

Testing the jupyterviz package

Nathan Carter, October 2018

Load the module

```
In [1]: LoadPackage( "jupyterviz" );
```

```
Out[1]: true
```

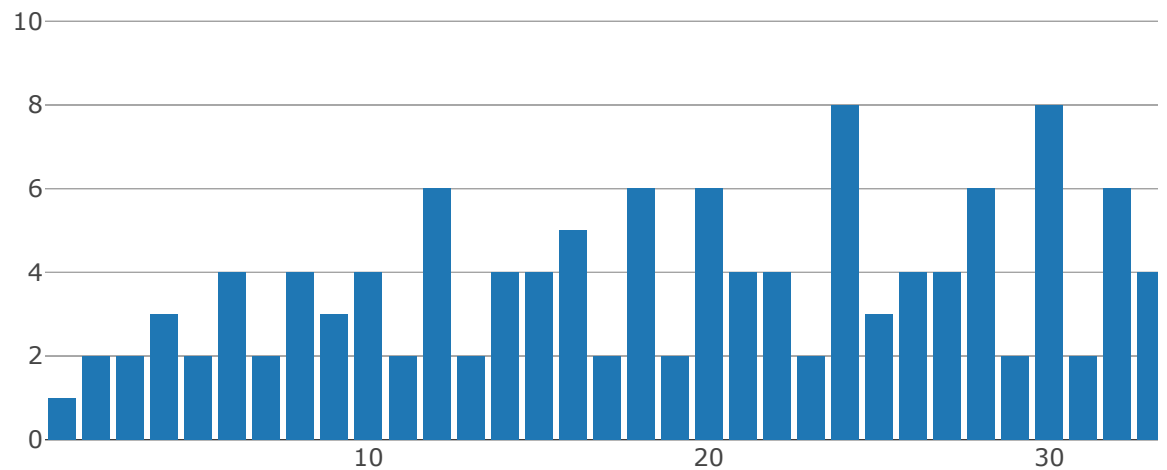
Test visualization with [Plotly \(https://plot.ly/\)](https://plot.ly/).

For $n = 1$ to 50, how many divisors does n have?

Hover over the graph for popup information.

```
In [2]: CreateVisualization( rec(
  tool := "plotly",
  data := rec(
    data := [
      rec(
        x := [1..50],
        y := List( [1..50], n -> Length(DivisorsInt(n)) ),
        type := "bar"
      )
    ],
    layout := rec( height := 400 )
  )
), "" );
```

Out[2]:



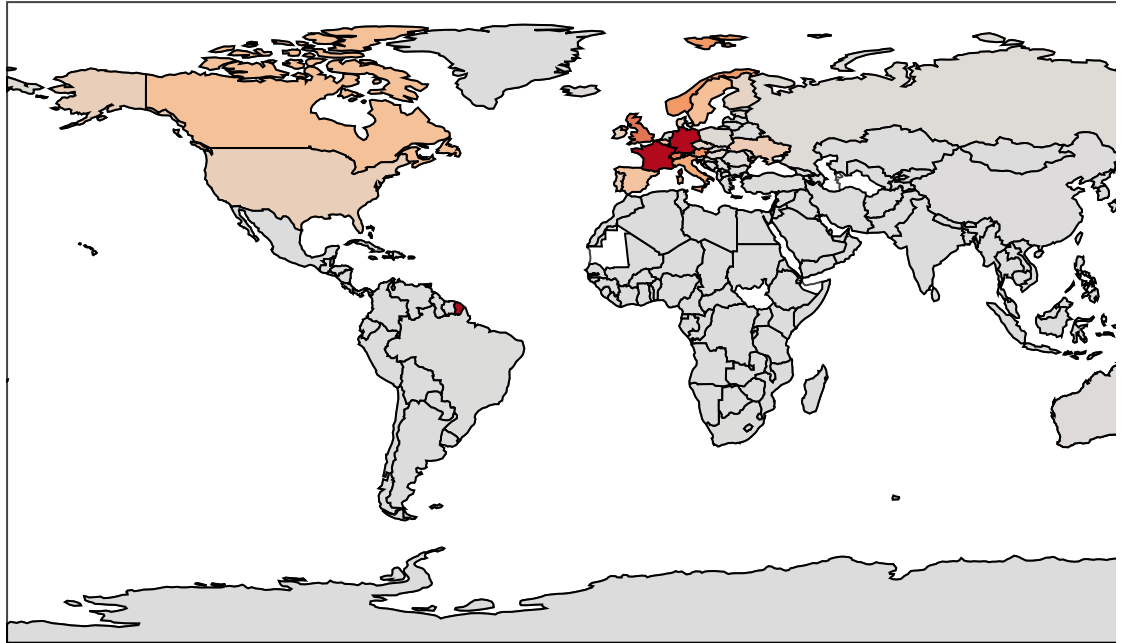
Load more complex Plotly chart from JSON file

This JSON file was [downloaded from the Plotly gallery \(https://plot.ly/~Dreamshot/9298\)](https://plot.ly/~Dreamshot/9298), and contains data about number of electric vehicle charge points installed in 2017, worldwide.

Hover the graph for more information.

```
In [7]: map := JsonStringToGap(  
  ReadAll( InputTextFile( "EV Charge Points.json" ) ) );;  
map.layout := rec( height := 500 );;  
CreateVisualization( rec(  
  tool := "plotly",  
  data := map  
) , "" );
```

Out[7]:



Test visualization with [ChartJS](https://www.chartjs.org/)

For $n = 1$ to 30, how many groups are there of size n ?

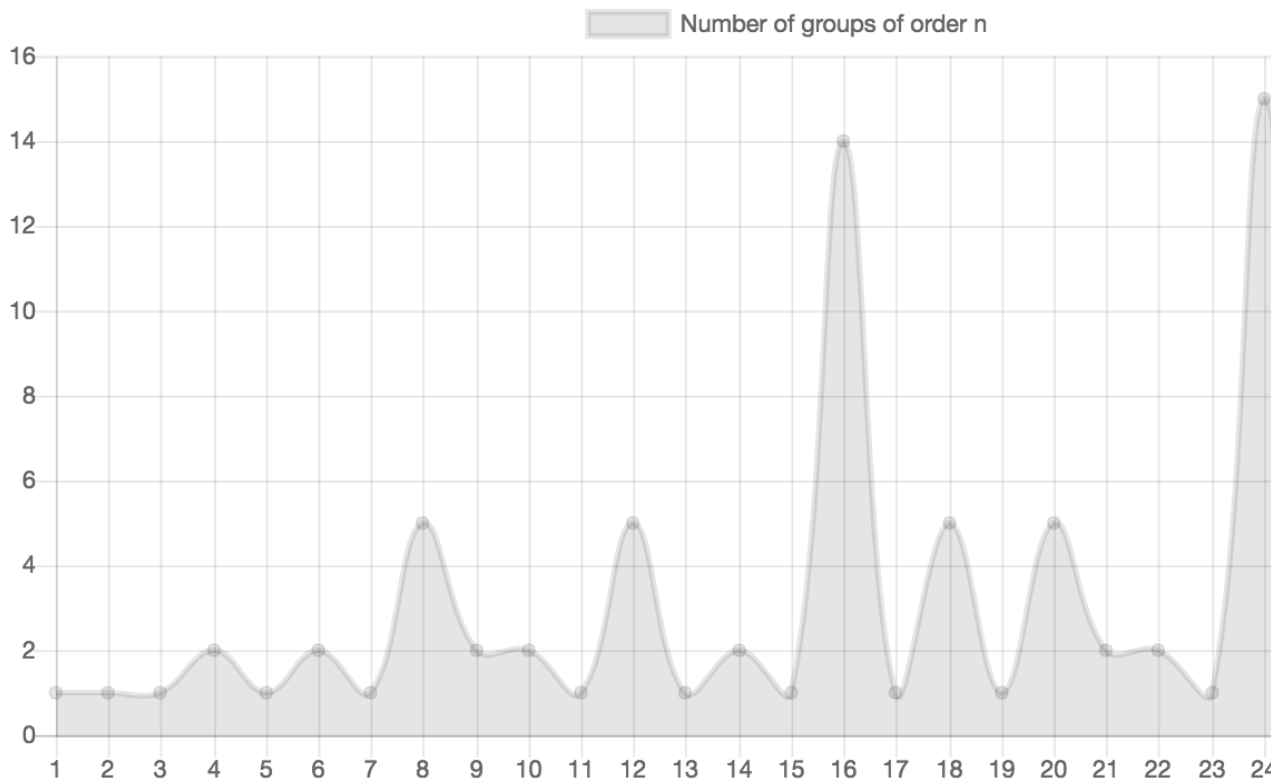
Hover the graph for more information.

```

In [8]: CreateVisualization( rec(
  tool := "chartjs",
  data := rec(
    type := "line",
    data := rec(
      labels := [1..30],
      datasets := [
        rec(
          label := "Number of groups of order n",
          data := List( [1..30], n -> NrSmallGroups( n ) )
        ),
      ]
    )
  ), "" );

```

Out[8]:



Test visualization with [CanvasJS](https://canvasjs.com/)

Just graphing 10 random integers in the range $\{1, \dots, 100\}$.

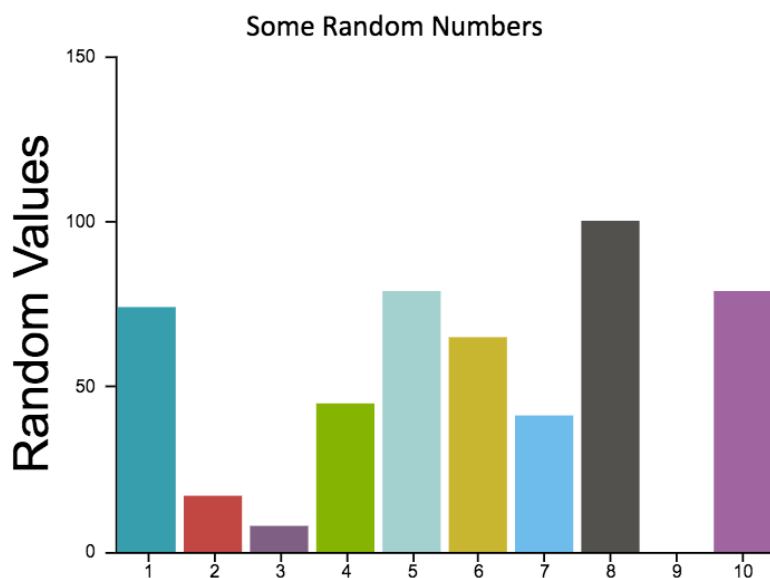
Hover the graph for more "information."

```

In [10]: CreateVisualization( rec(
  tool := "canvasjs",
  data := rec(
    animationEnabled := true,
    width := 400,
    height := 300,
    theme := "light2",
    title := rec( text := "Some Random Numbers" ),
    axisY := rec(
      title := "Random Values",
      titleFontSize := 24
    ),
    data := [
      rec(
        type := "column",
        dataPoints := List( [1..10],
          n -> rec( x := n, y := Random( 0, 100 ) )
        )
      )
    ]
  )
), "" );

```

Out[10]:



CanvasJS.com (<http://canvasjs.com/>)

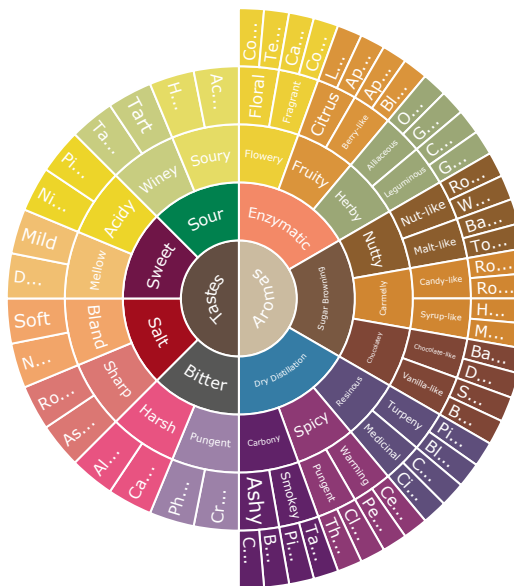
Test visualization with [AnyChart](https://www.anychart.com/)

This one was downloaded from the [AnyChart gallery](https://www.anychart.com/products/anychart/gallery/) (<https://www.anychart.com/products/anychart/gallery/>) to show the flexibility of this toolkit, which is probably the most robust of all the ones shown on this page.

```
In [12]: CreateVisualization( rec(
    tool := "anychart",
    data := JsonStringToGap(
        ReadAll( InputTextFile( "anychart-sample.json" ) ) )
), "" );
```

Out[12]:

Coffee Flavour Wheel



Test visualization with [Cytoscape](http://js.cytoscape.org/)

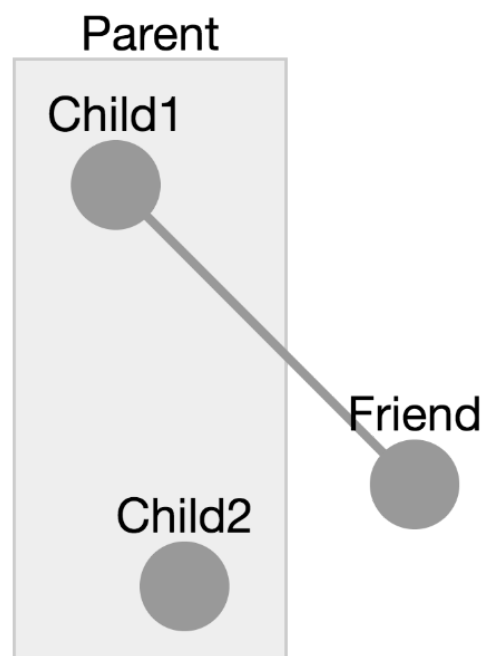
This simple graph was taken [from the Cytoscape documentation](http://js.cytoscape.org/#core/initialisation) (<http://js.cytoscape.org/#core/initialisation>) and slightly manipulated as part of this test.

```

In [14]: CreateVisualization( rec(
  tool := "cytoscape",
  height := 400,
  data := rec(
    elements := [
      rec( # node 1
        group := "nodes",
        data := rec( id := "Child1", parent := "Parent" ),
        position := rec( x := 100, y := 100 ),
        selected := false,
        selectable := true,
        locked := false,
        grabbable := true
      ),
      rec( # node 2
        data := rec( id := "Friend" ),
        renderedPosition := rec( x := 200, y := 200 )
      ),
      rec( # node 3
        data := rec( id := "Child2", parent := "Parent" ),
        position := rec( x := 123, y := 234 )
      ),
      rec( # node parent
        data := rec( id := "Parent", position := rec( x := 200, y :=
      ),
      rec( # edge 1
        data := rec( id := "Edge1", source := "Child1", target := "F
      )
    ],
    layout := rec( name := "preset" ),
    style := [
      rec( selector := "node", style := rec( content := "data(id)" ) )
    ]
  )
), "" );

```

Out[14]:



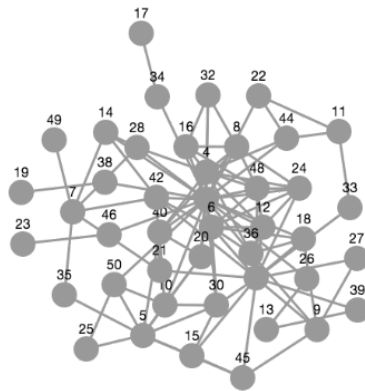
**Test creation of a graph with GAP code, then using
Cytoscape for layout**


```

In [20]: N := 50;;
elements := [ ];;
roots := [ ];;
for i in [2..N] do
  Add( elements, rec( data := rec( id := String( i ) ) ) );
  if IsPrime( i ) then
    Add( roots, i );
  fi;
  for j in [2..i-1] do
    if i mod j = 0 then
      Add( elements, rec( data := rec( source := String( j ), target :
      fi;
    od;
  od;
od;
CreateVisualization( rec(
  tool := "cytoscape",
  height := 600,
  data := rec(
    elements := elements,
    layout := rec( name := "cose" ),
    style := [
      rec( selector := "node", style := rec( content := "data(id)" ) )
    ]
  )
), "" );

```

Out[20]:



In []: