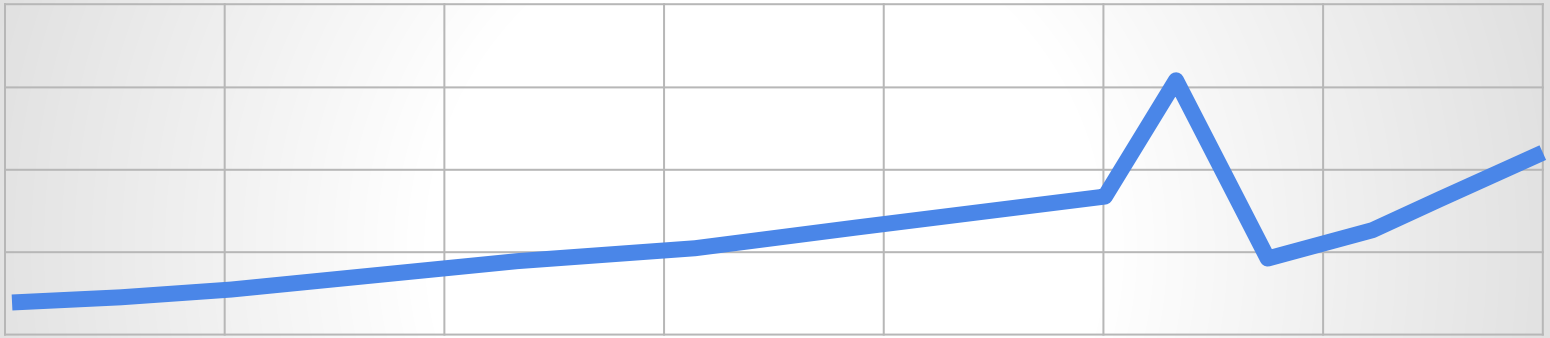
Connecting charts to live data



Presented at NICAR 2014, Baltimore

Timothy Barmann
The Providence Journal
tbarmann@providencejournal.com @timothybarmann

Presentation: bit.ly/1cfWL0O **Spreadsheet:** bit.ly/NcHbHL

Example: snowfall map

http://www.providencejournal.com/breaking-news/content/20140213-map-submit-snowfall-measurements-for-your-rhode-island-town.ece

MAP: Submit snowfall measurements for your Rhode Island town

February 13, 2014 01:26 PM

Comments 0

Share 85 Tweet 239 +1 2 Share 1 Pinterest 8 Email 4

Providence Journal reader snowfall report

Your snowfall reports are used to build our map! We'll average the latest three reports from each town and show the results on our color-coded map. Move your mouse over each town to see the snow depth. (If you are having trouble selecting a town, try using your mouse wheel or the arrow buttons on your keyboard.)

* Required

Location *
Choose the town of your snowfall report

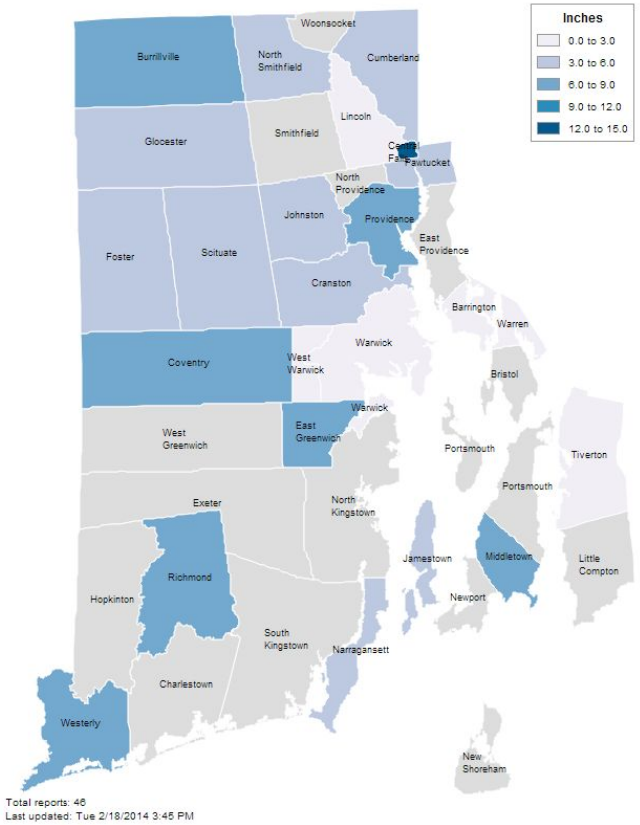
Inches *
How many inches have accumulated so far

Submit

Never submit passwords through Google Forms.

Powered by Google Drive This content is neither created nor endorsed by Google.

Report Abuse - Terms of Service - Additional Terms



Total reports: 46
Last updated: Tue 2/18/2014 3:45 PM

THE PROVIDENCE JOURNAL/TIMOTHY C. BARMANN

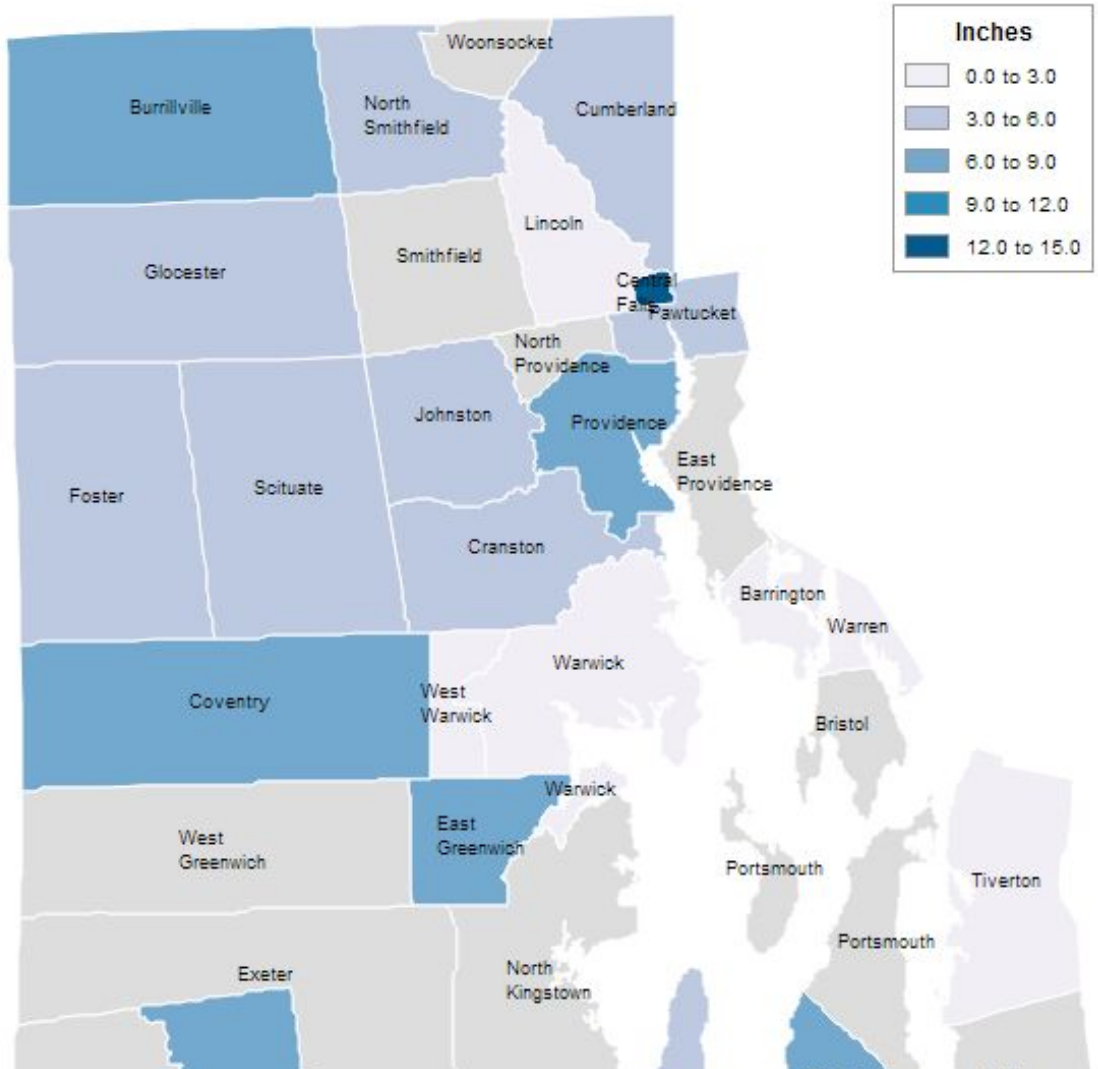
Providence Journal reader snowfall report

Your snowfall reports are used to build our map! We'll average the latest three reports from each town and show the results on our color-coded map. Move your mouse over each town to see the snow depth. (If you are having trouble selecting a town, try using your mouse wheel or the arrow buttons on your keyboard.)

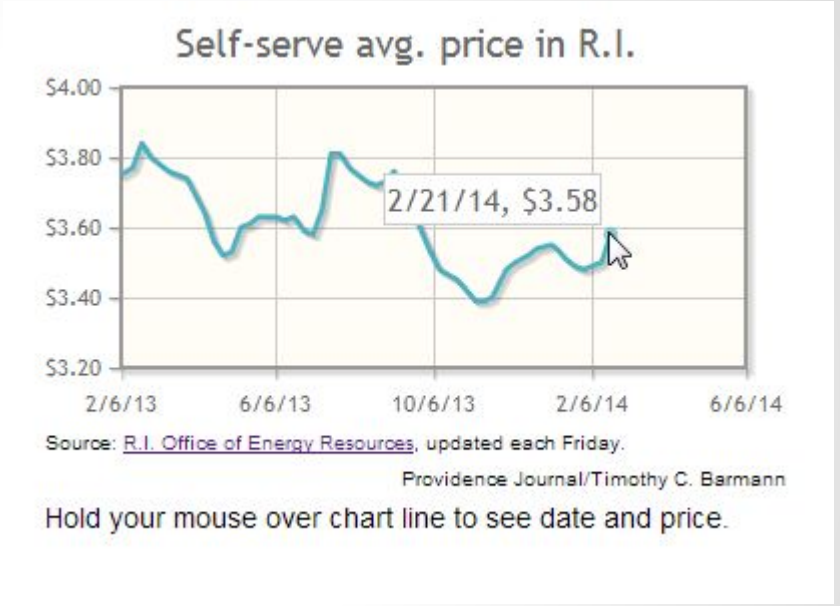
* Required

Location *
Choose the town of your snowfall report

Inches *
How many inches have accumulated so far

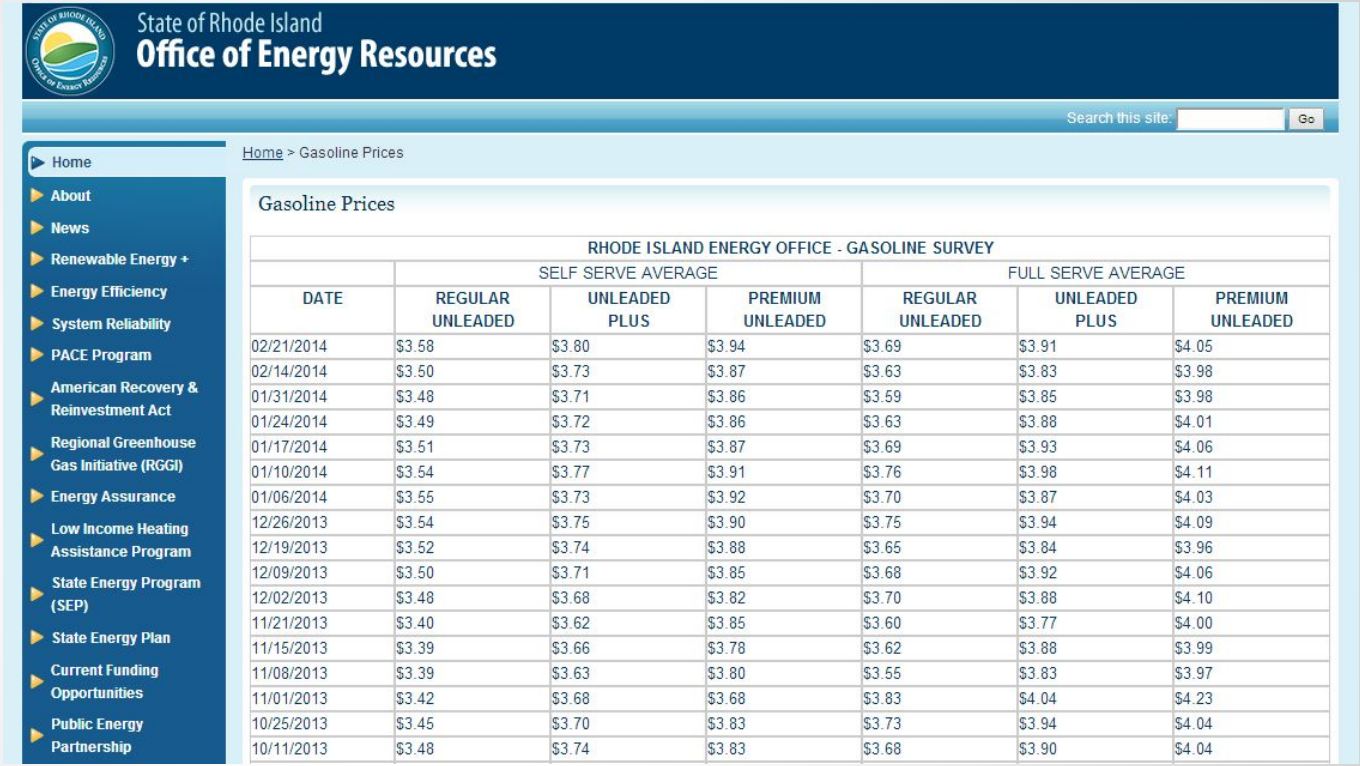


Example: fuel prices



http://www.providencejournal.com/traffic

Example: fuel prices



State of Rhode Island
Office of Energy Resources

Search this site: Go

- Home
- About
- News
- Renewable Energy +
- Energy Efficiency
- System Reliability
- PACE Program
- American Recovery & Reinvestment Act
- Regional Greenhouse Gas Initiative (RGGI)
- Energy Assurance
- Low Income Heating Assistance Program
- State Energy Program (SEP)
- State Energy Plan
- Current Funding Opportunities
- Public Energy Partnership

Home > Gasoline Prices

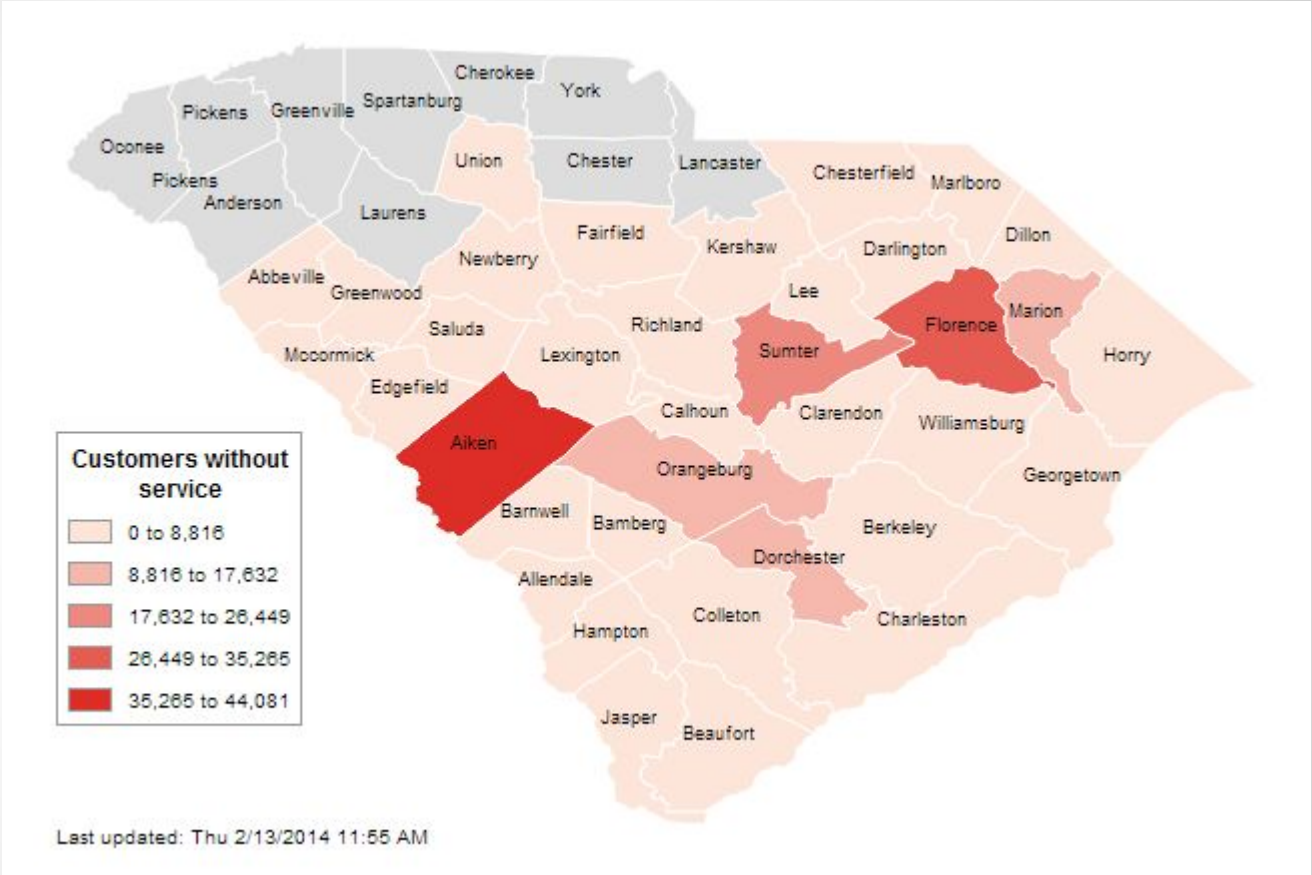
Gasoline Prices

RHODE ISLAND ENERGY OFFICE - GASOLINE SURVEY

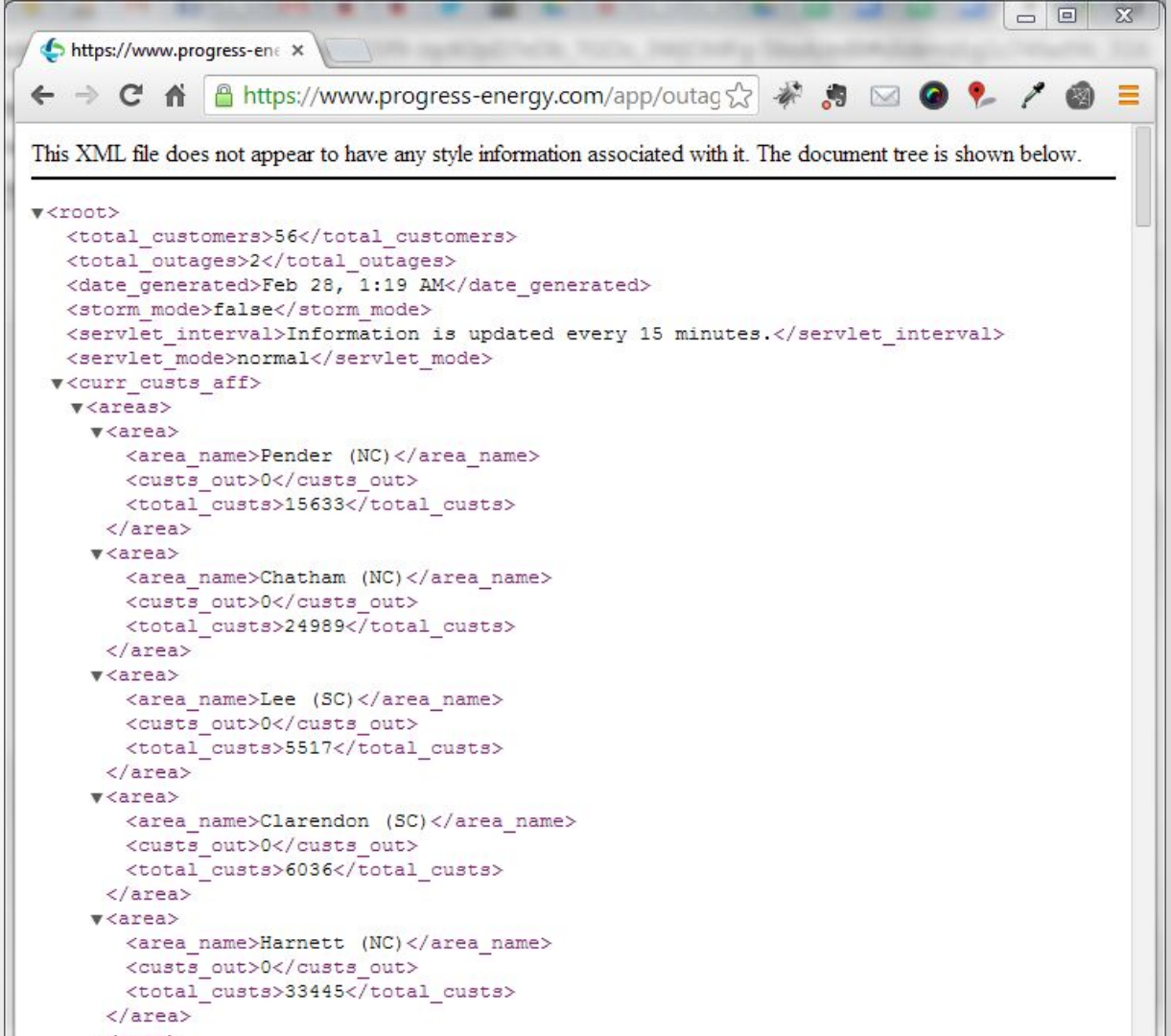
DATE	SELF SERVE AVERAGE			FULL SERVE AVERAGE		
	REGULAR UNLEADED	UNLEADED PLUS	PREMIUM UNLEADED	REGULAR UNLEADED	UNLEADED PLUS	PREMIUM UNLEADED
02/21/2014	$3.58	$3.80	$3.94	$3.69	$3.91	$4.05
02/14/2014	$3.50	$3.73	$3.87	$3.63	$3.83	$3.98
01/31/2014	$3.48	$3.71	$3.86	$3.59	$3.85	$3.98
01/24/2014	$3.49	$3.72	$3.86	$3.63	$3.88	$4.01
01/17/2014	$3.51	$3.73	$3.87	$3.69	$3.93	$4.06
01/10/2014	$3.54	$3.77	$3.91	$3.76	$3.98	$4.11
01/06/2014	$3.55	$3.73	$3.92	$3.70	$3.87	$4.03
12/26/2013	$3.54	$3.75	$3.90	$3.75	$3.94	$4.09
12/19/2013	$3.52	$3.74	$3.88	$3.65	$3.84	$3.96
12/09/2013	$3.50	$3.71	$3.85	$3.68	$3.92	$4.06
12/02/2013	$3.48	$3.68	$3.82	$3.70	$3.88	$4.10
11/21/2013	$3.40	$3.62	$3.85	$3.60	$3.77	$4.00
11/15/2013	$3.39	$3.66	$3.78	$3.62	$3.88	$3.99
11/08/2013	$3.39	$3.63	$3.80	$3.55	$3.83	$3.97
11/01/2013	$3.42	$3.68	$3.68	$3.83	$4.04	$4.23
10/25/2013	$3.45	$3.70	$3.83	$3.73	$3.94	$4.04
10/11/2013	$3.48	$3.74	$3.83	$3.68	$3.90	$4.04

http://www.energy.ri.gov/energyprices/gasprices.php

Example: power outages



Example: power outages



How to get the data

1. Scraping
2. Site’s API (if it offers one)

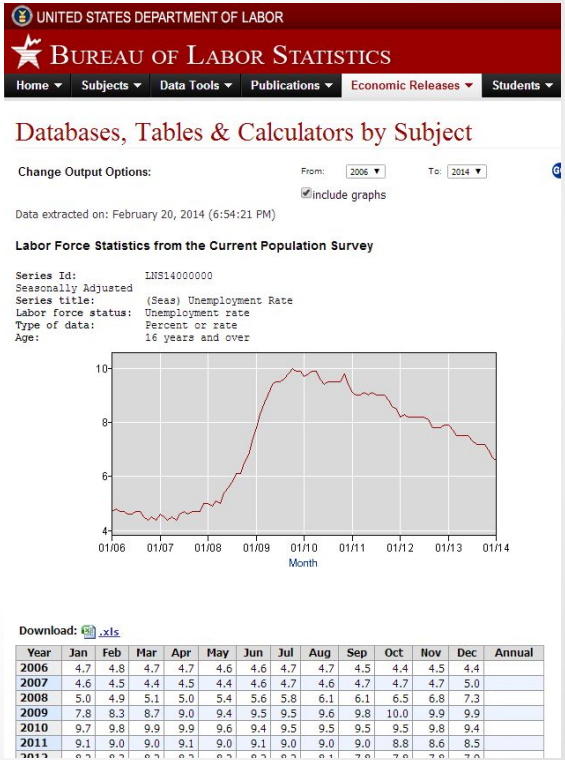
API is application programming interface.

A site that offers an API has decided to make a standard way of sharing its data

Examples of data sources with APIs

U.S. unemployment rate

How do you get a graph like this on your website and make it update automatically?



Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Annual
2006	4.7	4.8	4.7	4.7	4.6	4.6	4.7	4.7	4.5	4.4	4.5	4.4	
2007	4.6	4.5	4.4	4.5	4.4	4.6	4.7	4.6	4.7	4.7	4.7	5.0	
2008	5.0	4.9	5.1	5.0	5.4	5.6	5.8	6.1	6.1	6.5	6.8	7.3	
2009	7.8	8.3	8.7	9.0	9.4	9.5	9.5	9.6	9.8	10.0	9.9	9.9	
2010	9.7	9.8	9.9	9.9	9.6	9.4	9.5	9.5	9.5	9.5	9.8	9.4	
2011	9.1	9.0	9.0	9.1	9.0	9.1	9.0	9.0	9.0	8.8	8.6	8.5	

The good news …

The U.S. BLS does offer an API

HTTP Type:	**GET**
URL:	http://api.bls.gov/publicAPI/v1/timeseries/data/ *<series_id>*
Payload:	series_id
Example Payload:	LAUCN21219003

Source: http://www.bls.gov/developers/api_signature.htm

The good news … (con't.)

The good news … (con't.)


```
{
   "status":"REQUEST_SUCCEEDED",
   "responseTime":12,
   "message":[

   ],
   "Results":{
      "series":[
         {
            "seriesID":"LNS14000000",
            "data":[
               {
                  "year":"2014",
                  "period":"M01",
                  "periodName":"January",
                  "value":"6.6",
                  "footnotes":[
                     {

                     }
                  ]
               },
               {
                  "year":"2013",
```


JSON
JavaScript Object Notation

JSON formatter: http://jsonformatter.curiousconcept.com/

The bad news …

Browsers cannot directly read this type of file

- Same origin policy: browsers do not allow JavaScript code to load a plain JSON file that resides on a different domain
- Exceptions: JSONP with callback, Cross-origin resource sharing (CORS)

Source: http://en.wikipedia.org/wiki/Same_origin_policy ,
http://en.wikipedia.org/wiki/Cross-origin_resource_sharing

Solutions …

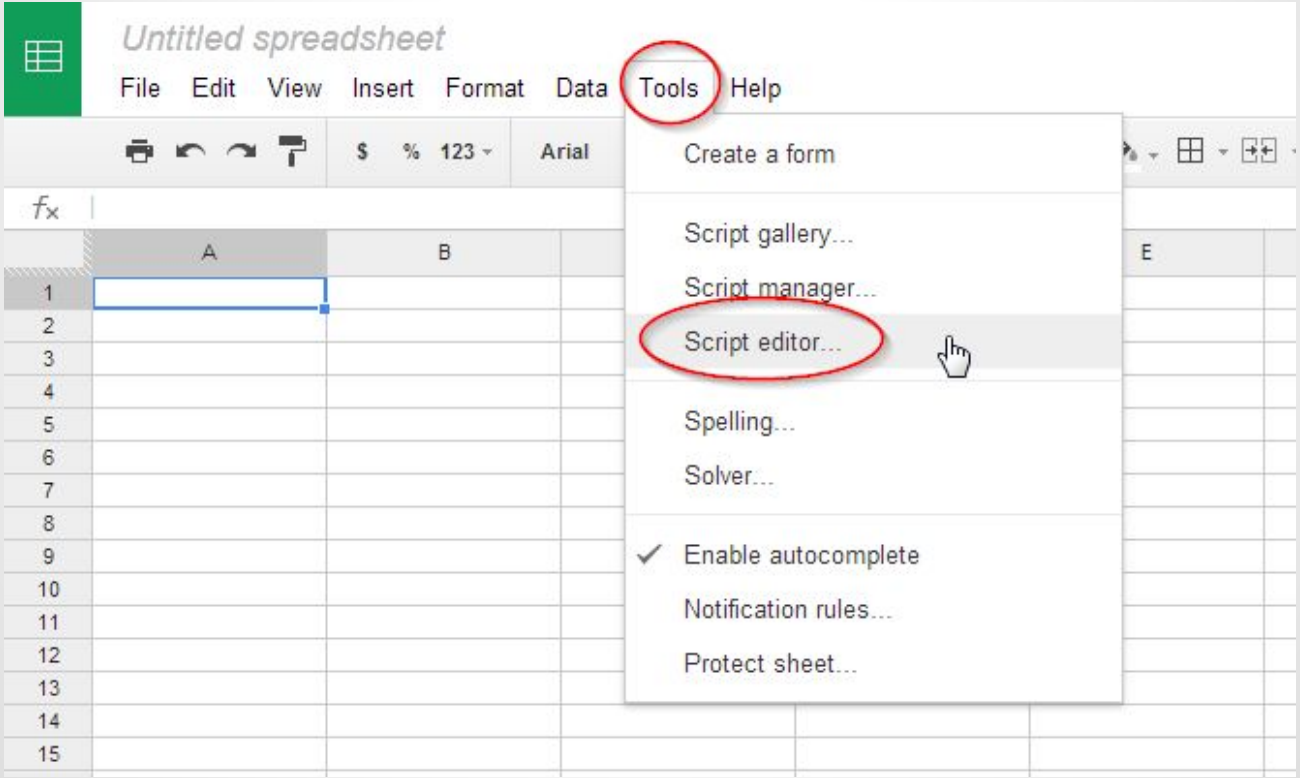
“Proxy” or a go-between:

- Google spreadsheets
- Yahoo Query Language
- Your own program on a server acting as a proxy

Google spreadsheets

- Google Application Script can fetch data from an API and store it in a spreadsheet
- A script can be set up to run automatically according to a schedule
- Data can be pulled out of a spreadsheet and used for visualizations

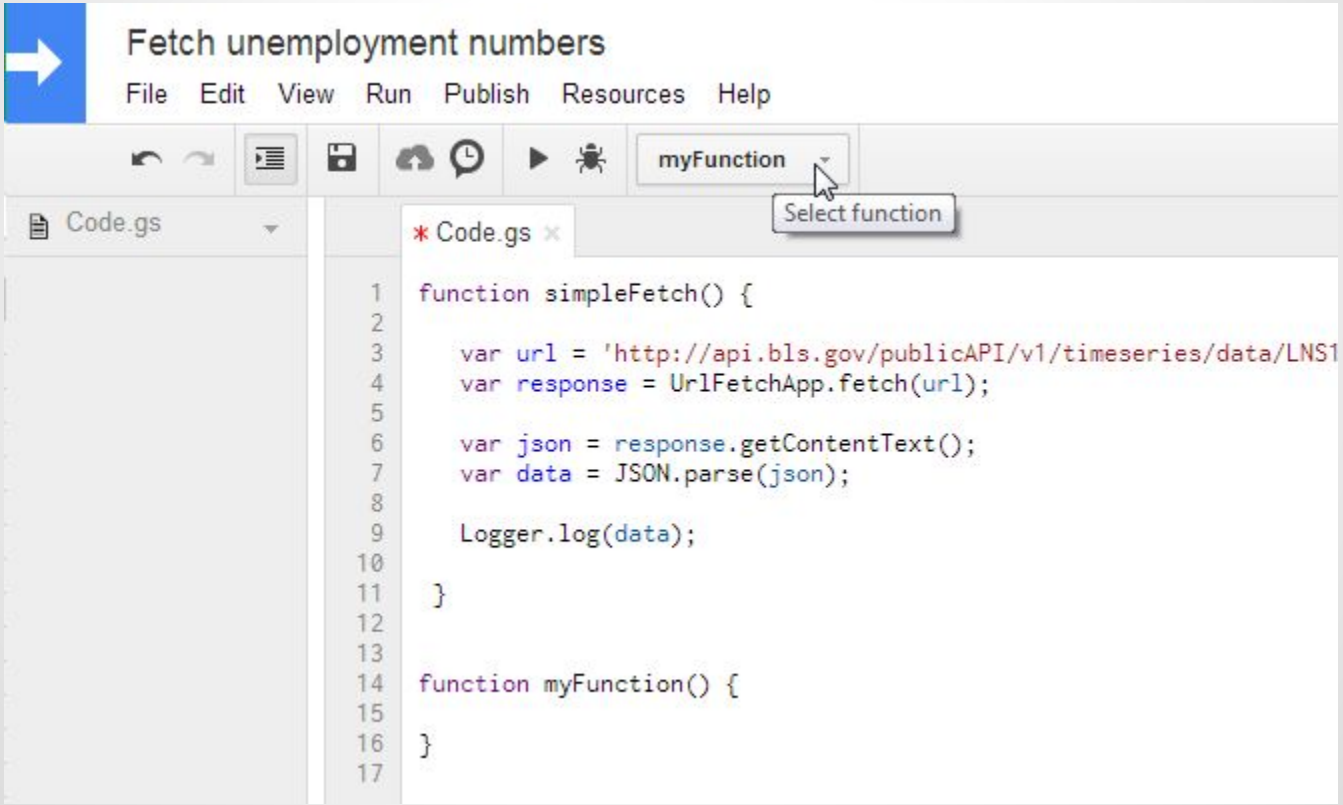
Google spreadsheets



Google spreadsheets

```
function simpleFetch() {

   var url = 'http://api.bls.gov/publicAPI/v1/timeseries/data/LNS14000000';
   var response = UrlFetchApp.fetch(url);

   var json = response.getContentText();
   var data = JSON.parse(json);

   Logger.log(data);

 }
```

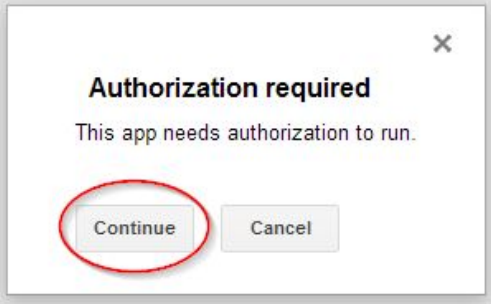
Google spreadsheets



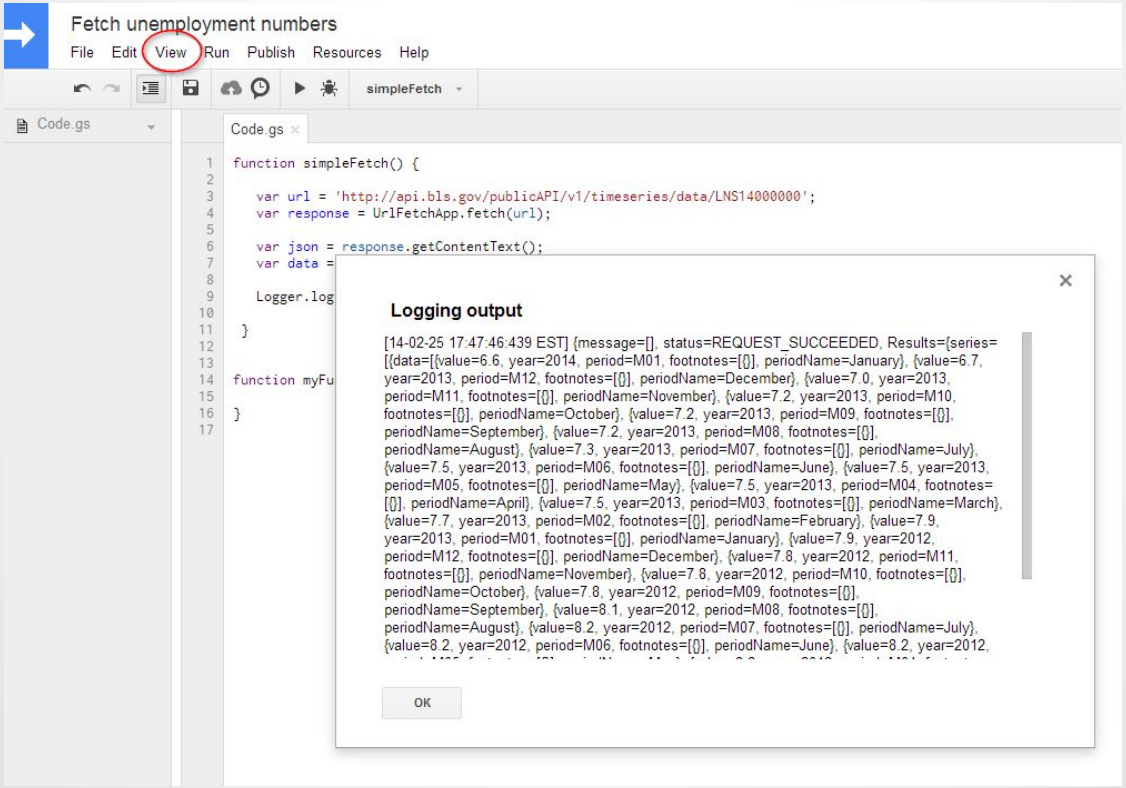
Google spreadsheets

Code.gs

```
function simpleFetch() {

   var url = 'http://api.bls.gov/publicAPI/v1/timeseries/data/LNS14000000';
   var response = UrlFetchApp.fetch(url);

   var json = response.getContentText();
   var data = JSON.parse(json);

   Logger.log(data);

 }


function myFunction() {

}
```



Google spreadsheets



Google spreadsheets

Code.gs

```
function simpleFetch() {

   var url = 'http://api.bls.gov/publicAPI/v1/timeseries/data/LNS14000000';
   var response = UrlFetchApp.fetch(url);

   var json = response.getContentText();
   var data = JSON.parse(json);

   var niceData = JSON.stringify(data,null," ");
   Logger.log(niceData);

 }

function myFunction() {

}
```

Google spreadsheets

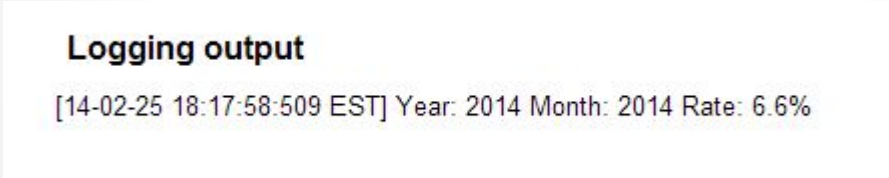
Google spreadsheets

Google spreadsheets

Code.gs

```
function simpleFetch() {

   var url = 'http://api.bls.gov/publicAPI/v1/timeseries/data/LNS14000000';
   var response = UrlFetchApp.fetch(url);

   var json = response.getContentText();
   var data = JSON.parse(json);

   var year = data.Results.series[0].data[0].year;
   var month = data.Results.series[0].data[0].year;
   var rate = data.Results.series[0].data[0].value;

 Logger.log("Year: " + year + " Month: " + month + " Rate: " + rate + "%");

 }
```

Google spreadsheets



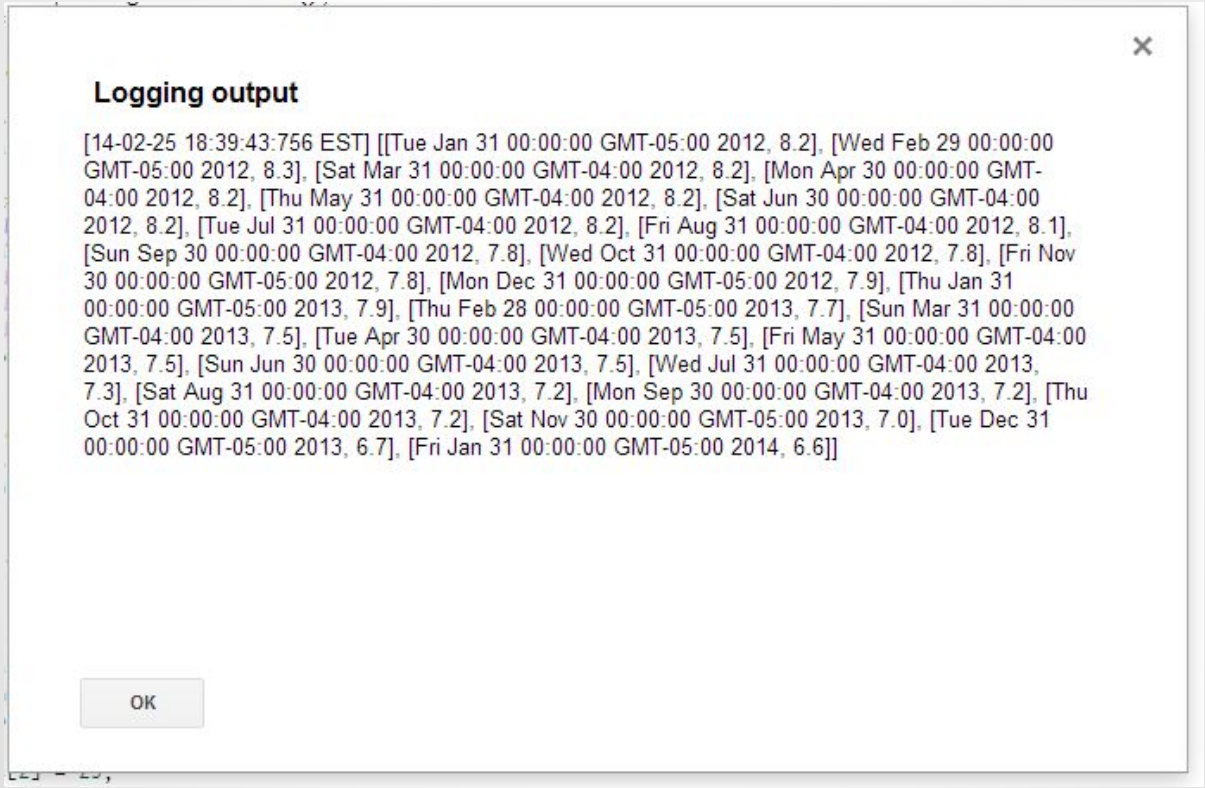
Google spreadsheets

```
function simpleFetch() {

  var url = 'http://api.bls.gov/publicAPI/v1/timeseries/data/LNS14000000';
  var response = UrlFetchApp.fetch(url);

  var json = response.getContentText();
  var data = JSON.parse(json);

  var dataArray = data.Results.series[0].data;
  var values = [];

  for (var x=0; x<dataArray.length; x++) {
    var thisMo = parseInt(dataArray[x].period.slice(-2),10);
    var thisYr = dataArray[x].year;
    var thisDy = daysInMonth(thisMo,thisYr); // last day of the month
    var thisDateStr = thisMo + "/" + thisDy + "/" + thisYr;
    var thisDate = new Date(thisDateStr);
    values.push([thisDate,dataArray[x].value]);
  }

  // sort by date, ascending
  values.sort(function(a,b){
    return (a[0] - b[0]);
  });

  Logger.log(values);
}
```

Google spreadsheets

```
// ---------------------------------------------------------------------------
function daysInMonth(mo,yr) {
    var isLeap = ((yr % 4) == 0 && ((yr % 100) != 0 || (yr % 400) == 0));
    var allMonths =  [null,31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31];
  if (isLeap) {
    allMonths[2] = 29;
  }
  return allMonths[mo];
}
```

Google spreadsheets



Google spreadsheets

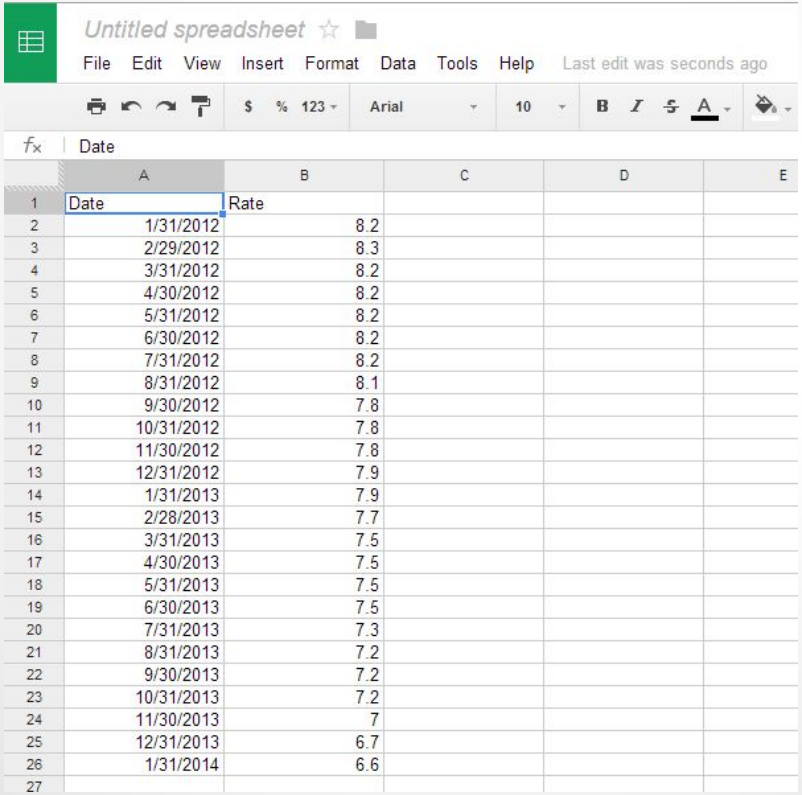
Code.gs

```
  var ss = SpreadsheetApp.getActiveSpreadsheet();

    if (ss.getSheetByName('us_unemployment')===null) {
      var sheet = ss.insertSheet('us_unemployment');
    }
    else {
      var sheet = ss.getSheetByName('us_unemployment');
    }

  var range = sheet.getDataRange();
  var lastRow = range.getLastRow();

  if (lastRow === 1) {
    range = sheet.getRange("A1:B1");
    range.setValues([['Date','Rate']]);
  }

  var startRow = range.getLastRow() + 1;
  range = sheet.getRange(startRow,1,values.length,values[0].length);
  range.setValues(values);

  removeDuplicates();

}
```

Google spreadsheets

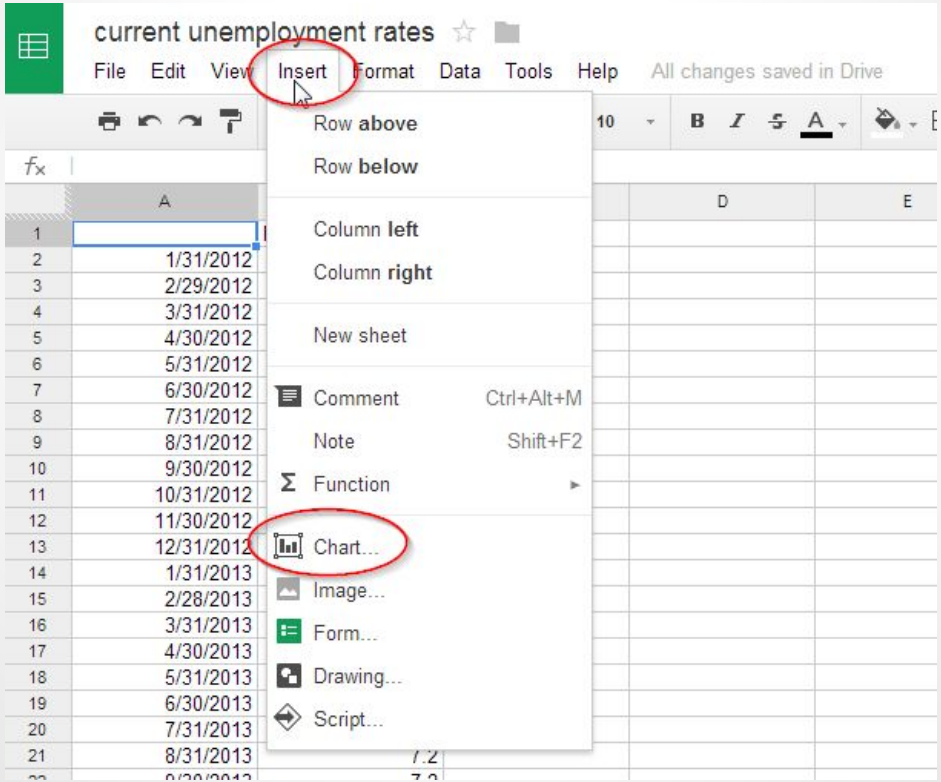
```
// ---------------------------------------------------------------------------------------
// from https://developers.google.com/apps-script/articles/removing_duplicates
function removeDuplicates() {
  var sheet = SpreadsheetApp.getActiveSheet();
  var data = sheet.getDataRange().getValues();
  var newData = new Array();
  for(i in data){
    var row = data[i];
    var duplicate = false;
    for(j in newData){
      if(row.join() == newData[j].join()){
        duplicate = true;
      }
    }
    if(!duplicate){
      newData.push(row);
    }
  }
  sheet.clearContents();
  sheet.getRange(1, 1, newData.length, newData[0].length).setValues(newData);
}
```

Google spreadsheets

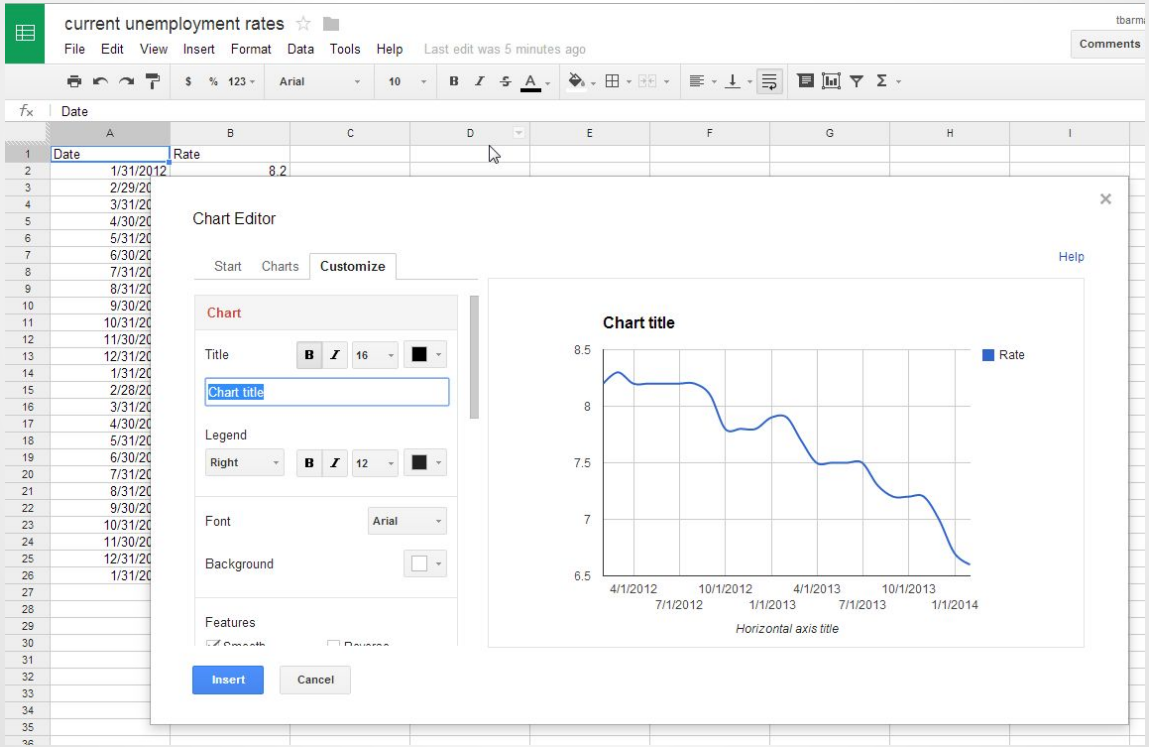


	A	B	C	D	E
1	Date	Rate			
2	1/31/2012	8.2			
3	2/29/2012	8.3			
4	3/31/2012	8.2			
5	4/30/2012	8.2			
6	5/31/2012	8.2			
7	6/30/2012	8.2			
8	7/31/2012	8.2			
9	8/31/2012	8.1			
10	9/30/2012	7.8			
11	10/31/2012	7.8			
12	11/30/2012	7.8			
13	12/31/2012	7.9			
14	1/31/2013	7.9			
15	2/28/2013	7.7			
16	3/31/2013	7.5			
17	4/30/2013	7.5			
18	5/31/2013	7.5			
19	6/30/2013	7.5			
20	7/31/2013	7.3			
21	8/31/2013	7.2			
22	9/30/2013	7.2			
23	10/31/2013	7.2			
24	11/30/2013	7			
25	12/31/2013	6.7			
26	1/31/2014	6.6			
27					

Google spreadsheets

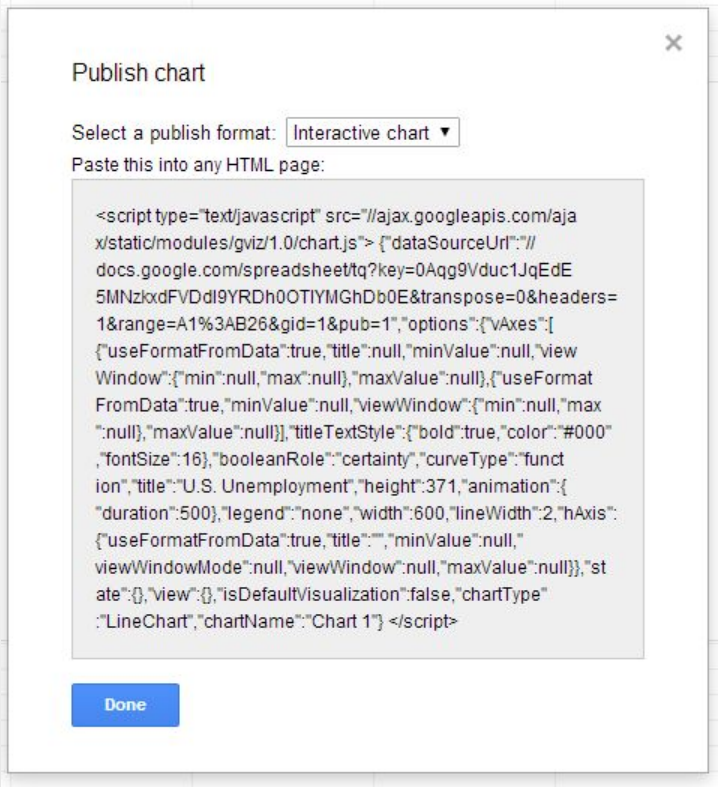


Google spreadsheets



Google spreadsheets

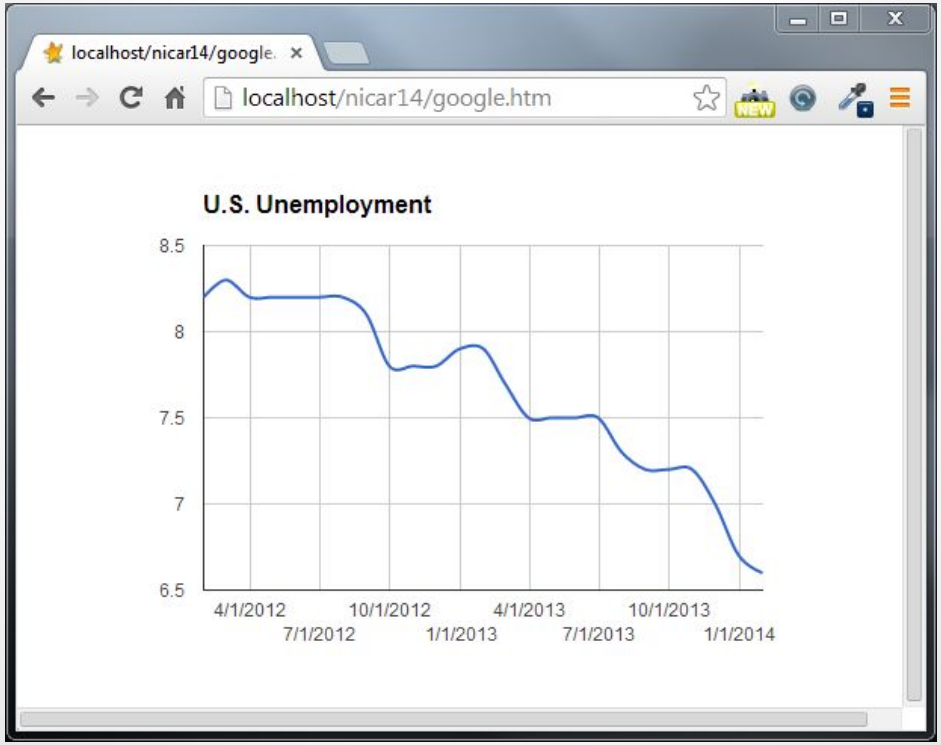
Google spreadsheets



Google spreadsheets

```
<html>
<head>
<style type="text/css">
</style>

</head>
<body>

<script type="text/javascript" src="//ajax.googleapis.com/ajax/static/modules/gviz/1.0/chart.js">
{"dataSourceUrl":"//docs.google.com/spreadsheet/tq?key=0Aqg9Vduc1JqEdE5MNzkxdFVDdl9YRDh0OTlYMGhDb0
E&transpose=0&headers=1&range=A1%3AB26&gid=1&pub=1","options":{"vAxes":[{"useFormatFromData":true,
"title":null,"minValue":null,"viewWindow":{"min":null,"max":null},"maxValue":null},{"useFormatFrom
Data":true,"minValue":null,"viewWindow":{"min":null,"max":null},"maxValue":null}],"titleTextStyle"
:{"bold":true,"color":"#000","fontSize":16},"booleanRole":"certainty","curveType":"function","titl
e":"U.S.
Unemployment","height":371,"animation":{"duration":500},"legend":"none","width":600,"lineWidth":2,
"hAxis":{"useFormatFromData":true,"title":"","minValue":null,"viewWindowMode":null,"viewWindow":nu
ll,"maxValue":null}},"state":{},"view":{},"isDefaultVisualization":false,"chartType":"LineChart","
chartName":"Chart 1"} </script>

</body>
</html>
```

Google spreadsheets



Google spreadsheets

```
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.3/jquery.min.js"
type="text/javascript"></script>
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">
google.load("visualization", '1', {packages:['corechart']});
google.setOnLoadCallback(drawChart);

function drawChart() {
  var url =
'https://docs.google.com/spreadsheet/pub?key=0Aqg9Vduc1JqEdE5MNzkxdFVDdl9YRDh0OTlYMGhDb0E&single=true&gi
d=1&output=html';
  var query = new google.visualization.Query(url);
  query.send(handleQueryResponse);
}
function handleQueryResponse(response) {
  if (response.isError()) {
    alert('Error in query: ' + response.getMessage() + ' ' + response.getDetailedMessage());
    return;
  }
// continued
```

Google spreadsheets

```
  var data = response.getDataTable();
  var chart = new google.visualization.LineChart(document.getElementById('columnchart'));
  chart.draw(data,
      {
      legend: {
            position: 'none'
      },
      width:800,
      height:400,
      vAxis: {minValue: 5, title: 'Percent'}
      });
} // handleQueryResponse
</script>
<title>U.S. Unemployment rate</title>
</head>
<body>
<span id='columnchart' style="width:800px;"></span>
</body>
</html>
```

Google spreadsheets

Google spreadsheets

```
<html>
<head>

<link rel="stylesheet" type="text/css" href="js/jqplot/jquery.jqplot.css" />

<script src="js/jquery.min.js" type="text/javascript"></script>
<script type="text/javascript" src="js/jqplot/jquery.jqplot.js"></script>
<script type="text/javascript" src="js/jqplot/plugins/jqplot.highlighter.min.js"></script>
<script type="text/javascript" src="js/jqplot/plugins/jqplot.cursor.min.js"></script>
<script type="text/javascript"
src="js/jqplot/plugins/jqplot.dateAxisRenderer.min.js"></script>
<script type="text/javascript"
src="js/jqplot/plugins/jqplot.canvasTextRenderer.min.js"></script>
<script type="text/javascript"
src="js/jqplot/plugins/jqplot.canvasAxisLabelRenderer.min.js"></script>
```

Google spreadsheets

```
<script type="text/javascript"
src='https://www.google.com/jsapi?autoload={"modules":[{"name":"visualization","version":"1
"}]}'></script>
<script type="text/javascript">

  google.setOnLoadCallback(drawVisualization);
  var line = [];

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

'https://docs.google.com/spreadsheet/pub?key=0Aqg9Vduc1JqEdE5MNzkxdFVDdl9YRDh0OTlYMGhDb0E&s
heet=us_unemployment');
      query.send(handleQueryResponse);

  }
```

Google spreadsheets

```
  function handleQueryResponse(response) {

    if (response.isError()) {
      alert('Error in query: ' + response.getMessage() + ' ' +
response.getDetailedMessage());
      return;
    }
      $.each(response.n.xf,function(){
                  line.push([this.c[0].v,this.c[1].v]);
      });

      drawChart();

  }
```

Google spreadsheets

```
function drawChart() {
      var plot1 = $.jqplot('chart1', [line], {
       title:'U.S. Unemployment Rate',
       width:960,
       series: [
                 {
                     lineWidth:3,
                     markerOptions: {size:6 }
           }

       ],

       axesDefaults: {
            labelRenderer: $.jqplot.CanvasAxisLabelRenderer
       },
```

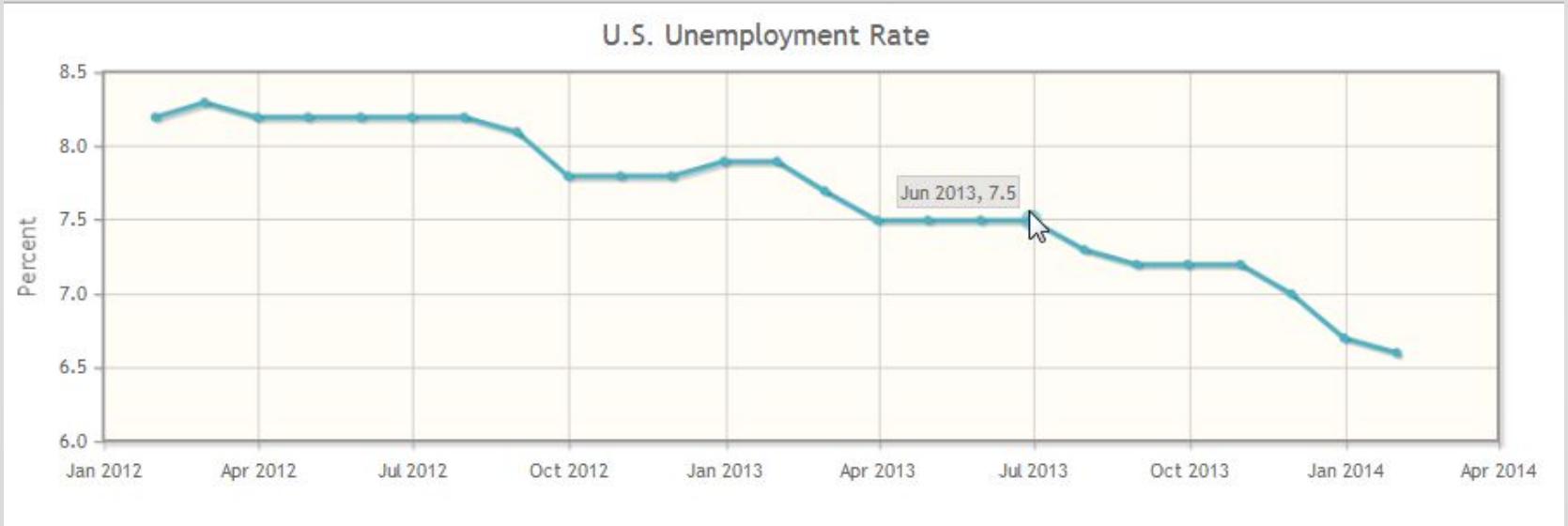
Google spreadsheets

```
axes:{
        xaxis:{
          renderer:$.jqplot.DateAxisRenderer,
          pad: 0,
          tickOptions:{
            formatString:'%b %Y'
          }
        },
        yaxis:{
          label: "Percent",
          tickOptions:{
            formatString:'%.1f'
            }
        }
      },
```

Google spreadsheets

```
      highlighter: {
        show: true,
        sizeAdjust: 7.5
      },
      cursor: {
        show: false
      }
  });
}

</script>
</head>
<body>
<div id="chart1" width="960" style="width:960px;"></div>
</body>
</html>
```

Google spreadsheets



Google spreadsheets

Release Calendar

SHARE ON:

BY MONTH

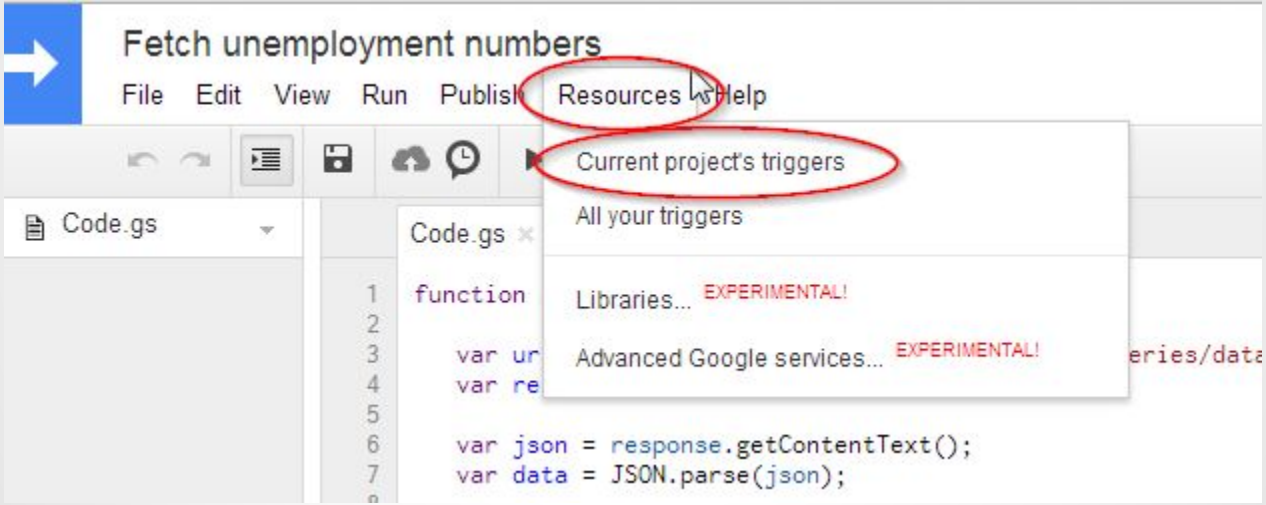
- NOVEMBER 2013
- DECEMBER 2013
- JANUARY 2014
- FEBRUARY 2014
- MARCH 2014
- APRIL 2014
- MAY 2014
- JUNE 2014
- JULY 2014
- AUGUST 2014
- SEPTEMBER 2014
- OCTOBER 2014
- NOVEMBER 2014
- DECEMBER 2014
- ENTIRE YEAR, 2014

Schedule of Releases for the Employment Situation

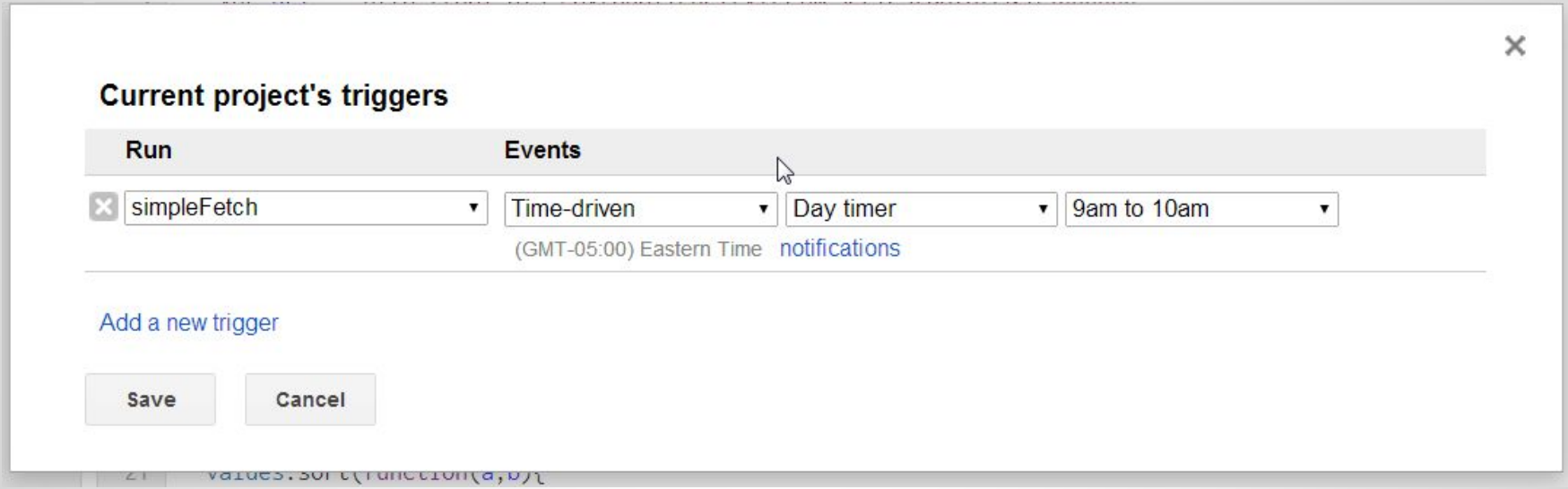
Reference Month	Release Date	Release Time
October 2013	Nov. 08, 2013	08:30 AM
November 2013	Dec. 06, 2013	08:30 AM
December 2013	Jan. 10, 2014	08:30 AM
January 2014	Feb. 07, 2014	08:30 AM
February 2014	Mar. 07, 2014	08:30 AM
March 2014	Apr. 04, 2014	08:30 AM
April 2014	May 02, 2014	08:30 AM
May 2014	Jun. 06, 2014	08:30 AM
June 2014	Jul. 03, 2014	08:30 AM
July 2014	Aug. 01, 2014	08:30 AM
August 2014	Sep. 05, 2014	08:30 AM
September 2014	Oct. 03, 2014	08:30 AM
October 2014	Nov. 07, 2014	08:30 AM
November 2014	Dec. 05, 2014	08:30 AM

Source: http://www.bls.gov/schedule/news_release/empsit.htm

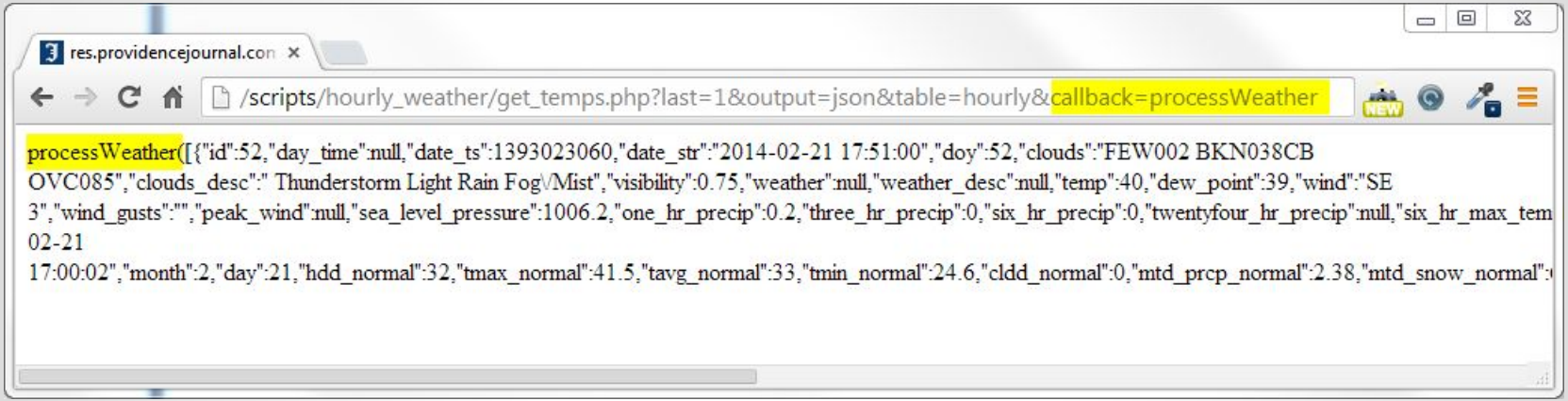
Google spreadsheets



Google spreadsheets



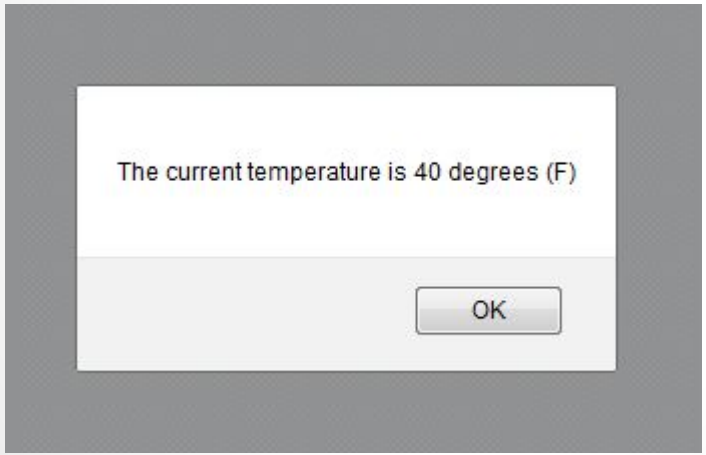
What is a callback?



Using callbacks with jQuery

```
<html>
<head>
<script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
<script type="text/javascript">
  function processWeather(data){
      alert("The current temperature is " + data[0].temp + " degrees (F)");
  }
  $.ajax({
url:"http://res.providencejournal.com/weather/assets/scripts/hourly_weather/get_temps.php?last=1&output=j
son&table=hourly",
    type: "GET",
    dataType:"jsonp",
    success: processWeather,
    crossDomain: true
  });
</script>
</head>
<body>
</body>
</html>
```

Result ...



Using callback, jQuery to graph data

```
<html>
<head>
  <link rel="stylesheet" type="text/css"
href="http://res.providencejournal.com/global-assets/styles/jquery.jqplot.css" />
  <script type="text/javascript"
src="http://res.providencejournal.com/global-assets/scripts/jquery-1.8.3.js"></script>
  <script language="javascript" type="text/javascript"
src="http://res.providencejournal.com/global-assets/scripts/jquery.jqplot.min.js"></script>
  <script type="text/javascript"
src="http://res.providencejournal.com/global-assets/scripts/underscore-min.js"></script>
  <script language="javascript" type="text/javascript"
src="http://res.providencejournal.com/global-assets/scripts/jqplot.dateAxisRenderer.min.js"></script>
  <script type="text/javascript"
src="http://res.providencejournal.com/global-assets/scripts/jqplot.highlighter.js"></script>
  <script type="text/javascript">

      $(document).ready(function(){
```

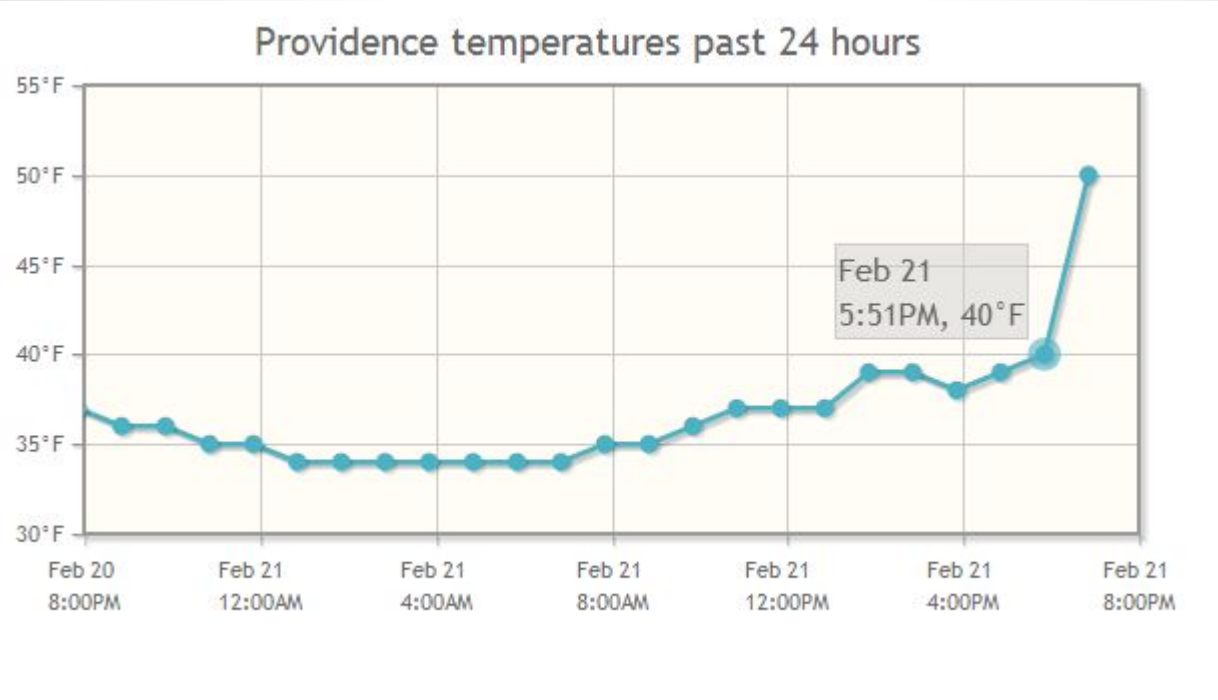
Using callback, jQuery to graph data

```
function processWeather(data){
  var times = _.map(_.pluck(data,'date_ts'),function(num){return num * 1000});
  var temperatures = _.pluck(data,'temp');
  var temperature_data = _.zip(times,temperatures);
  var ticks = _.map(_.pluck(data,'date_ts'),function(num){return (Math.round(num/3600)*3600)* 1000});
  var plot = $.jqplot('temperatures', [temperature_data], {
            title: 'Providence temperatures past 24 hours',
            axes:{
              xaxis:{
                renderer:$.jqplot.DateAxisRenderer,
                tickInterval: '4 hour',
                min:ticks[0],
                max:ticks[ticks.length-1],
                tickOptions:{ formatString:'%b %#d<br/>%#I:%M%p'}
              },
              yaxis:{
                tickOptions:{formatString:'%d&deg;F'}
              }
            },
            highlighter: {show: true,sizeAdjust: 7.5},
            cursor: {show: false}
  }); // plot
} // processWeather()
```

Using callback, jQuery to graph data

```
   $.ajax({
      url:"http://res.providencejournal.com/weather/assets/scripts/hourly_weather/get_temps.php?last=24&ou
      tput=json&table=hourly",
      type: "GET",
      dataType:"jsonp",
      success: processWeather,
      crossDomain: true
   }); // ajax
}); // document ready

</script>
</head>
<body>
      <div id="temperatures"></div>
</body>
</html>
```

Result ...


Providence temperatures past 24 hours
55°F
50°F
45°F
40°F
35°F
30°F
Feb 21
5:51PM, 40°F
Feb 20
8:00PM
Feb 21
12:00AM
Feb 21
4:00AM
Feb 21
8:00AM
Feb 21
12:00PM
Feb 21
4:00PM
Feb 21
8:00PM

Questions ...