django-nvd3 Documentation

Release 0.9.7

Arezqui Belaid

Nov 17, 2017

Contents

Release 0.9.7

Date Nov 17, 2017

Keywords django, python, plot, graph, nvd3, d3

Author Arezqui Belaid

Description Django wrapper for nvd3, build re-usable charts and chart components for d3.js

Contents:

CHAPTER 1

Introduction

Version 0.9.7

Date Nov 17, 2017

Keywords django, python, plot, graph, nvd3, d3

Author Arezqui Belaid

License MIT

Description Python wrapper for nvd3, build re-usable charts and chart components for d3.js

NVD3 NVD3 http://nvd3.org/

– Django-nvd3 is a Django wrapper for NVD3 graph library. NVD3 is an attempt to build re-usable charts and chart components for d3.js without taking away the power that d3.js gives you.

1.1 Overview

Django-nvd3 is a Django wrapper for NVD3 graph library. Visit NVD3 website for futher information : http://nvd3.org/

1.1.1 Installation

Install, upgrade and uninstall django-nvd3.py with these commands:

```
$ pip install django-nvd3
$ pip install --upgrade django-nvd3
$ pip uninstall django-nvd3
```

1.1.2 Dependencies

Django-nvd3 have one major dependencie:

- python-nvd3 : https://github.com/areski/python-nvd3

Bower will be used to install D3 and NvD3, see bower website for futher info : http://bower.io/

Bower depends on Node and npm. It's installed globally using npm:

```
npm install -g bower
```

To easy the integration with Django we will advice you to use django-bower.

For instance to run our demo project, you will install the dependencies from requirements.txt and then install django-bower. Django-bower is not a mandatory dependencies as the user should be free to install JS files using different method.

To install django-bower:

```
$ pip install django-bower
```

Read the documentation about Django-bower to find out how to configure it properly for your project: https://github.com/nvbn/django-bower

Then in the demo project directory just type the following:

```
$ python manage.py bower_install
$ python manage.py collectstatic
```

This will create a directory "components" where d3 & nvd3 will be installed.

You can see example settings file in demoproject.

1.1.3 Example how to create a pieChart

Let's say we have a simple view in which we want to display the amount of calories per fruit.

So to achieve this, we will edit our view.py, we will prepare the data that will be displayed:

```
xdata = ["Apple", "Apricot", "Avocado", "Banana", "Boysenberries", "Blueberries",
↪"Dates", "Grapefruit", "Kiwi", "Lemon"]
ydata = [52, 48, 160, 94, 75, 71, 490, 82, 46, 17]
chartdata = {'x': xdata, 'y': ydata}
charttype = "pieChart"
chartcontainer = 'piechart_container'
data = {
    'charttype': charttype,
    'chartdata': chartdata,
    'chartcontainer': chartcontainer,
    'extra': {
        'x_is_date': False,
        'x_axis_format': '',
        'tag_script_js': True,
        'jquery_on_ready': False,
    }
}
return render_to_response('piechart.html', data)
```

We will render the template 'piechart.html' with a dictionary 'data' which contains 'charttype' and 'chartdata'. 'extra' will contains a list of additional settings:

```
* ``x_is_date`` - if enabled the x-axis will be display as date format
* ``x_axis_format`` - set the x-axis date format, ie. "%d %b %Y"
* ``tag_script_js`` - if enabled it will add the javascript tag '<script>'
* ``jquery_on_ready`` - if enabled it will load the javascript only when page is␣
↪loaded
    this will use jquery library, so make sure to add jquery to the template.
* ``color_category`` - Define color category (eg. category10, category20, category20c)
```

Our template piechart.html could look like this:

```
{% load nvd3_tags %}
<head>
    {% include_chart_jscss %}
    {% load_chart charttype chartdata chartcontainer extra %}
</head>
<body>
    <h1>Fruits vs Calories</h1>
    {% include_container chartcontainer 400 600 %}
</body>
```

We use include the Javascript and CSS code for D3/NVD3. We start preparing and display the javascript code needed to render our pieChart:

```
{% load_chart charttype chartdata "piechart_container" extra %}
```

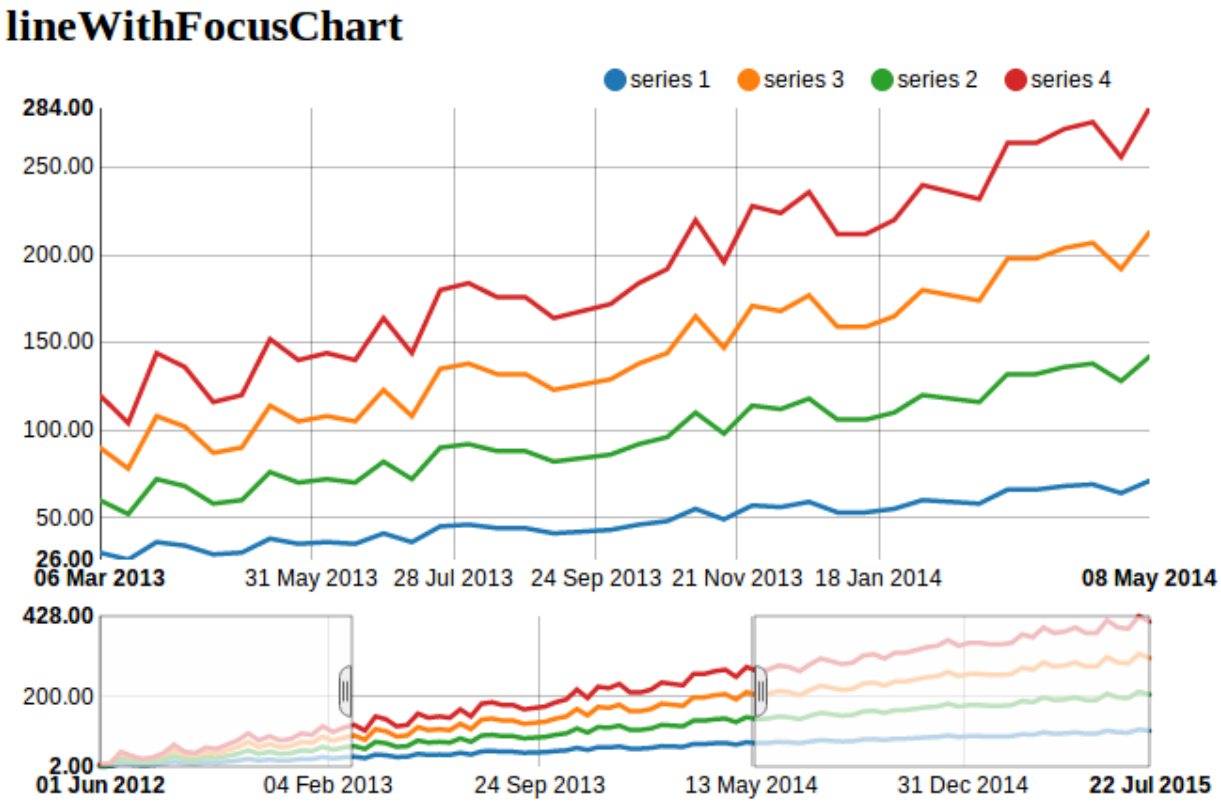
Finally we create a div container which will be used to display the chart.

See the demo project in directory 'demoproject' for examples of django-nvd3 usage.

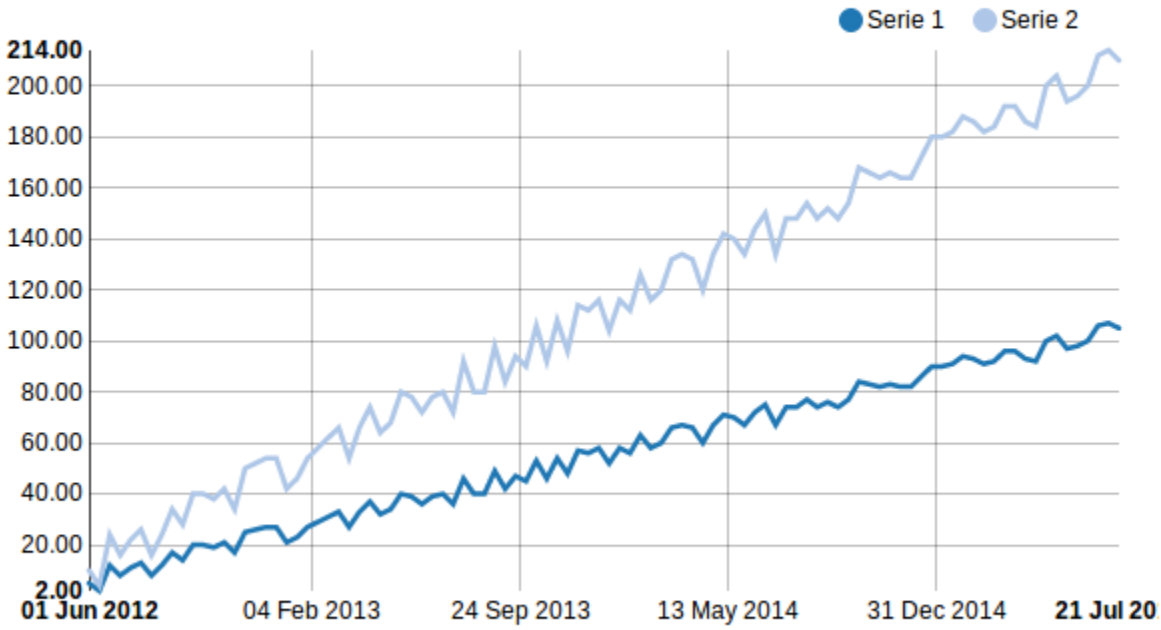
1.1.4 Live demo of NVD3

See a live demo on jsfiddle : http://jsfiddle.net/areski/z4zuH/246/

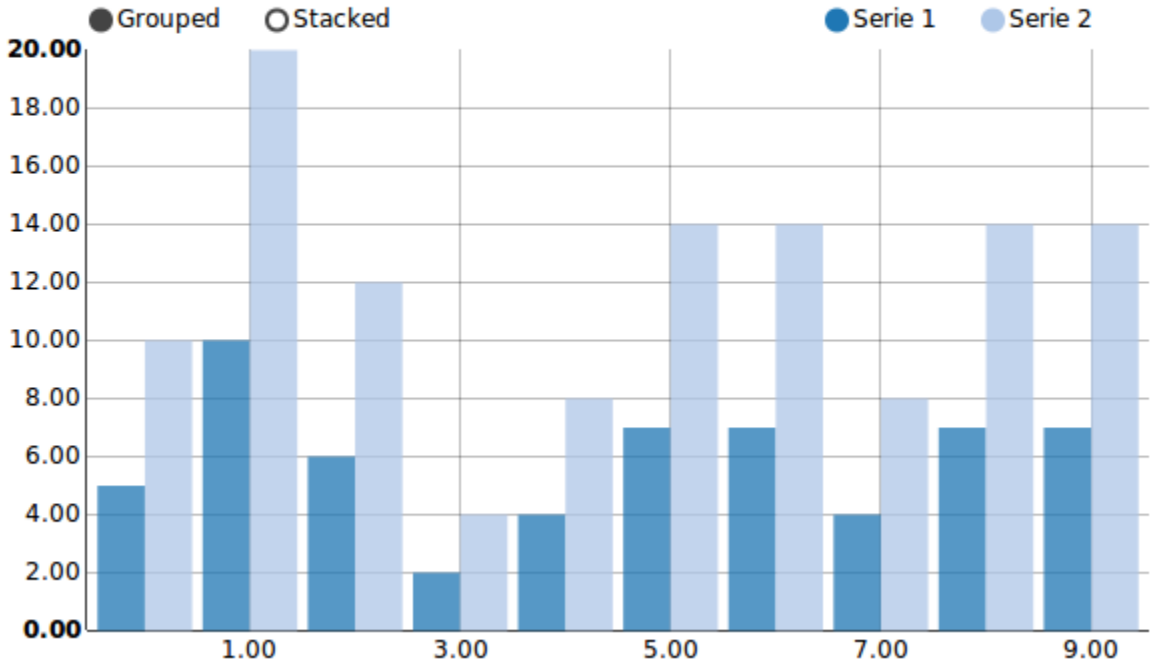
1.1.5 Supported nvd3 charts



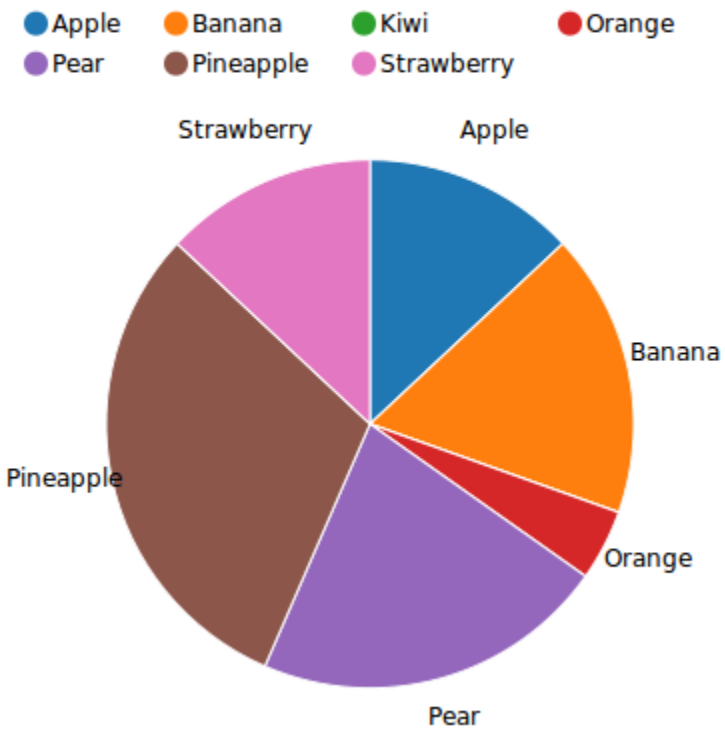
lineChart


Serie 1
Serie 2
214.00
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
2.00
01 Jun 2012
04 Feb 2013
24 Sep 2013
13 May 2014
31 Dec 2014
21 Jul 20

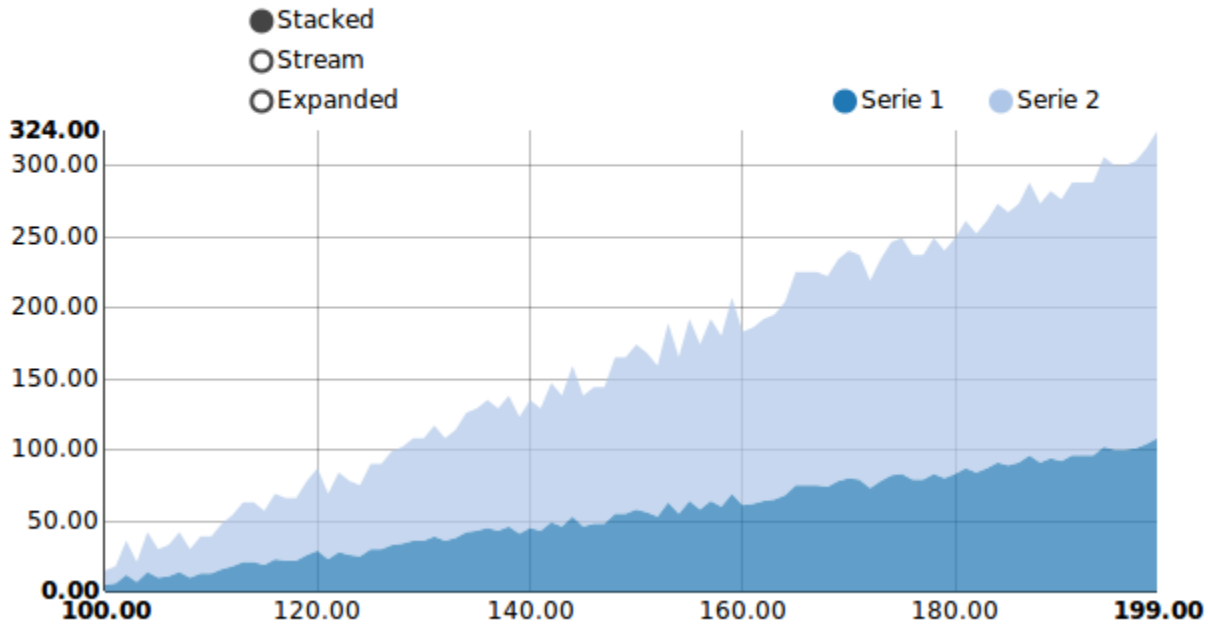
MultiBarChart



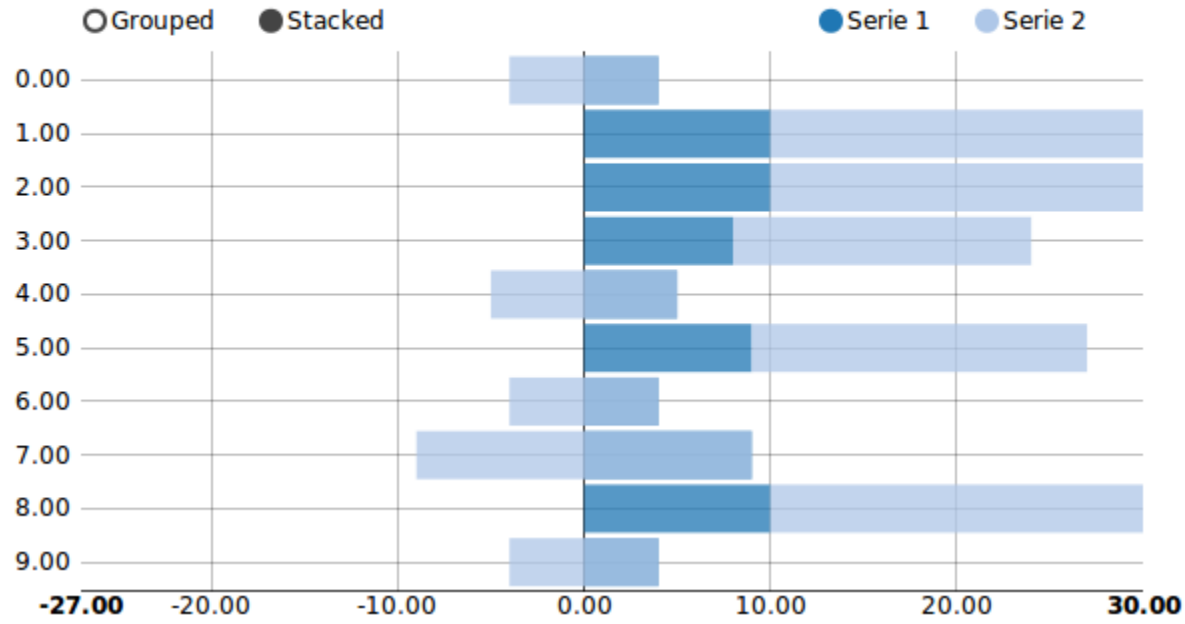
pieChart


Apple
Banana
Kiwi
Orange
Pear
Pineapple
Strawberry
Strawberry
Apple
Banana
Pineapple
Orange
Pear

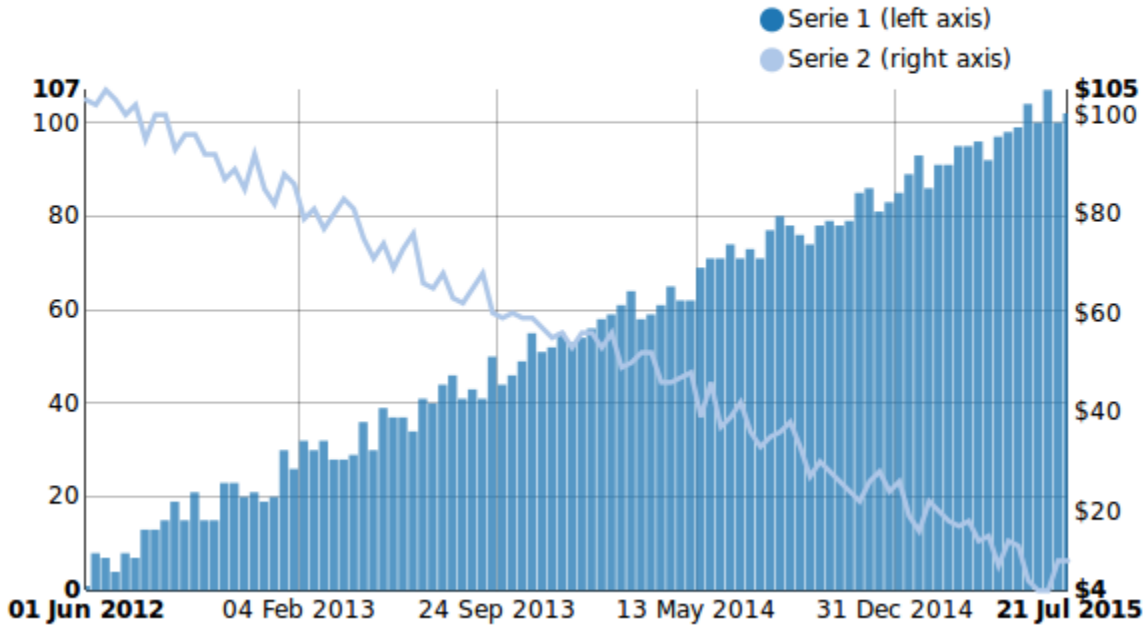
stackedAreaChart



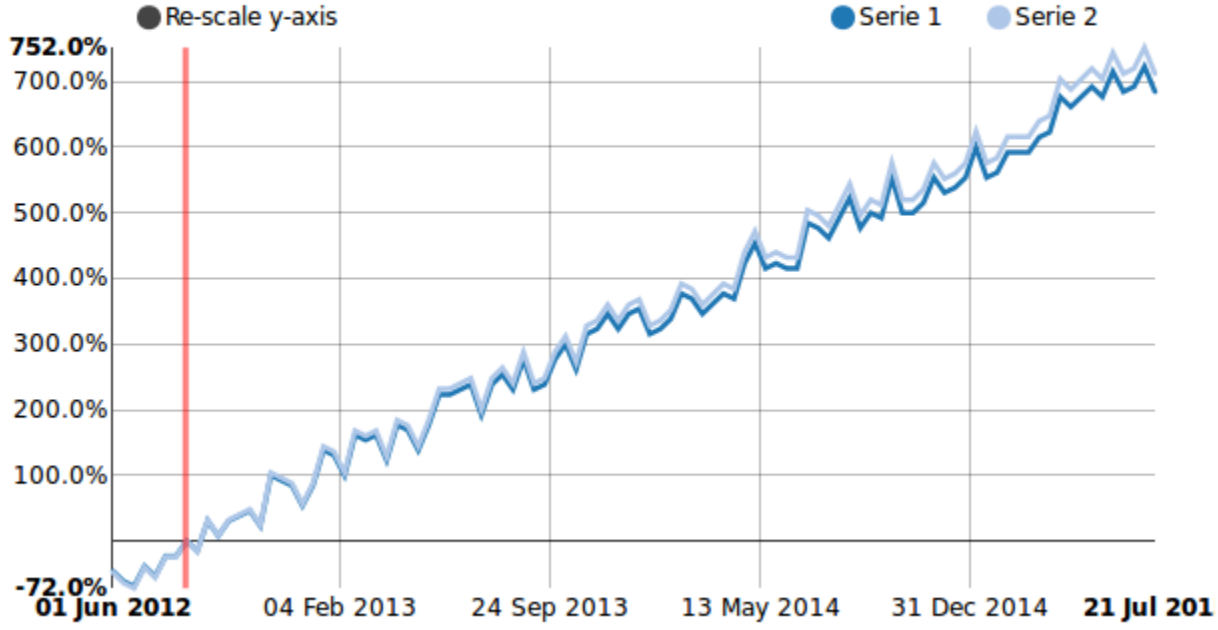
multiBarHorizontalChart



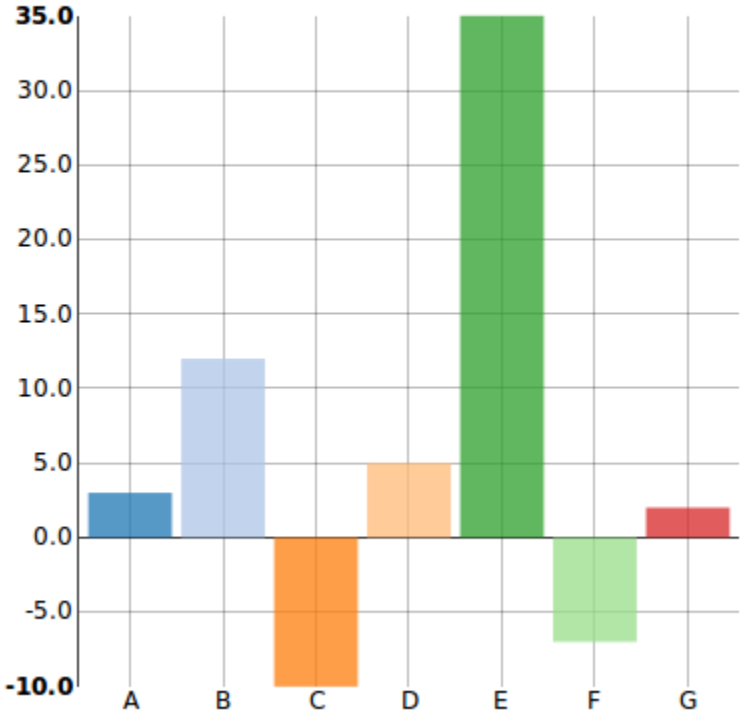
linePlusBarChart

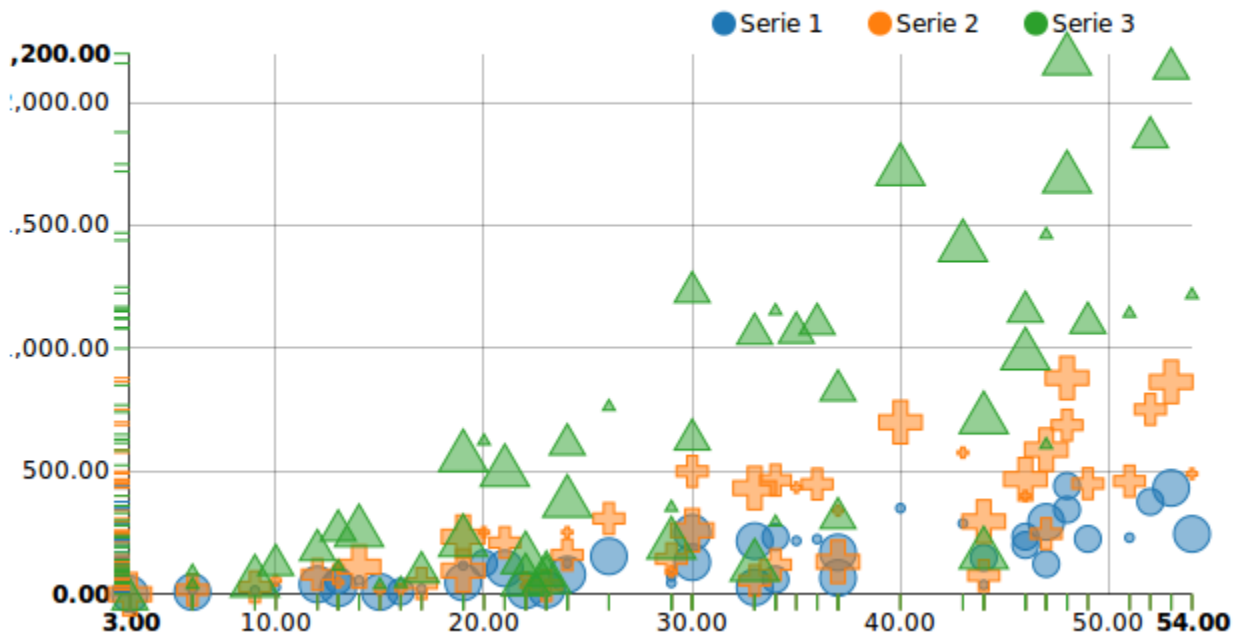


cumulativeLineChart



discreteBarChart





1.1.6 Projects using Django-nvd3

- CDR-Stats : http://www.cdr-stats.org
- Newfies-Dialer : http://www.newfies-dialer.org

1.1.7 Documentation

Documentation is available on 'Read the Docs': http://django-nvd3.readthedocs.org

1.1.8 Changelog

Changelog summary : https://github.com/areski/django-nvd3/blob/master/CHANGELOG.rst

1.2 Contributing

If you've found a bug, add a feature or improve django-nvd3 and think it is useful then please consider contributing. Patches, pull requests or just suggestions are always welcome!

Source code: http://github.com/areski/django-nvd3

If you don't like Github and Git you're welcome to send regular patches.

Bug tracker: http://github.com/areski/django-nvd3/issues

1.3 License

CHAPTER 2

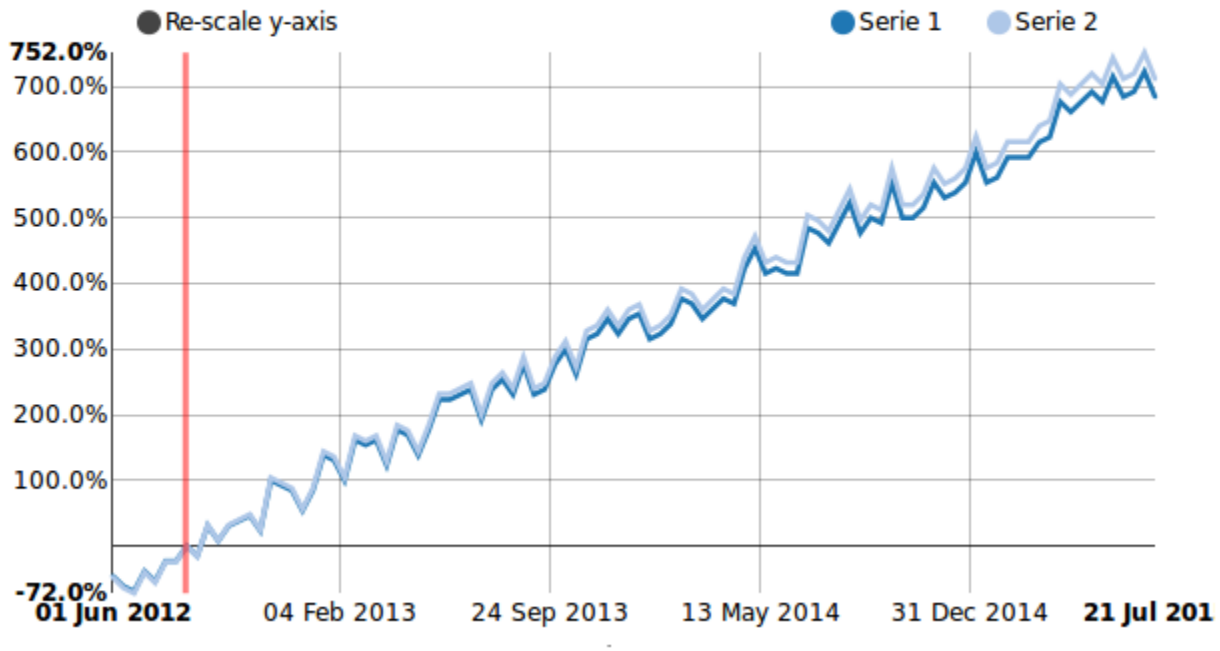
Chart Classes Reference

Contents:

2.1 cumulativeLineChart

A cumulative line chart is used when you have one important grouping representing an ordered set of data and one value to show, summed over time.

cumulativeLineChart



Django example:

```
from django.shortcuts import render_to_response
import random
import datetime
import time

def demo_cumulativelinechart(request):
    """
    cumulativelinechart page
    """
    start_time = int(time.mktime(datetime.datetime(2012, 6, 1).timetuple()) * 1000)
    nb_element = 100
    xdata = range(nb_element)
    xdata = map(lambda x: start_time + x * 1000000000, xdata)
    ydata = [i + random.randint(1, 10) for i in range(nb_element)]
    ydata2 = map(lambda x: x * 2, ydata)

    tooltip_date = "%d %b %Y %H:%M:%S %p"
    extra_serie1 = {"tooltip": {"y_start": "", "y_end": " calls"},
                    "date_format": tooltip_date}
    extra_serie2 = {"tooltip": {"y_start": "", "y_end": " min"},
                    "date_format": tooltip_date}

    chartdata = {
        'x': xdata,
        'name1': 'series 1', 'y1': ydata, 'extra1': extra_serie1,
        'name2': 'series 2', 'y2': ydata2, 'extra2': extra_serie2,
    }
```

```
    charttype = "cumulativeLineChart"
    data = {
        'charttype': charttype,
        'chartdata': chartdata,
    }
    return render_to_response('cumulativelinechart.html', data)
```

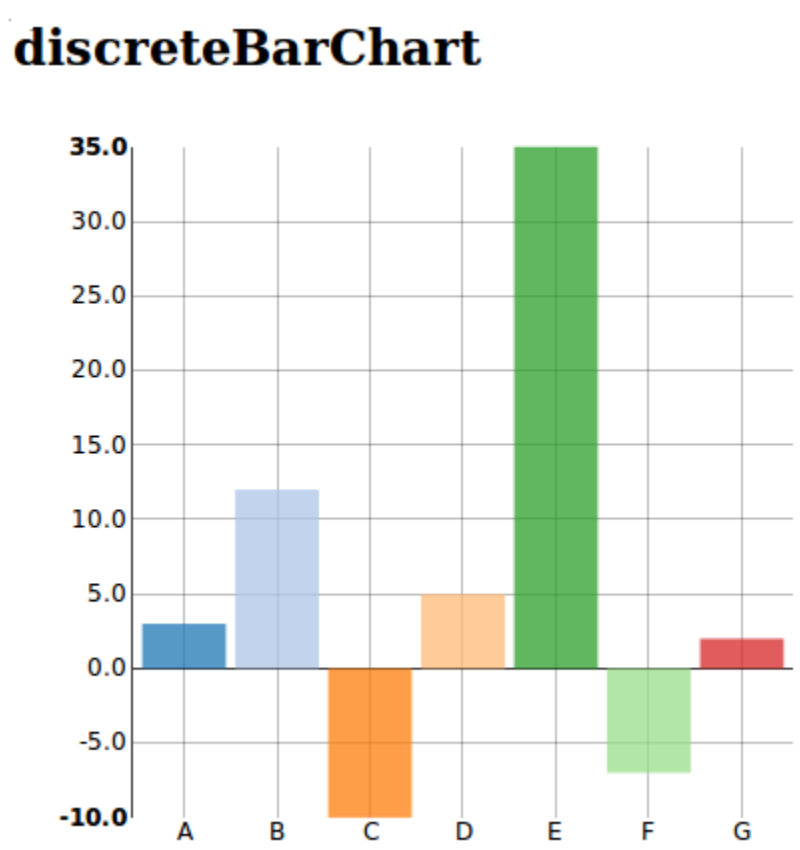
Template example:

```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "cumulativelinechart_container" True "%d %b %Y
↪%H" %}
</head>
<body>
    {% include_container "cumulativelinechart_container" 400 600 %}
</body>
```

2.2 discreteBarChart

A discrete bar chart or bar graph is a chart with rectangular bars with lengths proportional to the values that they represent.



Django example:

```
from django.shortcuts import render_to_response
import random
import datetime
import time

def demo_discretebarchart(request):
"""
discretebarchart page
"""
xdata = ["A", "B", "C", "D", "E", "F", "G"]
ydata = [3, 12, -10, 5, 35, -7, 2]

extra_serie1 = {"tooltip": {"y_start": "", "y_end": " cal"}}
chartdata = {
    'x': xdata, 'name1': '', 'y1': ydata, 'extra1': extra_serie1,
}
charttype = "discreteBarChart"
data = {
    'charttype': charttype,
    'chartdata': chartdata,
}
return render_to_response('discretebarchart.html', data)
```

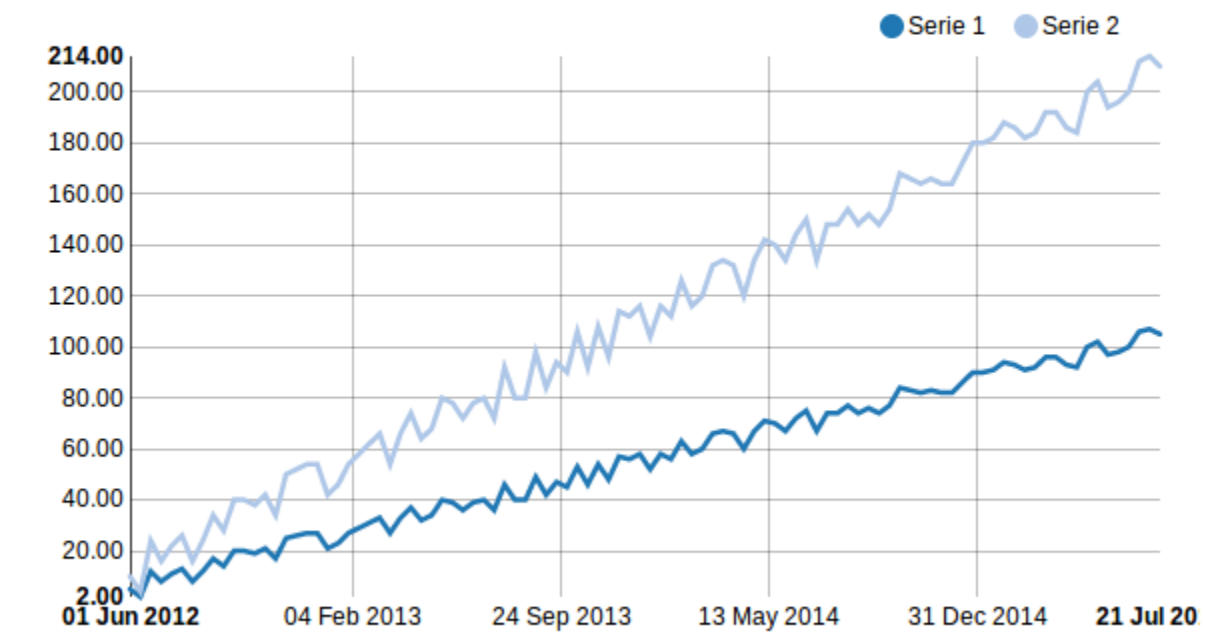
Template example:

```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "discretebarchart_container" %}
</head>
<body>
    {% include_container "discretebarchart_container" 400 600 %}
</body>
```

2.3 lineChart

A line chart or line graph is a type of chart which displays information as a series of data points connected by straight line segments.



Django example:

```
from django.shortcuts import render_to_response
import random
import datetime
```

```
import time

def demo_linechart(request):
    """
    lineChart page
    """
    start_time = int(time.mktime(datetime.datetime(2012, 6, 1).timetuple()) * 1000)
    nb_element = 100
    xdata = range(nb_element)
    xdata = map(lambda x: start_time + x * 1000000000, xdata)
    ydata = [i + random.randint(1, 10) for i in range(nb_element)]
    ydata2 = map(lambda x: x * 2, ydata)

    tooltip_date = "%d %b %Y %H:%M:%S %p"
    extra_serie = {"tooltip": {"y_start": "", "y_end": " cal"},
                   "date_format": tooltip_date}
    chartdata = {'x': xdata,
                 'name1': 'series 1', 'y1': ydata, 'extra1': extra_serie,
                 'name2': 'series 2', 'y2': ydata2, 'extra2': extra_serie}
    charttype = "lineChart"
    data = {
        'charttype': charttype,
        'chartdata': chartdata
    }
    return render_to_response('linechart.html', data)
```

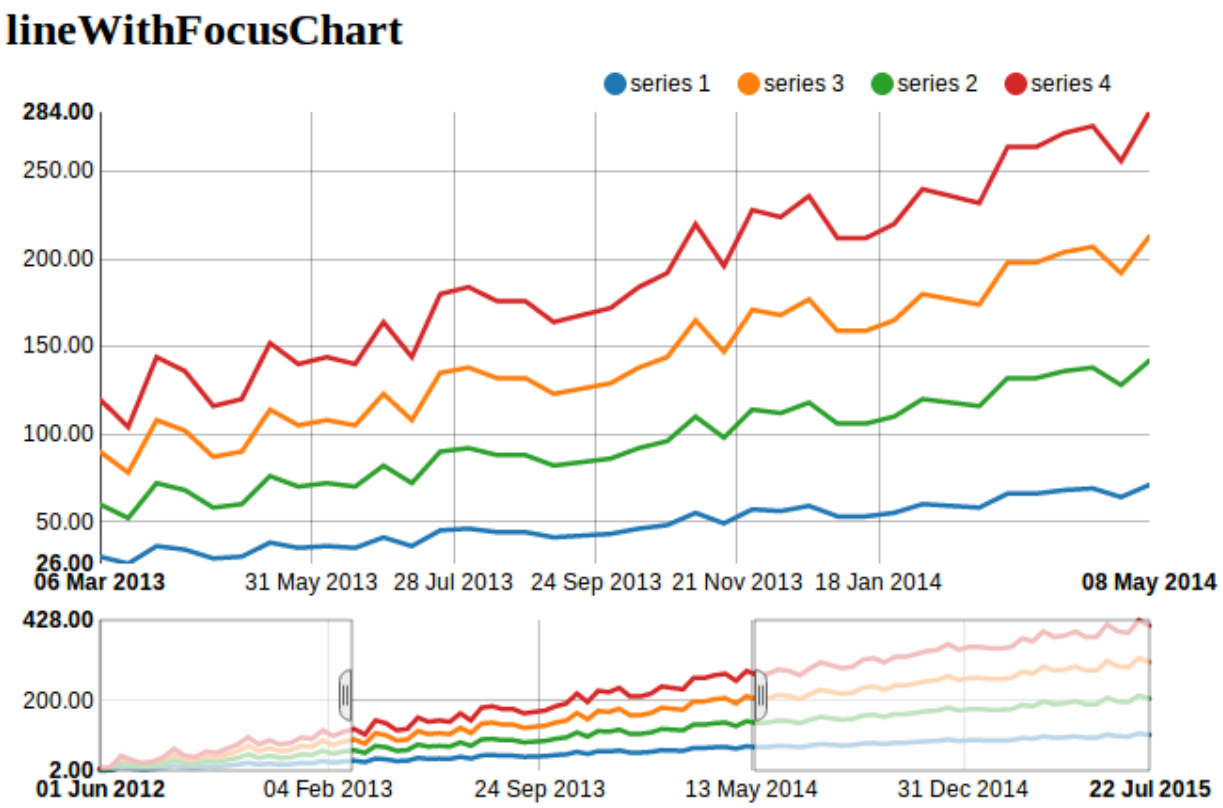
Template example:

```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "linechart_container" True "%d %b %Y %H" %}
</head>
<body>
    {% include_container "linechart_container" 400 600 %}
</body>
```

2.4 lineWithFocusChart

A lineWithFocusChart or line graph is a type of chart which displays information as a series of data points connected by straight line segments. The lineWithFocusChart provide a smaller chart that act as a selector, this is very useful if you want to zoom on a specific time period.



Django example:

```
from django.shortcuts import render_to_response
import random
import datetime
import time

def demo_linewithfocuschart(request):
    """
    linewithfocuschart page
    """
    nb_element = 100
    start_time = int(time.mktime(datetime.datetime(2012, 6, 1).timetuple()) * 1000)

    xdata = range(nb_element)
    xdata = map(lambda x: start_time + x * 1000000000, xdata)
    ydata = [i + random.randint(1, 10) for i in range(nb_element)]
    ydata2 = map(lambda x: x * 2, ydata)
    ydata3 = map(lambda x: x * 3, ydata)
    ydata4 = map(lambda x: x * 4, ydata)

    tooltip_date = "%d %b %Y %H:%M:%S %p"
    extra_serie = {"tooltip": {"y_start": "There are ", "y_end": " calls"},
                   "date_format": tooltip_date}
    chartdata = {
        'x': xdata,
        'name1': 'series 1', 'y1': ydata, 'extra1': extra_serie,
        'name2': 'series 2', 'y2': ydata2, 'extra2': extra_serie,
        'name3': 'series 3', 'y3': ydata3, 'extra3': extra_serie,
```

```
        'name4': 'series 4', 'y4': ydata4, 'extra4': extra_serie
    }
    charttype = "lineWithFocusChart"
    data = {
        'charttype': charttype,
        'chartdata': chartdata
    }
    return render_to_response('linewithfocuschart.html', data)
```

Template example:

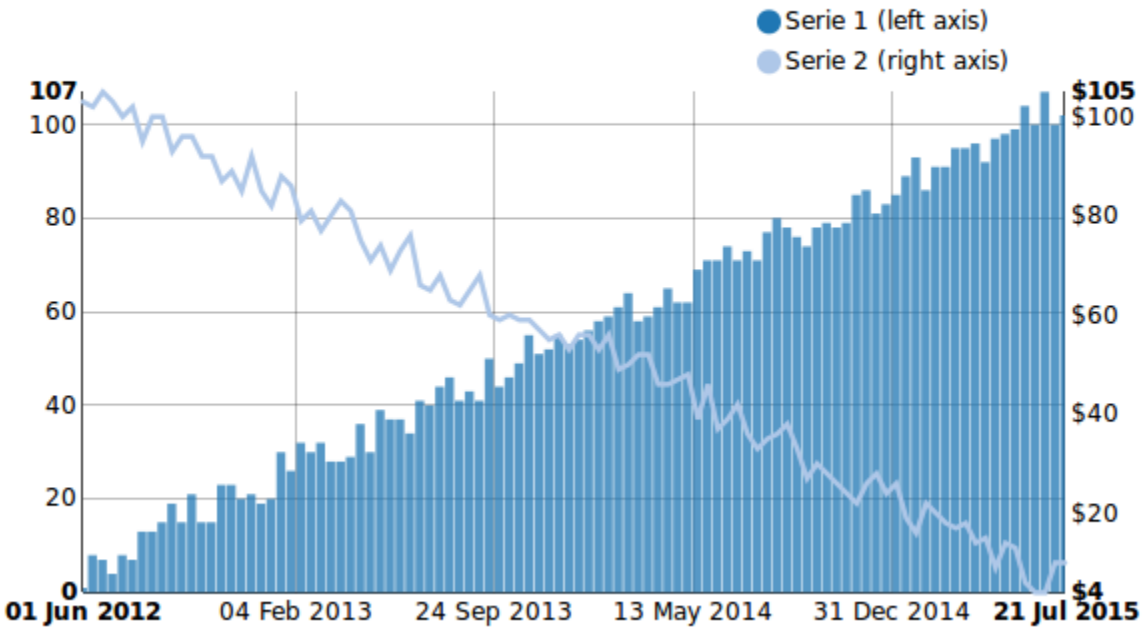
```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "linewithfocuschart_container" True "%d %b %Y %H
↪" %}
</head>
<body>
    {% include_container "linewithfocuschart_container" 400 '100%' %}
</body>
```

2.5 linePlusBarChart

A linePlusBarChart Chart is a type of chart which displays information as a series of data points connected by straight line segments and with some series with rectangular bars with lengths proportional to the values that they represent

linePlusBarChart



Django example:

```
from django.shortcuts import render_to_response
import random
import datetime
import time

def demo_lineplusbarchart(request):
    """
    lineplusbarchart page
    """
    start_time = int(time.mktime(datetime.datetime(2012, 6, 1).timetuple()) * 1000)
    nb_element = 100
    xdata = range(nb_element)
    xdata = map(lambda x: start_time + x * 1000000000, xdata)
    ydata = [i + random.randint(1, 10) for i in range(nb_element)]
    ydata2 = [i + random.randint(1, 10) for i in reversed(range(nb_element))]

    kwargs1 = {}
    kwargs1['bar'] = True

    tooltip_date = "%d %b %Y %H:%M:%S %p"
    extra_serie1 = {"tooltip": {"y_start": "$ ", "y_end": ""},
                    "date_format": tooltip_date}
    extra_serie2 = {"tooltip": {"y_start": "", "y_end": " min"},
                    "date_format": tooltip_date}

    chartdata = {
        'x': xdata,
```

```
        'name1': 'series 1', 'y1': ydata, 'extra1': extra_serie1, 'kwargs1': kwargs1,
        'name2': 'series 2', 'y2': ydata2, 'extra2': extra_serie2,
    }

    charttype = "linePlusBarChart"
    data = {
        'charttype': charttype,
        'chartdata': chartdata,
        'extra': {
            'focus_enable': True,
        },
    }
    return render_to_response('lineplusbarchart.html', data)
```

Template example:

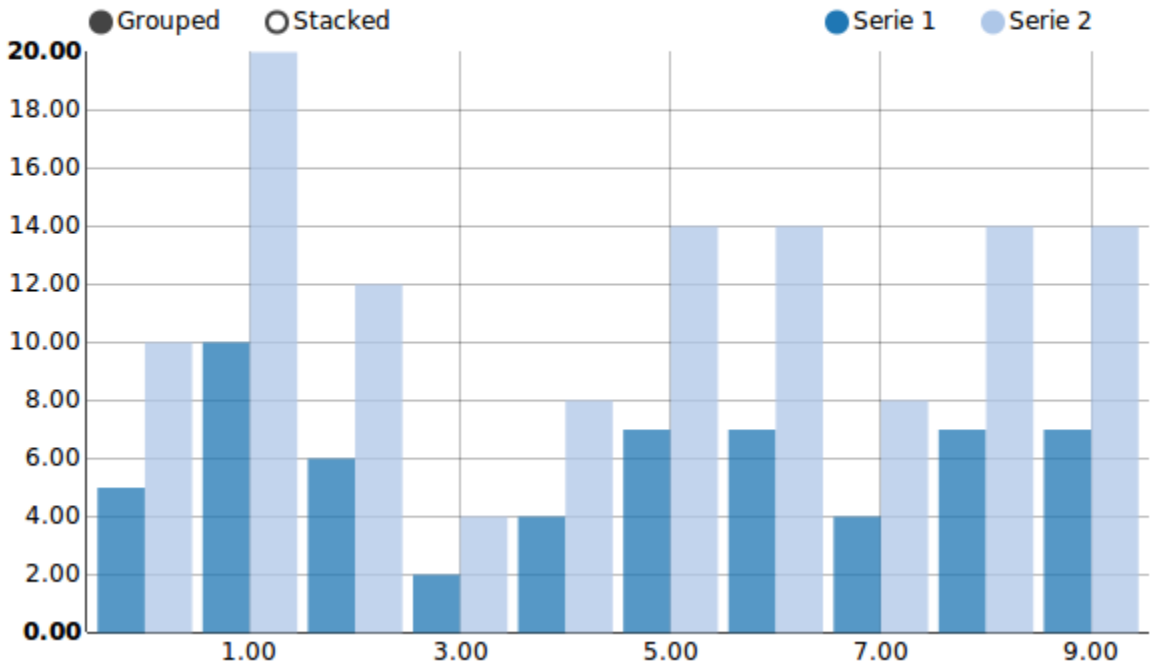
```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "lineplusbarchart_container" True "%d %b %Y %H"
↪%}
</head>
<body>
    {% include_container "lineplusbarchart_container" 400 600 %}
</body>
</body>
```

2.6 multiBarChart

A multiple bar graph contains comparisons of two or more categories or bars. One axis represents a quantity and the other axis identifies a specific feature about the categories. Reading a multiple bar graph includes looking at extremes (tallest/longest vs. shortest) in each grouping.

MultiBarChart



Django example:

```
from django.shortcuts import render_to_response
import random
import datetime
import time

def demo_multibarchart(request):
    """
    multibarchart page
    """
    nb_element = 10
    xdata = range(nb_element)
    ydata = [random.randint(1, 10) for i in range(nb_element)]
    ydata2 = map(lambda x: x * 2, ydata)
    ydata3 = map(lambda x: x * 3, ydata)
    ydata4 = map(lambda x: x * 4, ydata)

    extra_serie = {"tooltip": {"y_start": "There are ", "y_end": " calls"}}

    chartdata = {
        'x': xdata,
        'name1': 'series 1', 'y1': ydata, 'extra1': extra_serie,
        'name2': 'series 2', 'y2': ydata2, 'extra2': extra_serie,
        'name3': 'series 3', 'y3': ydata3, 'extra3': extra_serie,
        'name4': 'series 4', 'y4': ydata4, 'extra4': extra_serie
    }
```

```
    charttype = "multiBarChart"
    data = {
        'charttype': charttype,
        'chartdata': chartdata
    }
    return render_to_response('multibarchart.html', data)
```

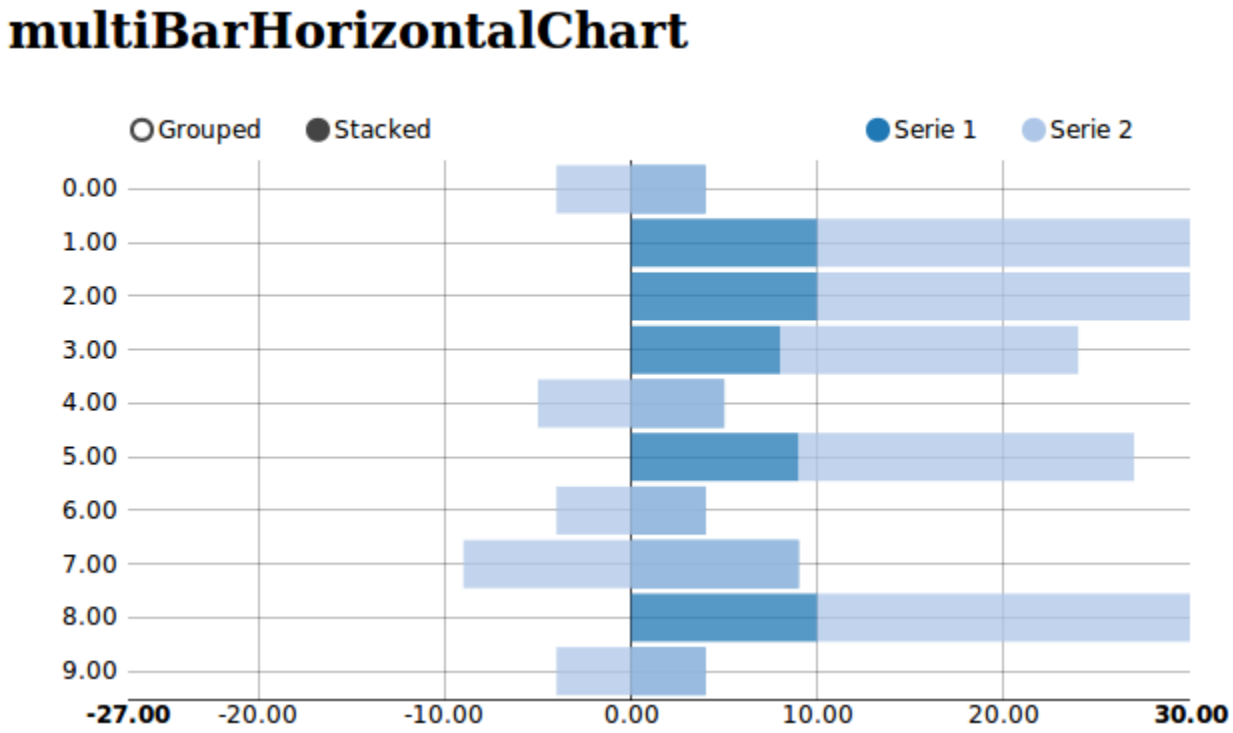
Template example:

```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "multibarchart_container" %}
</head>
<body>
    {% include_container "multibarchart_container" 400 600 %}
</body>
```

2.7 multiBarHorizontalChart

A multiple horizontal bar graph contains comparisons of two or more categories or bars.



Django example:

```
from django.shortcuts import render_to_response
import random
import datetime
import time

def demo_multibarhorizontalchart(request):
    """
    multibarhorizontalchart page
    """
    nb_element = 10
    xdata = range(nb_element)
    ydata = [i + random.randint(-10, 10) for i in range(nb_element)]
    ydata2 = map(lambda x: x * 2, ydata)

    extra_serie = {"tooltip": {"y_start": "", "y_end": " mins"}}

    chartdata = {
        'x': xdata,
        'name1': 'series 1', 'y1': ydata, 'extra1': extra_serie,
        'name2': 'series 2', 'y2': ydata2, 'extra2': extra_serie,
    }

    charttype = "multiBarHorizontalChart"
    data = {
        'charttype': charttype,
        'chartdata': chartdata
    }
    return render_to_response('multibarhorizontalchart.html', data)
```

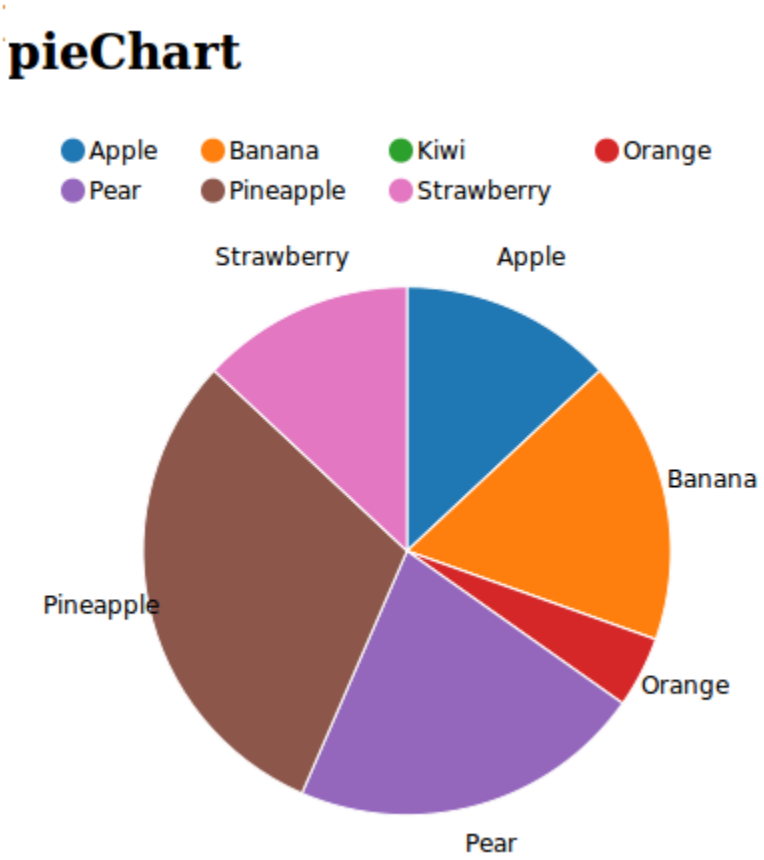
Template example:

```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "multibarhorizontalchart_container" %}
</head>
<body>
    {% include_container "multibarhorizontalchart_container" 400 600 %}
</body>
```

2.8 pieChart

A pie chart (or a circle graph) is a circular chart divided into sectors, illustrating numerical proportion. In chart, the arc length of each sector is proportional to the quantity it represents.



Django example:

```
from django.shortcuts import render_to_response
import random
import datetime
import time

def demo_piechart(request):
    """
    pieChart page
    """
    xdata = ["Apple", "Apricot", "Avocado", "Banana", "Boysenberries", "Blueberries",
↪"Dates", "Grapefruit", "Kiwi", "Lemon"]
    ydata = [52, 48, 160, 94, 75, 71, 490, 82, 46, 17]

    extra_serie = {"tooltip": {"y_start": "", "y_end": " cal"}}
    chartdata = {'x': xdata, 'y1': ydata, 'extra1': extra_serie}
    charttype = "pieChart"

    data = {
        'charttype': charttype,
        'chartdata': chartdata,
    }
    return render_to_response('piechart.html', data)
```

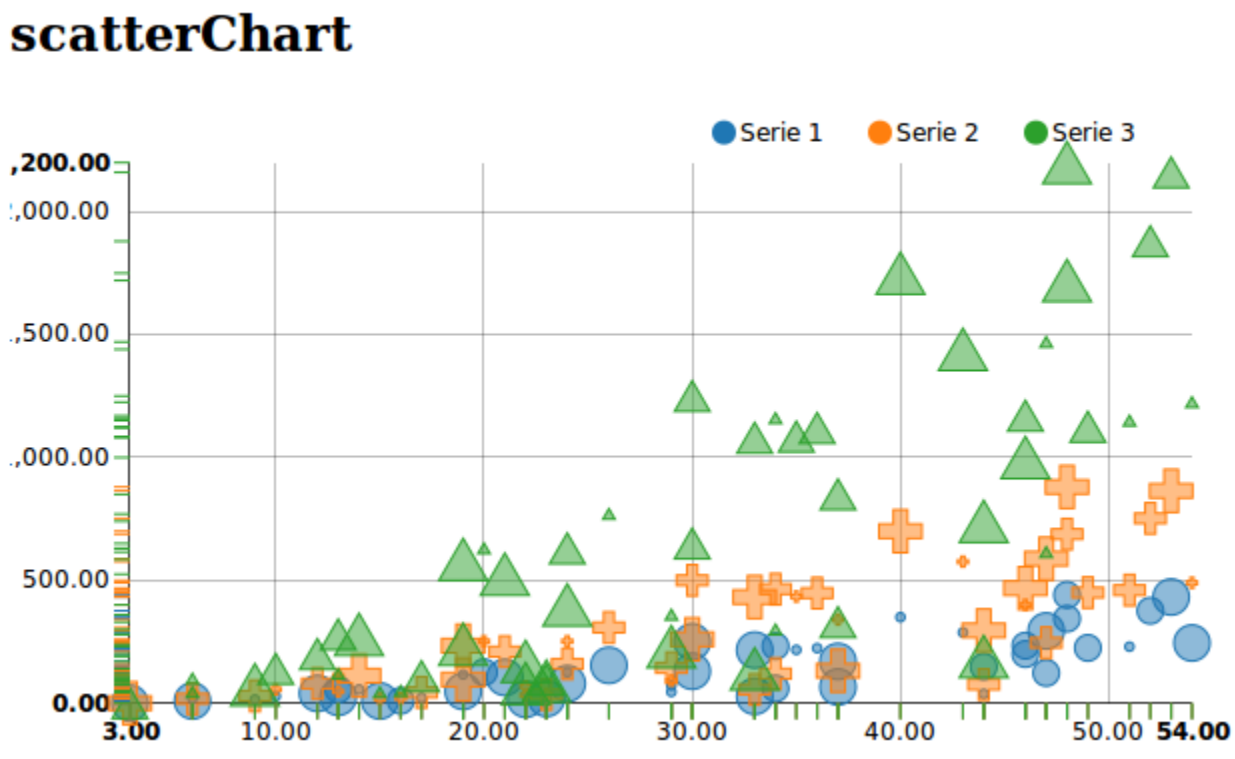
Template example:

```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "piechart_container" %}
</head>
<body>
    <h1>Fruits vs Calories</h1>
    {% include_container "piechart_container" 400 500 %}
</body>
```

2.9 scatterChart

A scatter plot or scattergraph is a type of mathematical diagram using Cartesian coordinates to display values for two variables for a set of data. The data is displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.



Django example:

```
from django.shortcuts import render_to_response
import random
```

```
import datetime
import time

def demo_scatterchart(request):
    """
    scatterchart page
    """
    nb_element = 50
    xdata = [i + random.randint(1, 10) for i in range(nb_element)]
    ydata1 = [i * random.randint(1, 10) for i in range(nb_element)]
    ydata2 = map(lambda x: x * 2, ydata1)
    ydata3 = map(lambda x: x * 5, ydata1)

    kwargs1 = {'shape': 'circle'}
    kwargs2 = {'shape': 'cross'}
    kwargs3 = {'shape': 'triangle-up'}

    extra_serie1 = {"tooltip": {"y_start": "", "y_end": " balls"}}

    chartdata = {
        'x': xdata,
        'name1': 'series 1', 'y1': ydata1, 'kwargs1': kwargs1, 'extra1': extra_serie1,
        'name2': 'series 2', 'y2': ydata2, 'kwargs2': kwargs2, 'extra2': extra_serie1,
        'name3': 'series 3', 'y3': ydata3, 'kwargs3': kwargs3, 'extra3': extra_serie1
    }
    charttype = "scatterChart"
    data = {
        'charttype': charttype,
        'chartdata': chartdata,
    }
    return render_to_response('scatterchart.html', data)
```

Template example:

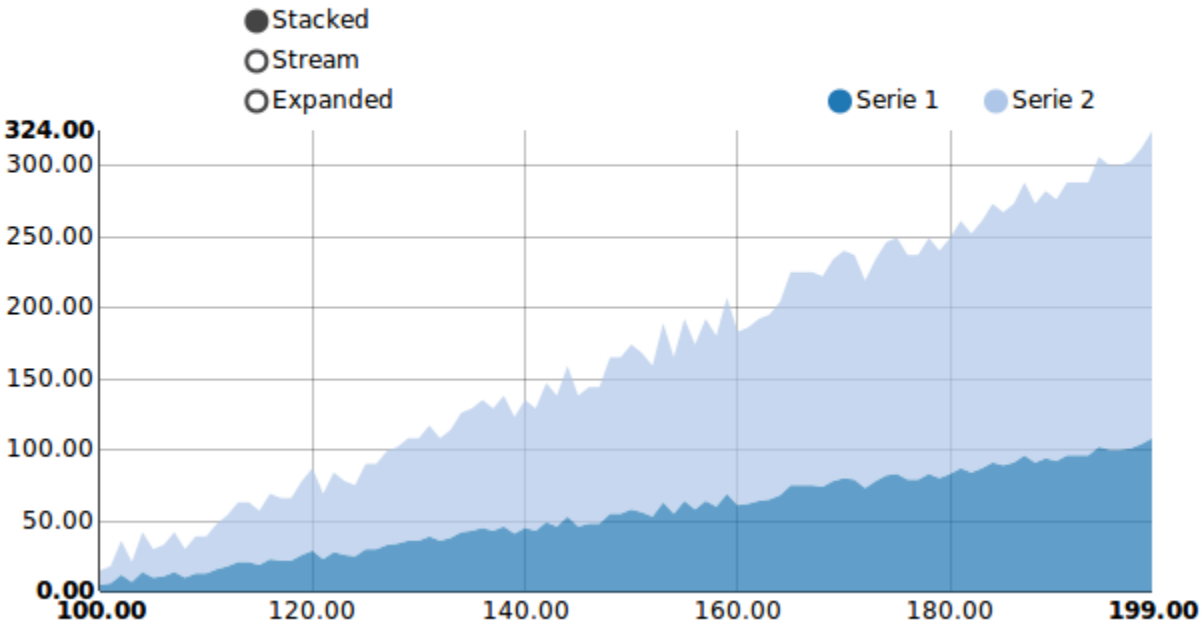
```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "scatterchart_container" %}
</head>
<body>
    {% include_container "scatterchart_container" 400 600 %}
</body>
```

2.10 stackedAreaChart

The stacked area chart is identical to the area chart, except the areas are stacked on top of each other, rather than overlapping. This can make the chart much easier to read.

stackedAreaChart



Django example:

```
from django.shortcuts import render_to_response
import random
import datetime
import time

def demo_stackedareachart(request):
    """
    stackedareachart page
    """
    nb_element = 100
    xdata = range(nb_element)
    xdata = map(lambda x: 100 + x, xdata)
    ydata = [i + random.randint(1, 10) for i in range(nb_element)]
    ydata2 = map(lambda x: x * 2, ydata)

    extra_serie1 = {"tooltip": {"y_start": "", "y_end": " balls"}}
    extra_serie2 = {"tooltip": {"y_start": "", "y_end": " calls"}}

    chartdata = {
        'x': xdata,
        'name1': 'series 1', 'y1': ydata, 'extra1': extra_serie1,
        'name2': 'series 2', 'y2': ydata2, 'extra2': extra_serie2,
    }
    charttype = "stackedAreaChart"
    data = {
        'charttype': charttype,
        'chartdata': chartdata
```

```
    }
    return render_to_response('stackedareachart.html', data)
```

Template example:

```
{% load static %}
<link media="all" href="{% static 'nvd3/src/nv.d3.css' %}" type="text/css" rel=
↪"stylesheet" />
<script type="text/javascript" src='{% static 'd3/d3.min.js' %}'></script>
<script type="text/javascript" src='{% static 'nvd3/nv.d3.min.js' %}'></script>

{% load nvd3_tags %}
<head>
    {% load_chart charttype chartdata "stackedareachart_container" %}
</head>
<body>
    {% include_container "stackedareachart_container" 400 600 %}
</body>
```

CHAPTER 3

Resources

3.1 Feedback

Your feedback is more than welcome. Write email to areski@gmail.com or post bugs and feature requests on github:
http://github.com/areski/django-nvd3/issues

3.2 Source download

The source code is currently available on github. Fork away!
http://github.com/areski/django-nvd3

CHAPTER 4

Indices and tables

- genindex
- modindex
- search