Advanced Online Media
Dr. Cindy Royal
Texas State University - San Marcos
School of Journalism and Mass Communication

Using Google Visualizations to Present Data on a Web Site

Google has provided an amazing and free service that can add interactivity to your Web pages. Using the Google Visualizations API (application programming interface), you can insert interactive charts and graphs. Data can be coded by hand or come from an external source.

There are a number of different types of charts that you can select. Go to http://code.google.com/apis/charttools/ to get started. The Google Code Playground is a great place to see and work with code http://code.google.com/apis/ajax/playground/. Open Visualizations in the Pick API box.

Here are some examples and explanations that you can use:
Inserting data by hand in html code

```
<html>
  <head>
    <title>
     Google Visualization API Sample
    </title>

<!-- This is the Google AJAX API. Must be loaded. Just copy and paste -->
    <script type="text/javascript" src="http://www.google.com/jsapi"></script>

<!-- This is the Google Visualization API. See below or different packages -->
<!-- This first section loads the type of visualization package you want to use. -->
<script type="text/javascript">
     google.load('visualization', '1', {packages: ['piechart']});

// Set a callback to run when the Google Visualization API is loaded.
     google.setOnLoadCallback(drawVisualization);

//This is the data.
     function drawVisualization() {
       var data = new google.visualization.DataTable();
       data.addColumn('string', 'City');
       data.addColumn('number', 'Number of artists');

       data.addRows([
         ['Austin', 11],
         ['Dallas', 2],
         ['San Antonio', 2],
         ['San Marcos', 2],
```

```
        ['Houston', 15],
        ['Other', 7],

      ]);

// This is where the chart is drawn
      var chart = new
google.visualization.PieChart(document.getElementById('visualization'));
          chart.draw(data, {width: 400, height: 240, is3D: true, title: 'Page Name'});
     }


   </script>
  </head>
  <body style="font-family: Arial;border: 0 none;">
<!-- Then you need a div to put the visualization. The name here has to correspond to the
name you gave right above with the getElementByID method when the chart is drawn. -->

<div id="visualization" style="width: 300px; height: 300px;"></div>
  </body>
</html>
```

You may notice in some of the chart examples that the data is presented in a slightly different manner. The method above is a shorthand, putting the data in what is called a hash. You may see it in this long form method:

```
function drawVisualization() {
       // Create and populate the data table.
       var data = new google.visualization.DataTable();
       data.addColumn('string', 'Task');
       data.addColumn('number', 'Hours per Day');
       data.addRows(5);
       data.setValue(0, 0, 'Work');
       data.setValue(0, 1, 11);
       data.setValue(1, 0, 'Eat');
       data.setValue(1, 1, 2);
       data.setValue(2, 0, 'Commute');
       data.setValue(2, 1, 2);
       data.setValue(3, 0, 'Watch TV');
       data.setValue(3, 1, 2);
       data.setValue(4, 0, 'Sleep');
       data.setValue(4, 1, 7);
```

In the data.setValue method the first number is the row number, starting with 0. So, if there are 5 rows, you will see 0-4 there. The second number is the column, in this case either 0 or 1, since there are two columns. The third item is the actual data that goes in the column.

Some interactive sample chart formats
Must change in Visualization API and the object where the chart is drawn. There are others, but these are some of the most common you will use. See the Code Playground for more information and sample code.

Area Chart
{packages: ['areachart']} new google.visualization.AreaChart(...)

Bar Chart
{packages: ['barchart']} new google.visualization.BarChart(...)

Column Chart
{packages: ['columnchart']} new google.visualization.ColumnChart(...)

Line Chart
{packages: ['linechart']} new google.visualization.LineChart(...)

Pie Chart
{packages: ['piechart']} new google.visualization.PieChart(...)

Scatter Chart
{packages: ['scatterchart']} new google.visualization.ScatterChart(...)

When you draw any chart, notice the options for width, height, is3D and title.

Using a Google Spreadsheet to populate a Chart with Data

You can also put data in a Google spreadsheet and use that for your charts. You need a Google account to do this (free – like your gmail). Create a spreadsheet with your data in it, for example:

Austin	75
Dallas	50
Waco	1
San Antonio	30
Houston	50
San Marcos	20
Other	10

Then you must share the link to the spreadsheet, but do not allow the public to edit.
Click on Share tab, Get the Link to Share, and check the first box. Then grab that code. You can select a range by appending $range:A1:B6 to the url in the code below.

You can see more about this on the Code Playground. Look for Data Source Request in the Pick an API section.

```
<html>
  <head>
      <title>
     Google Visualization API Sample
    </title>
```

```
    <script type="text/javascript" src="http://www.google.com/jsapi"></script>
    <script type="text/javascript">
      google.load('visualization', '1', {packages: ['piechart']});
    </script>
    <script type="text/javascript">

    function drawVisualization() {
        var query = new google.visualization.Query(

'http://spreadsheets.google.com/ccc?key=0Aq18ZGjLDszFdHR6c2pDaHUxUC1hck4zZ29IdDJ
HSWc&hl=en');

      // Send the query with a callback function.
      query.send(handleQueryResponse);
    }

    function handleQueryResponse(response) {
      if (response.isError()) {
        alert('Error in query: ' + response.getMessage() + ' ' +
response.getDetailedMessage());
        return;
      }

      var data = response.getDataTable();
    var chart =  new
google.visualization.PieChart(document.getElementById('visualization'));
    chart.draw(data, {is3D: true});
    }

    google.setOnLoadCallback(drawVisualization);
    </script>
  </head>
  <body style="font-family: Arial;border: 0 none;">
    <div id="visualization" style="height: 400px; width: 400px;"></div>
  </body>
</html>
```

This has basically the same elements as the previous file, except that in the drawVisualization function, you query the spreadsheet, handle the query and then set up a variable to run the method getDataTable() (in lieu of creating a new data table in the code). Pretty simple.