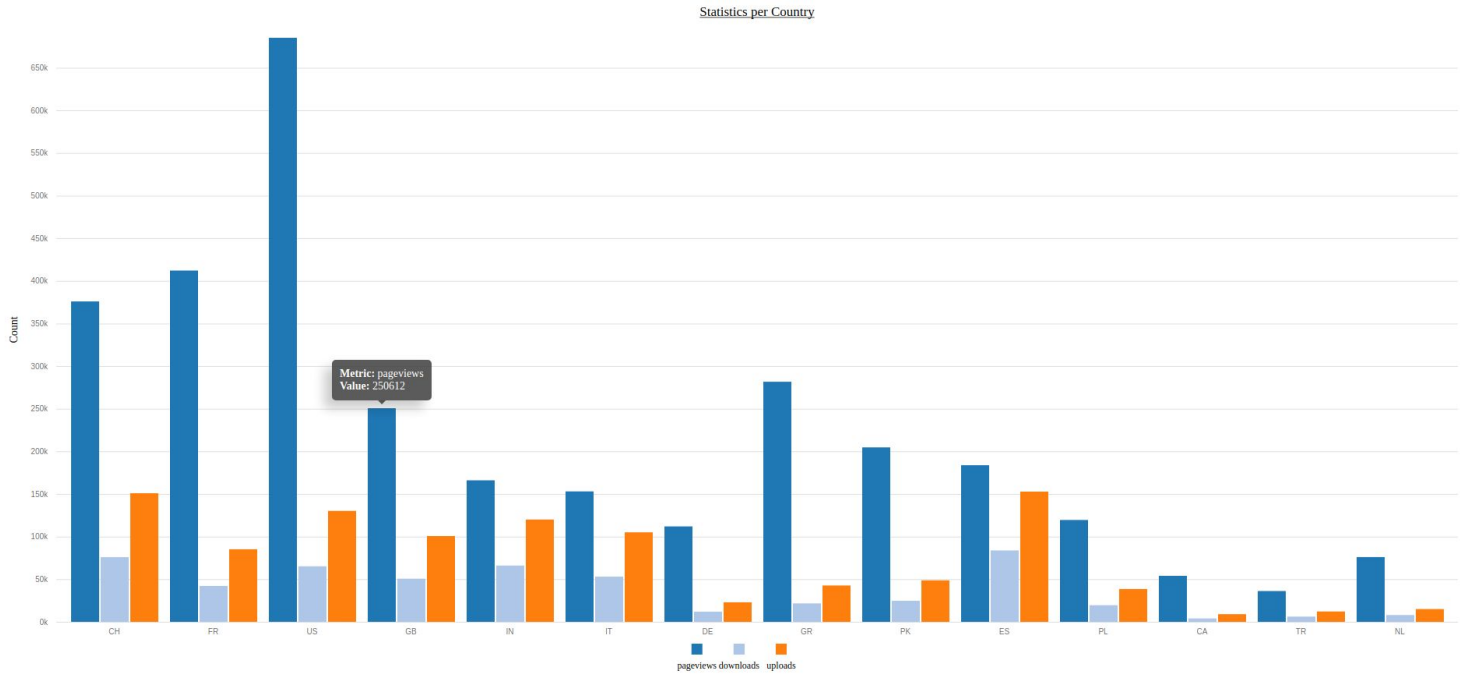
Multi Line/Area Chart



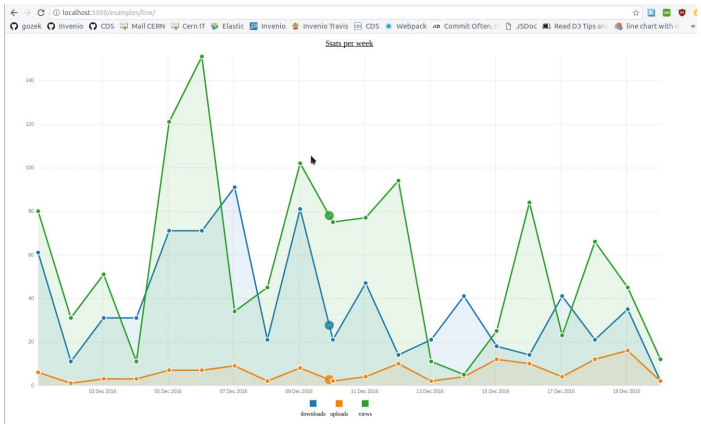
Bar Chart

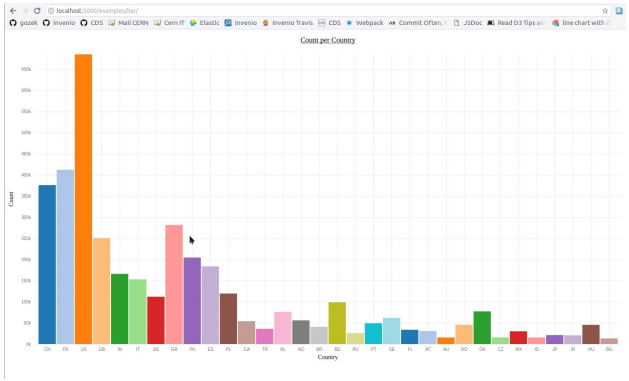

Count per Country
Count
650k
600k
550k
500k
450k
400k
350k
300k
250k
200k
150k
100k
50k
0k
Country: GR
Count: 281636
CH FR US GB IN IT DE GR PK ES PL CA TR NL NO BR BE RU PT SE FI AT AU RO DK CZ MX ID JP IR HU BG
Country

Grouped Bar Chart



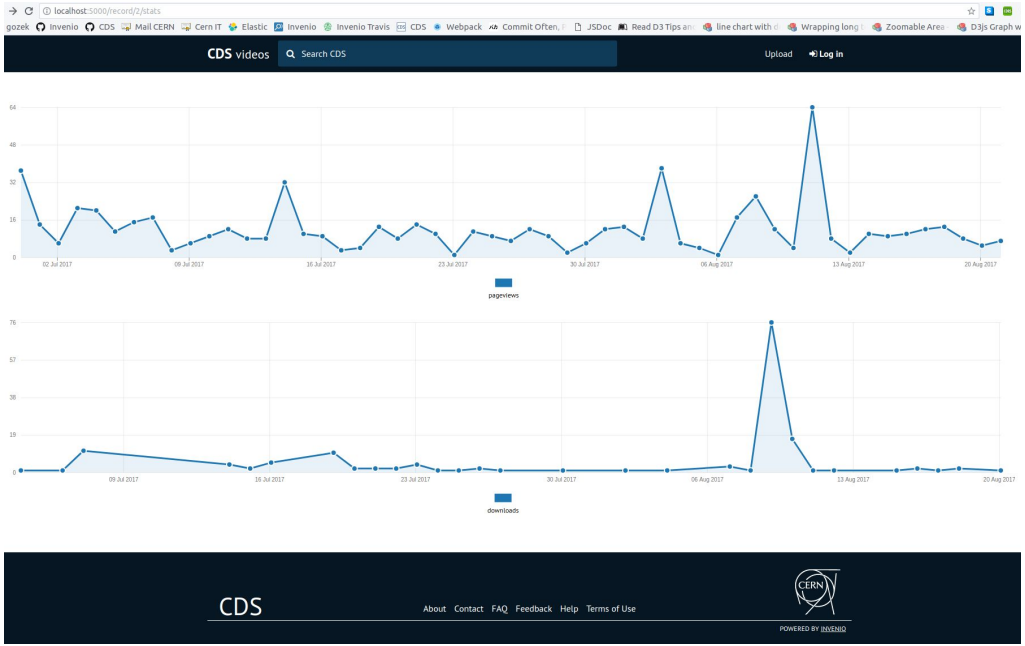
Zoomable, responsive, interactive





Current integration

- Set up instance of **`cds-labs`**
- Add a legacy **`stats module*`** to query the ES production cluster :
  `/api/stats/pageviews`
  `/api/stats/downloads`
- Digest response
- Extend **`records module`** with a testing page for statistics :
  `/record/<record_id>/stats`



**To be replaced with Invenio-Stats*

Documentation, tutorials and tests

Class: Graph

Graph(config, input, classElement)

Class representing an abstract Graph.

Constructor

`new Graph(config, input, classElement)`

Create a graph.

Parameters:

Name	Type	Description
config	Object	The configuration of the graph.
input	Array.<Object>	The input data.
classElement	String	The class of the SVG element.

Source: graph/graph.js, line 31

Methods

`enableZoom()`

Enable zoom in the graph.

Source: graph/graph.js, line 605

`getType() → {string}`

Get the type of the graph.

Source: graph/graph.js, line 68

Returns:

The type value.

Type
string

465 lines (432 sloc) | 13 KB   Raw Blame History

Simple Line Graph with default configuration

In this example we are going to create a simple line graph using the default configuration.

1. Before rendering

This is how the `HTML` document looks before rendering the line graph. In this example, the `div` element classed as `simple-line-graph` is the placeholder of our `SVG` graph and has fixed dimensions.

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <script src="../node_modules/d3/build/d3.min.js"></script>
    <script src="../node_modules/d3-svg-legend/d3-legend.min.js"></script>
    <script src="../node_modules/d3-tip/index.js"></script>
    <script src="../node_modules/lodash/lodash.min.js"></script>
    <style type="text/css"></style>
  </head>
  <body>
    <div class="simple-line-graph" style="width: 600px; height: 450px;"></div>
    <script src="./simple-line-graph.js"></script>
  </body>
</html>
```

2. Fetching the data

Assuming that we are working with `invenio-stats` as our backend, we have the following API and we'd like to display retrieved data using the `invenio-statistics-js`.

All files

86.97% Statements 414/476 60.47% Branches 156/258 82.22% Functions 74/90 86.85% Lines 403/464

File		Statements		Branches		Functions		Lines	
examples/bar		100%	2/2	100%	0/0	100%	0/0	100%	2/2
examples/groupedBar		100%	2/2	100%	0/0	100%	0/0	100%	2/2
examples/line		100%	3/3	100%	0/0	100%	0/0	100%	3/3
src		100%	4/4	100%	0/0	100%	0/0	100%	4/4
src/bar		75%	42/56	44.74%	17/38	68.75%	11/16	75%	42/56
src/graph		88.44%	199/225	74.59%	91/122	100%	16/16	88.02%	191/217
src/groupedBar		87.34%	69/79	52.94%	18/34	70.97%	22/31	88.16%	67/76
src/line		88.57%	93/105	46.88%	30/64	92.59%	25/27	88.46%	92/104

#TODOS

- Additional styling
- Package and publish on NPM
- Pie, Country charts (?)

Questions?

Special thanks to:

Harris Tzovanakis

Sebastian Witowski

More info:

https://github.com/CERNDocumentServer/invenio-stats-js

androulidakis.ioannis@gmail.com