nyaplotjs Documentation

Release 2.0.0

Naoki Nishida

November 17, 2015

Contents

Nyaplotjs is a rich plotting library built on the top of d3.js. Its goal is to provide the useful JSON interface for plotting to other programming languages including Ruby and Python.

Nyaplotjs is originally developed as a back-end library for Nyaplot, a visualization library written in Ruby. A simple JavaScript interface was added in v2 and Python wrapper is now being developed.

CHAPTER 1

Getting Started

1.1 Installation

Run the line shown below to install nyaplotjs to your local project.

```
npm install nyaplot
```

1.2 Try nyaplotjs using Simple API

```
<html>
  <head>
    <script src='http://d3js.org/d3.v3.min.js'></script>
    <script src='https://cdn.rawgit.com/domitry/Nyaplotjs/a1b10b68f08cfbd5afc4e145103ac46297951d45/re
    <script>
    window.onload = function(){
       // code for plotting from here
    };
    </script>
  </head>
  <body>
    <div id="vis"></div>
  </body>
</html>
```

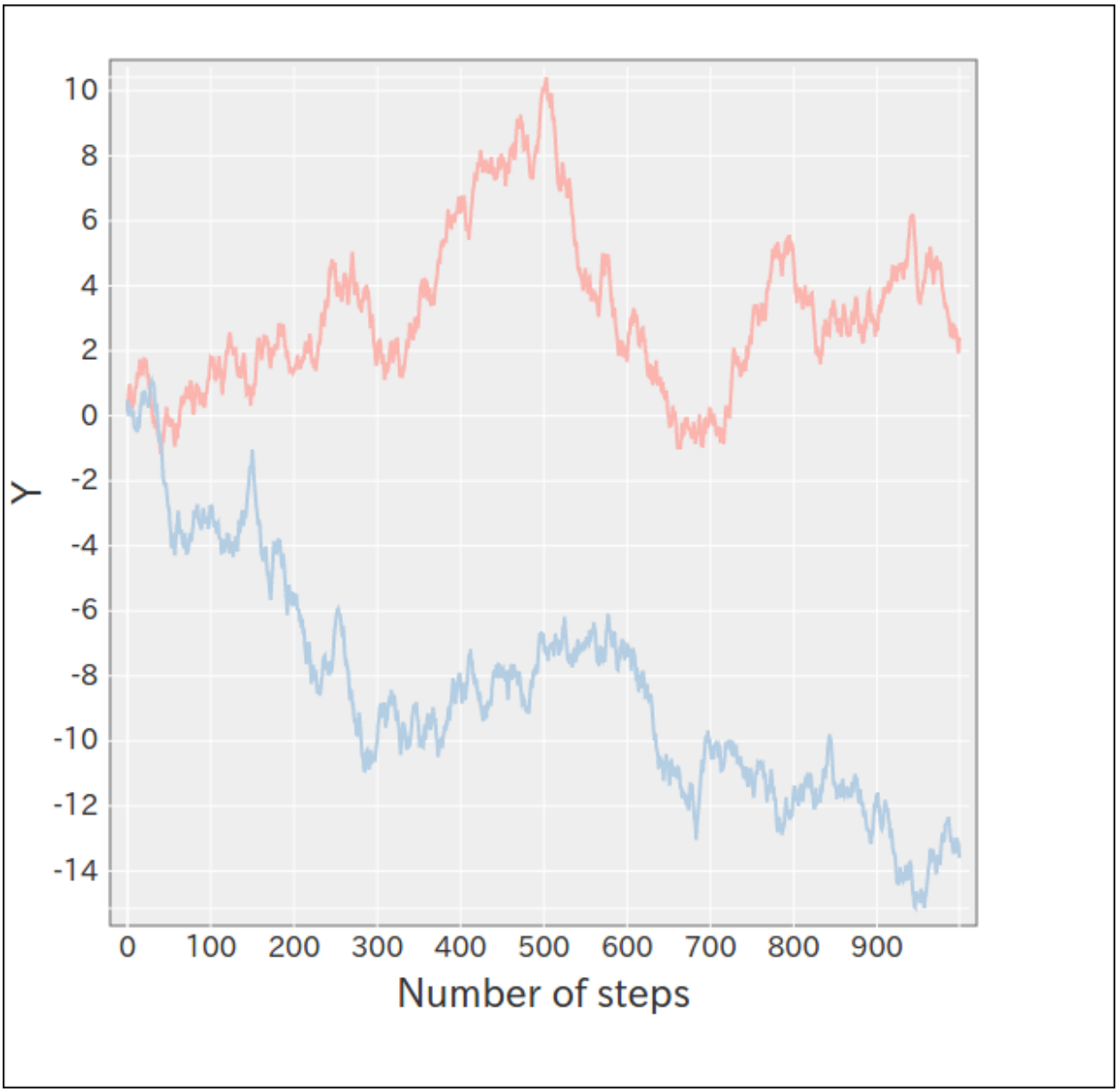
Prepare data to visualize.

```
var xarr=[], yarr1=[], yarr2=[], N=1000, curry1=0.0, curry2=0.0;
for(var i=0;i < N;i++){
  xarr.push(i);
  yarr1.push(curry1+=(Math.random()-0.5));
  yarr2.push(curry2+=(Math.random()-0.5));
}
```

Specify div element which svg will be appended to ("vis" in this example) and begin rendering.

```
var plot = new Nyaplot.Plot({xlabel: "Number of steps"});
var line1 = new Nyaplot.Line(xarr, yarr1, {color: "#fbb4ae"});
var line2 = new Nyaplot.Line(xarr, yarr2, {color: "#b3cde3"});

plot.add([line1, line2]);
plot.show("vis");
```



1.3 Try more

Read the code here to try more examples.

CHAPTER 2

API Reference

2.1 External API

2.1.1 Plot

```
new Nyaplot.Plot(options)
```

Base class for general 2D plots.

2.1.2 Charts

```
new Nyaplot.Scatter(xarr, yarr, options)
```

```
new Nyaplot.Histogram(arr, options)
```

```
new Nyaplot.Line(xarr, yarr, options)
```

2.2 Internal API

These functions are only for developers.

2.2.1 Nyaplot.core

```
Nyaplot.core.register_parser(name, args, optional_args, parser)
```

Register parser. ex: stage2d

2.2.2 Nyaplot.glyph_manager

```
Nyaplot.glyph_manager.register_glyph(name, args, optional_args, parser)
```

2.2.3 Nyaplot.sheet_manager

```
Nyaplot.sheet_manager.register_sheet(name, args, optional_args, parser)
```

CHAPTER 3

Architecture of nyaplotjs

Author: Naoki Nishida

3.1 JSON parser and core.parse

Nyaplot handles a plot as one JSON object. For example, consider to create a scatter plot using Nyaplot.

```
var inside={x: [], y: []}, outside={x: [], y: []}, N=7000;

for(var i=0;i < N;i++){
  var x= Math.random();
  var y= Math.random();
  if(Math.sqrt(x*x+y*y)<1.0){
    inside.x.push(x);
    inside.y.push(y);
  }else{
    outside.x.push(x);
    outside.y.push(y);
  }
}
var plot = new Nyaplot.Plot();
var sc1 = new Nyaplot.Scatter(inside.x, inside.y, {color: "#1b9e77"});
var sc2 = new Nyaplot.Scatter(outside.x, outside.y, {color: "#d95f02"});

plot.add([sc1, sc2]);
console.log(plot.create_models());
plot.show("vis");
```

If you execute the code JSON object shown below appears in your browser console.

```
  [
  {
    "type":"data",
    "uuid":"b98944912cf845a481175a05e19d0a41",
    "args":{
      "data":[
        {
          "x":0.0035517734941095114,
          "y":0.07813602453097701
        },
        {
          "x":0.7182991192676127,
```

```
          "y":0.06800984055735171
        },
        {
          "x":0.38948169839568436,
          "y":0.3537671882659197
        },
        {
          "x":0.21537378802895546,
          "y":0.24944813642650843
        },
        {
          "x":0.3799612079747021,
          "y":0.7313088402152061
        },
        {
          "x":0.5874782362952828,
          "y":0.16591674066148698
        },
        {
          "x":0.49753625388257205,
          "y":0.47685628780163825
        }
      ]
    }
  },
  {
    "type":"data",
    "uuid":"55049859aa44400095ed728c21f17b9d",
    "args":{
      "data":[
        {
          "x":0.8678227821364999,
          "y":0.9131462422665209
        },
        {
          "x":0.5947368117049336,
          "y":0.9414090760983527
        },
        {
          "x":0.39573086868040264,
          "y":0.9548168019391596
        }
      ]
    }
  },
  {
    "type":"scale",
    "uuid":"5c5f0d6d2c7e43f7b1979ef6dfc5a636",
    "args":{
      "type":"linear",
      "domain":[
        0.0035517734941095114,
        0.8678227821364999
      ],
      "range":[
        10,
        490
      ]
```

```
    }
  },
  {
    "type":"scale",
    "uuid":"527d57180d58494fa96017e283b422b0",
    "args":{
      "type":"linear",
      "domain":[
        0.06800984055735171,
        0.9548168019391596
      ],
      "range":[
        490,
        10
      ]
    }
  },
  {
    "type":"position2d",
    "uuid":"2f91ee4b06184529b77aae1161edcde1",
    "args":{

    },
    "sync_args":{
      "x":"5c5f0d6d2c7e43f7b1979ef6dfc5a636",
      "y":"527d57180d58494fa96017e283b422b0"
    }
  },
  {
    "type":"scatter",
    "uuid":"66bd35a2e00040d09526a7d65d5dfbbc",
    "args":{
      "x":"x",
      "y":"y",
      "color":"#1b9e77"
    },
    "sync_args":{
      "position":"2f91ee4b06184529b77aae1161edcde1",
      "data":"b98944912cf845a481175a05e19d0a41"
    }
  },
  {
    "type":"scatter",
    "uuid":"b9e68f98ff1246c8a3538a30c888e619",
    "args":{
      "x":"x",
      "y":"y",
      "color":"#d95f02"
    },
    "sync_args":{
      "position":"2f91ee4b06184529b77aae1161edcde1",
      "data":"55049859aa44400095ed728c21f17b9d"
    }
  },
  {
    "type":"axis2d",
    "uuid":"35f8a39c74dd407da92d371314f90f33",
    "args":{
```

```
      "width":500,
      "height":500
    },
    "sync_args":{
      "xscale":"5c5f0d6d2c7e43f7b1979ef6dfc5a636",
      "yscale":"527d57180d58494fa96017e283b422b0"
    }
  },
  {
    "type":"label",
    "uuid":"ac9217ea26354847be6ed73131398ed2",
    "args":{
      "x":"X",
      "y":"Y",
      "width":500,
      "height":500,
      "margin":{
        "bottom":70,
        "left":60
      }
    }
  },
  {
    "type":"background2d",
    "uuid":"f4546241916f45d4bf45b293b55cfc25",
    "args":{
      "width":500,
      "height":500
    }
  },
  {
    "type":"context2d",
    "uuid":"ca5bbc2116a546bd9b1abc76596206f0",
    "args":{
      "width":500,
      "height":500
    },
    "sync_args":{
      "glyphs":[
        "66bd35a2e00040d09526a7d65d5dfbbc",
        "b9e68f98ff1246c8a3538a30c888e619"
      ]
    }
  },
  {
    "type":"interactive_wheel",
    "uuid":"ac4370adb4eb4bedbf58bd950fa7de90",
    "args":{
      "size":[
        500,
        500
      ],
      "stage_uuid":"cc8dc98190db460c916a2b8761f345c9"
    },
    "sync_args":{
      "xscale":"5c5f0d6d2c7e43f7b1979ef6dfc5a636",
      "yscale":"527d57180d58494fa96017e283b422b0",
      "updates":[
```

```
        "35f8a39c74dd407da92d371314f90f33",
        "66bd35a2e00040d09526a7d65d5dfbbc",
        "b9e68f98ff1246c8a3538a30c888e619"
      ]
    }
  },
  {
    "type":"stage2d",
    "uuid":"cc8dc98190db460c916a2b8761f345c9",
    "args":{
      "width":700,
      "height":700,
      "margin":{
        "x":60,
        "y":10
      }
    },
    "sync_args":{
      "sheets":[
        "f4546241916f45d4bf45b293b55cfc25",
        "35f8a39c74dd407da92d371314f90f33",
        "ca5bbc2116a546bd9b1abc76596206f0",
        "ac9217ea26354847be6ed73131398ed2",
        "ac4370adb4eb4bedbf58bd950fa7de90"
      ]
    }
  },
  {
    "type":"pane",
    "uuid":"pane",
    "args":{
      "parent_id":"vis",
      "layout":{
        "type":"rows",
        "contents":[
          0
        ]
      }
    },
    "sync_args":{
      "stages":[
        "cc8dc98190db460c916a2b8761f345c9"
      ]
    }
  }
]
```

As shown above, one plots consists of arrays of hash (called *object* in JavaScript) which has keys *type*, *uuid*, *args*, and *sync_args*.

This JSON object will be passed to Nyaplot.core.parse.

Nyaplot.core.parse will pass each hash in JSON object to parsers. The parser to which *core.parse* pass hash is determined using *type* key. Consider the hash below which is in JSON object generated above.

```
{
  "type":"scatter",
  "uuid":"66bd35a2e00040d09526a7d65d5dfbbc",
  "args":{
```

```
    "x":"x",
    "y":"y",
    "color":"#1b9e77"
  },
  "sync_args":{
    "position":"2f91ee4b06184529b77aae1161edcde1",
    "data":"b98944912cf845a481175a05e19d0a41"
  }
}
```

This hash will passed to scatter parser.

Args in the key *args* as *x*, *y* and *color* will be passed to the parser as immediate values.

Ones in the key *sync_args* like *position* and *data* will be replaced by results of other parsers as position parser and data parser respectively.

3.2 General architecture of 2D plots

CHAPTER 4

Indices and tables

- genindex
- search