

# Tema 10: Utilización de librerías javascript para gráficos estadísticos (d3.js, Highcharts).

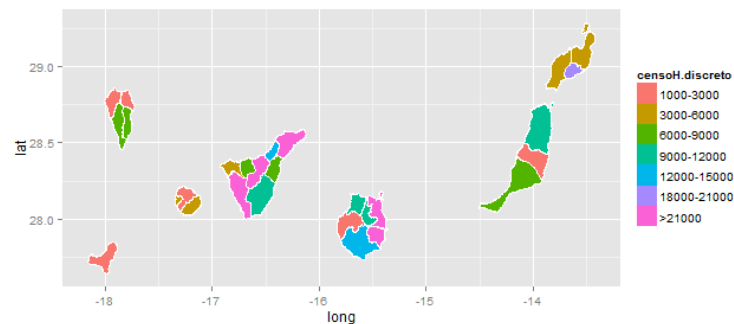
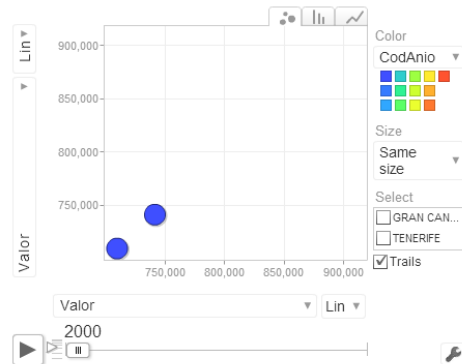
Máster Universitario en Ingeniería Informática  
Extracción y Explotación de la Información

# Índice

- Visualización gráfica en la web
  - Google charts

# Visualización gráfica en la web

- En temas anteriores se han explorado las primeras posibilidades para la representación gráfica de la información.
  - Gráficos interactivos (librería googleVis de R)
    - [http://cran.r-project.org/web/packages/googleVis/vignettes/googleVis\\_examples.html](http://cran.r-project.org/web/packages/googleVis/vignettes/googleVis_examples.html)
  - Gráficos geoespaciales (librería maptools de R)
    - <http://spatial.ly/r/>



# Visualización gráfica en la web

- En este tema veremos como podemos generar este tipo de visualizaciones en aplicaciones web donde otros usuarios puedan acceder a ellas.
- Ya hemos visto que la librería googleVis daba la posibilidad de generar el código javascript del gráfico para insertar en una web
- En este sentido, R permite generar el gráfico para unos datos de entrada determinados. Sin embargo, en ocasiones los usuarios seleccionan una fuente de datos a partir de los cuales desean generar el gráfico.

# Visualización gráfica en la web

- Los cuadros de mando o dashboards representan un tipo de herramienta de las conocidas como inteligencia de negocios (business intelligence).
- Otras herramientas de inteligencia de negocios son los OLAP (consulta de estructuras multidimensionales de datos), las aplicaciones de generación de informes, y las aplicaciones de minería de datos.

# Ventajas e inconvenientes

- La utilización de cuadros de mando presenta algunas ventajas e inconvenientes

## Ventajas

- El usuario no precisa instalar software en su ordenador salvo un navegador.
- El cuadro de mando se ejecuta, principalmente, en “client-side”.
- Resulta más fácil descargar los datos de interés en formatos portables (XML, CSV, etc..).
- Se integran fácilmente dentro de aplicaciones web.

## Inconvenientes

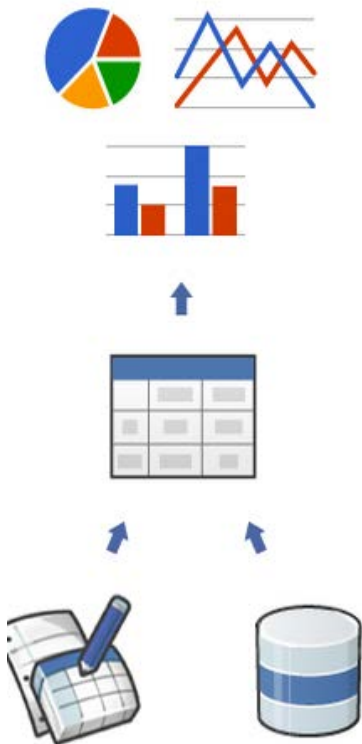
- En el lado de servidor (“server-side”) se requiere conectar con bases de datos, SOAP, etc..
- El desarrollo de los cuadros de mando puede ser algo lento (programación en lenguajes script).



- Nos centraremos en las utilidades para generar cuadros de mando.

# Google charts

- La API de Google Charts proporciona la posibilidad de representar gráficos sencillos de líneas a gráficos jerárquicos más complejos.



- La forma más habitual de utilizar esta API es mediante Javascript en la página web.
- Los orígenes de datos pueden ser los propios de Google (Google Spreadsheets, Google Fusion Tables, etc...). El acceso a los datos se lleva a cabo mediante lenguajes like-SQL.
- Los gráficos se pueden personalizar (aspecto y apariencia) y son altamente interactivos.

# Google charts

- Seguir el ejemplo desarrollado en:
  - [https://developers.google.com/chart/interactive/docs/quick\\_start](https://developers.google.com/chart/interactive/docs/quick_start)
- Podemos utilizar <http://jsfiddle.net/> para ejecutar HTML, Javascript, ....
- Existen algunos recursos que podemos utilizar para generar el código javascript:
  - <http://www.jonwinstanley.com/charts/>
  - <http://charts.hohli.com/new/>

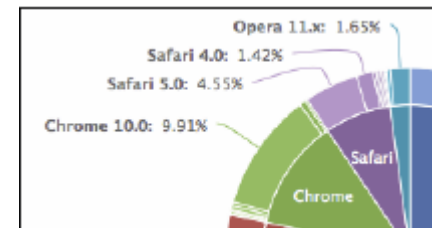
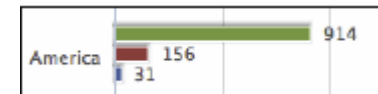


# Highcharts

- Esta librería (<http://highcharts.com>) está construida también con Javascript, aunque no está implementada totalmente en un framework particular. Por ejemplo, permite trabajar también con jQuery (<https://jquery.org/>) pero también en Mootools (<http://mootools.net/>) y Prototype (<http://prototypejs.org/>).
- También para esta librería existen algunos recursos en la web para generar el código:
  - <http://builtvisible.com/highcharts-generator/>

# Highcharts

- Highcharts tiene algunas características visuales que compiten con las ofrecidas por Google charts.
- Colores suaves, sin contrastes bruscos
- Efectos de sombra y bordes blancos  
(<http://localhost/highcharts/examples/bar-basic/index.htm>)
- Animaciones suaves y graduales  
(<http://localhost/highcharts/examples/pie-donut/index.htm>)
- Leyendas y descripciones emergentes que no atraen bruscamente la atención del usuario



# Highcharts

- Highcharts tiene una licencia libre Creative Commons no comercial para uso personal y fines no lucrativos.
- La API de Highcharts es muy sencilla.
- La parte de HTML+estilos CSS:

```
<html>
  <head>
    <style type='text/css'>
      #container {
        width: 300px; height: 300px; border: 1px solid #000;
        padding: 20px; margin: 10px;
      }
    </style>
  </head>
  <body>
    <div id='container'></div>
  </body>
</html>
```

# Highcharts

- La parte de Javascript:

```
<script src='./jquery/jquery.js'></script>
<script src='./highcharts/highcharts-all.js'></script>
<script type='text/javascript'>
    $(document).ready(function() {
        #
        # el código iría aquí
        #
    });
</script>
```

# Highcharts

- Un ejemplo sería el siguiente:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Highcharts Example</title>
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        var options = {
          chart: { type: 'bar' },
          title: { text: 'Un primer gráfico' },
          series: [{ name: 'Barra #1', data: [1,2,3,4] }]
        };
        $('#container').highcharts(options);
      });
    </script>
  </head>
  ...
```

# Highcharts

```
...  
<body>  
  <script src="js/highcharts.js"></script>  
  <script src="js/modules/exporting.js"></script>  
  <div id="container" style="min-width: 310px; max-width: 800px; height:  
400px; margin: 0 auto"></div>  
  </body>  
</html>
```

