dyplot Documentation

Release

Tsung-Han Yang

February 25, 2016

Contents

Contents

CHAPTER 1

Motivation

The motivation of writing dyplot came to me when I plotted a 10-year price chart on a stock by matplotlib. I cannot see the exact price as the points all became too small to see. I can zoom in if I use svg, but it is diffcult to navigate. There is no way of interacting with data and chart. I searched and found dygraphs. It's easy to use, flexible and allows users to interact with the data. It can also handles large dataset.

Then I needed to convert some of my scripts to use dygraphs, and I could not find any python library to do that. That's the start of dyplot.

dygraphs does support line chart, so I found c3.js. In additional to line chart, c3.js supports pie and bar. To handle large data sets, c3.js is not as good as dygraphs.

For a while, I'm unable to find a good javascript library for scatter chart. nvd3 has a nice scatter chart.

Chart	js library
pie	c3.js
bar	c3.js
histogram	c3.js
line	dygraph
scatter	nvd3

CHAPTER 2

Introduction

We can look at the javascript example at dygraphs.com.

```
<script type="text/javascript">
g = new Dygraph(
document.getElementById("demodiv"),
        "ny-vs-sf.txt",
        {
        rollPeriod: 14,
        showRoller: true,
        customBars: true,
        title: 'NYC vs. SF',
        ylabel: 'Temperature (F)',
        legend: 'always'
        }
        );
</script>
```

For dygraph to work, we need to specify at least three things in html code.

1. *g*: the javascript variable of the Dygraph object
2. *demodiv*: The div id. dygraph will draw the chart on this part of the page.
3. *ny-vs-sf.txt*: The url path to the data file.

We can use the following python codes to generate a random array and plot it with dyplot.

```
import numpy as np
import pandas as pd
from dyplot.dygraphs import Dygraphs
if __name__ == '__main__':
    foo = np.random.rand(100)
    boo = pd.Series(foo, index=range(0,100))
    dg = Dygraphs(boo.index, "date")
    dg.plot(series="Random", mseries=boo)
    dg.set_options(title="Tutorial 0")
    div = dg.savefig(csv_file="tutorial0.csv", div_id="demodiv", js_vid="jid", html_file="tutorial0.h
```

dg.savefig return the html codes to be embedded in html and save html codes to tutorial0.html. The content of div is:

```
<div id="demodiv" style="width:1024px; height:600px;"></div>
<script type="text/javascript">
jid = new Dygraph(
    document.getElementById("demodiv"),
    "tutorial0.csv", // path to CSV file
```

```
{"legend": "always", "series": {"Random": {"axis": "y"}}, "showRoller": false, "ylabel": "Y Values",
  jid.ready(function() {
    jid.setAnnotations([]    );
  });
</script>
```

You can see the output.

CHAPTER 3

Tutorials

3.1 Plot three series

See the output at http://blacksburg98.github.io/dyplot/html/tutorial1.html

```
import pandas as pd
from dyplot.dygraphs import Dygraphs
a = pd.Series([1,2,3,4,5,6,7,9,10])
b = pd.Series([1,3,5,9,2,8,5,5,15])
lc= pd.Series([1,3,4,5,6,7,9,3,2])
c = pd.Series([2,4,5,7,8,8,9,4,3])
hc= pd.Series([3,5,7,7,9,11,9,5,8])
dg = Dygraphs(a.index, "index")
dg.plot(series="a", mseries=a)
dg.plot(series="b", mseries=b)
dg.plot(series="c", mseries=c,lseries=lc, hseries=hc)
dg.set_options(title="Test")
dg.set_axis_options(axis="x",axisLabelColor="red")
div = dg.savefig(csv_file="tutorial1.csv", html_file="tutorial1.html")
```

3.2 Pie Chart

See the output at http://blacksburg98.github.io/dyplot/html/c3_pie.html

```
from dyplot.pie import Pie
frac = [30, 20, 50]
labels = ["setosa", "versicolor", "viginica"]
g = Pie(frac=frac, labels=labels)
c = {}
c["columns"] = []
c["columns"].append(["setosa", 100])
g.animate("load", c, 1000)
g.savefig(html_file="c3_pie.html")
```

3.3 Bar Chart

See the output at http://blacksburg98.github.io/dyplot/html/c3_bar.html

```
from dyplot.bar import Bar
h = [30, 20, 50, 40]
label = "setosa"
g = Bar(height=h, label=label)
h2 = [50, 30, 20, 30]
label2 = "barora"
h3 = [40, 20, 10, 50]
label3 = "exama"
g = Bar(height=h, label=label)
g(height=h2, label=label2)
g(height=h3, label=label3)
g.set_xticklabels(["G1", "G2", "G3", "G4"], "category")
g.savefig(html_file="c3_bar.html")
```

3.4 Histogram

See the output at http://blacksburg98.github.io/dyplot/html/c3_hist.html

```
from dyplot.hist import Hist
a = [3,4,4,5,6,6,6,6,5,9,3,3,4,4,5,6,7,8,9,3,2,2,4,5]
g = Hist(a, xlabel="Frequency")
g.savefig(html_file="c3_hist.html")
```

3.5 Scatter Plot

See the output at http://blacksburg98.github.io/dyplot/html/nvd3_scatter.html

```
import numpy as np
import csv
from dyplot.scatter import Scatter
if __name__ == '__main__':
    x = np.random.rand(20)
    y = np.random.rand(20)
    s = Scatter(x, y, "Group 1", s=2, slope=-1, intercept=0.9)
    x = np.random.rand(10)
    y = np.random.rand(10)
    s(x, y, "Group 2", marker="+", s=[1,3,2,4,5,6,6,5,4,3], slope=1, intercept=0.1)
    x = np.random.rand(10)
    y = np.random.rand(10)
    s(x, y, "Group 3", s=2, marker=["^", 'v', 'o', '+', 's', 's', 's', 'D', 'D', 'v'])
    s.savefig(csv_file="../gh-pages/html/nvd3_scatter.csv",html_file="../gh-pages/html/nvd3_scatter.h
        ,csv_url="nvd3_scatter.csv")
```

CHAPTER 4

dyplot package

4.1 Submodules

4.2 dyplot.dygraphs module

class dyplot.dygraphs.**Dygraphs**(*x*, *xname*)

matplotlib-like plot functions for dygraphs.js and c3.js. Dygraph.py makes it easier to integrate dygraphs.js into webpage. Pleas see dygaphs.com for detail.

__init__(*x*, *xname*)

Initialization

Parameters

- **x** (*index list from pandas.*) – x-axis.
- **xname** (*string*) – x-axis name.

__module__ = 'dyplot.dygraphs'

annotate(*series*, *x*, *shortText*, *text=''*)

To annotate a particular point in the plot.

Parameters

- **series** – series name
- **x** – the x coordinate for the annotation.
- **shortText** – Short Text to display on the plot.
- **text** – the text shown when the mouse is on top of the short text.

auto_range(*axis='y'*)

To automatically adjust vertical range

Parameters **axis** (*string*) – "y" or "y2"

candleplot(*open*, *close*, *high*, *low*, ***kwargs*)

Candle sticks plot function for stock analysis. All four arguments, open, close, high and low, are mandatory.

Parameters

- **open** (*pandas series*) – open price series
- **close** (*pandas series*) – close price series

- **high** (*pandas series*) – high price series
- **low** (*pandas series*) – low price series

plot (*series*, *mseries*, *lseries=None*, *hseries=None*, ***kwargs*)
the function to plot the series. If lseries is specified, dygraphs will shade the space between the main series and the low series. If hseries is specified, dygraphs will shade the space between the main series and the high series.

Parameters

- **series** (*string*) – Series name
- **mseries** (*pandas series.*) – The main series.
- **lseries** (*pandas series.*) – The low series.
- **hseries** (*pandas series.*) – The high series.

save_csv (*csv_file*, *dt_fmt*)
To save all necessary data to a csv file.

param csv_file csv file name

type csv_file string

param dt_fmt date time format if x-axis is date-time

type dt_fmt string

save_html (*html_file*, *div*)
To save the plot to a html file.

savefig (*csv_file='dyplot.csv'*, *div_id='dyplot'*, *js_vid='g'*, *dt_fmt='%Y-%m-%d'*, *html_file=None*, *width='1024px'*, *height='600px'*, *csv_url=''*)
To save the plot to a html file or to return html code.

Parameters

- **csv_file** (*string*) – the csv file name
- **csv_url** (*string*) – the csv url used by javascript
- **div_id** (*string*) – div id to be used in html
- **js_vid** (*string*) – id to be used in javascript
- **dt_fmt** (*string*) – date-time format if the seriers are time series.
- **html_file** (*string*) – the file name of html file
- **width** (*int*) – The width of the plot
- **height** (*int*) – The height of the plot

Return div the html code to be embedded in the webpage is return.

set_axis_options (*axis*, ***kwargs*)
To set the option of an axis. Please find the options on dygraphs.com.

Parameters **axis** (*string*) – "x", "y" or "y2"

For example, to set the label of x axis of g to red.

```
g.set_axis_options(axis="x",axisLabelColor="red")
```

set_options (***kwargs*)
To set global option.

set_series_options(*series*, ***kwargs*)
To set the option of a series. Please find the options on dygraphs.com.

Parameters **series** (*string*) – series name

4.3 dyplot.bar module

class dyplot.bar.**Bar**(*height*, *label*)
Bases: dyplot.c3.C3

__call__(*height*, *label*)

__init__(*height*, *label*)
To plot a bar chart.

Parameters

- **height** (*array_like*) – The data to plot. A list of data.
- **label** (*string*) – The name of the data.

__module__ = 'dyplot.bar'

4.4 dyplot.hist module

class dyplot.hist.**Hist**(*a*, *bins=10*, *r=None*, *weights=None*, *density=None*, *xlabel=''*, ***kwargs*)
Bases: dyplot.c3.C3

Compute the histogram of a set of data.

__init__(*a*, *bins=10*, *r=None*, *weights=None*, *density=None*, *xlabel=''*, ***kwargs*)

Parameters

- **a** (*array_like*) – input data. The histogram is computed over the flattened array.
- **bins** (*(float, float), optional*) – The lower and upper range of the bins. If not provided, range is simply (a.min(), a.max()). Values outside the range are ignored.
- **weights** (*bool, optional*) – An array of weights, of the same shape as a. Each value in a only contributes its associated weight towards the bin count (instead of 1). If normed is True, the weights are normalized, so that the integral of the density over the range remains 1
- **density** (*bool, optional*) – If False, the result will contain the number of samples in each bin. If True, the result is the value of the probability density function at the bin, normalized such that the integral over the range is 1. Note that the sum of the histogram values will not be equal to 1 unless bins of unity width are chosen; it is not a probability mass function. Overrides the normed keyword if given.
- **xlabel** (*string*) – The name of the x label.

__module__ = 'dyplot.hist'

4.5 dyplot.pie module

class dyplot.pie.**Pie**(*frac*, *labels*)
Bases: dyplot.c3.C3

__init__(*frac*, *labels*)
To plot a pie chart.

Parameters

- **frac** (*array_like. A list of float or int.*) – The list to plot.
- **labels** (*array_like. A list of string.*) – The list of the slice names.

For example, the following would give a pie chart with three slices: 30%, 20% and 50%.

```
from dyplot.pie import Pie
frac = [30, 20, 50]
labels = ["setosa", "versicolor", "viginica"]
g = Pie(frac=frac, labels=labels)
```

__module__ = 'dyplot.pie'

4.6 dyplot.scatter module

class dyplot.scatter.**Scatter**(*x*, *y*, *g*, *s=1*, *marker='o'*, *slope=''*, *intercept=''*, *option={}*)
Bases: dyplot.nvd3.NVD3

__call__(*x*, *y*, *g*, *s=1*, *marker='o'*, *slope=''*, *intercept=''*)

__init__(*x*, *y*, *g*, *s=1*, *marker='o'*, *slope=''*, *intercept=''*, *option={}*)
To plot a scatter chart. If slope and intercept are defined, then a line will be drawn.

param x, y x, y coordinate

type x, y array_like.

param s The sizes of each data point.

type s int or a list of int

param marker

- 'o' : circle
- '+' : cross
- '^' : triangle-up
- 'v' : triangle-down
- 'D' : diamond
- 's' : square

type marker string or a list of string

param g The group name.

type g string

param slope The slope of the line.

type slope float

param intercept The Y intercept of the line

type intercept float

__module__ = 'dyplot.scatter'

m2s = {'D': 'diamond', '+': 'cross', 'o': 'circle', 's': 'square', 'v': 'triangle-down', '^': 'triangle-up'}

save_csv(*csv_file*)

savefig(*csv_file='nvd3.csv'*, *div_id='nvd3'*, *html_file=None*, *width='600px'*, *height='400px'*, *csv_url=''*)

To save the plot to a html file or to return html code.

Parameters

- **csv_file** (*string*) – the csv file name
- **csv_url** (*string*) – the csv url used by javascript
- **div_id** (*string*) – div id to be used in html
- **js_vid** (*string*) – id to be used in javascript
- **html_file** (*string*) – the file name of html file
- **width** (*string*) – The width of the plot. For exampe, "400px".
- **height** (*string*) – The height of the plot. For example, "300px".

Return div the html code to be embedded in the webpage is return.

4.7 Module contents

dyplot (c) 2014 Tsung-Han Yang This source code is released under the Apache license. Created on July 1, 2014

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

d

Symbols

A

B

C

D

H

M

P

S