pyexcel-matplotlib Documentation

Release 0.0.1

Onni Software Ltd.

Jul 09, 2017

Contents

1 Introduction **3**

2 Installation **5**

3 Content **7**

3.1 Plot pyexcel data in Jupyter Notebook 7

3.2 Save pyexcel data as svg chart 9

Author C.W.

Source code http://github.com/pyexcel/pyexcel-matplotlib.git

Issues http://github.com/pyexcel/pyexcel-matplotlib/issues

License New BSD License

Released 0.0.1

Generated Jul 09, 2017

CHAPTER 1

Introduction

pyexcel-matplotlib is a tiny plugin of pyexcel to turn pyexcel data into chart graphics using pygal.

To see pyexcel-matplotlib in action with Jupyter notebook, please checkout the resource on github folder

CHAPTER 2

Installation

You can install it via pip:

```
$ pip install pyexcel-matplotlib
```

or clone it and install it:

```
$ git clone https://github.com/pyexcel/pyexcel-matplotlib.git
$ cd pyexcel-matplotlib
$ python setup.py install
```

CHAPTER 3

Content

Plot pyexcel data in Jupyter Notebook

There are currently four type of data layouts for rendering charts.

1 Simple Layout

Series names are placed in the first row. The rest of the rows are data sets.

Pie chart

```
title = 'Browser usage in February 2012 (in %)'
sheet = pyexcel.get_sheet(file_name='pie.csv')
svg = sheet.plot(chart_type='pie', title=title)
```

Box chart

Here is the source code using pyexcel:

```
title = 'V8 benchmark results'
sheet = pyexcel.get_sheet(file_name='box.csv')
svg = sheet.plot(chart_type='box',
     title=title, width=600, height=400, explicit_size=True)
```

2 Complex layout

On top of previous layout, x labels were inserted as the first column. In other words, each column represents series data and the first column contains x labels. y labels locate in the first row

Line

Here is the source code using pyexcel:

```
title = 'Browser usage evolution (in %)'
sheet = pyexcel.get_sheet(file_name='line.csv')
svg = sheet.plot(chart_type='line',
    title=title, width=600, height=400, explicit_size=True)
```

Dot chart

Here is the source code using pyexcel:

```
title = 'V8 benchmark results'
sheet = pyexcel.get_sheet(file_name='radar.csv')
svg = sheet.plot(chart_type='dot',
    title=title, width=600, height=400, explicit_size=True)
```

Funnel chart

Here is the source code using pyexcel:

```
title = 'V8 benchmark results'
sheet = pyexcel.get_sheet(file_name='funnel.csv')
svg = sheet.plot(chart_type='funnel',
    title=title, width=600, height=400, explicit_size=True)
```

Radar chart

Here is the source code using pyexcel:

```
title = 'V8 benchmark results'
sheet = pyexcel.get_sheet(file_name='radar.csv')
svg = sheet.plot(chart_type='radar',
    title=title, width=600, height=400, explicit_size=True)
```

Histogram

To draw a histogram, heights, starts and stops should be placed sequentially in first, second and third columns.

Here is the source code using pyexcel:

```
sheet = pyexcel.get_sheet(file_name='histogram_single.csv')
svg = sheet.plot(chart_type='histogram')
```

In order to draw multiple histogram on the same chart, you will need to use a Book, each sheet of which become a histogram. Here is how you can draw multiple histogram.

Here is the source code using pyexcel

```
sheet = pyexcel.get_sheet(file_name='histogram.csv')
svg = sheet.plot(chart_type='histogram', alpha=0.5)
```

XY

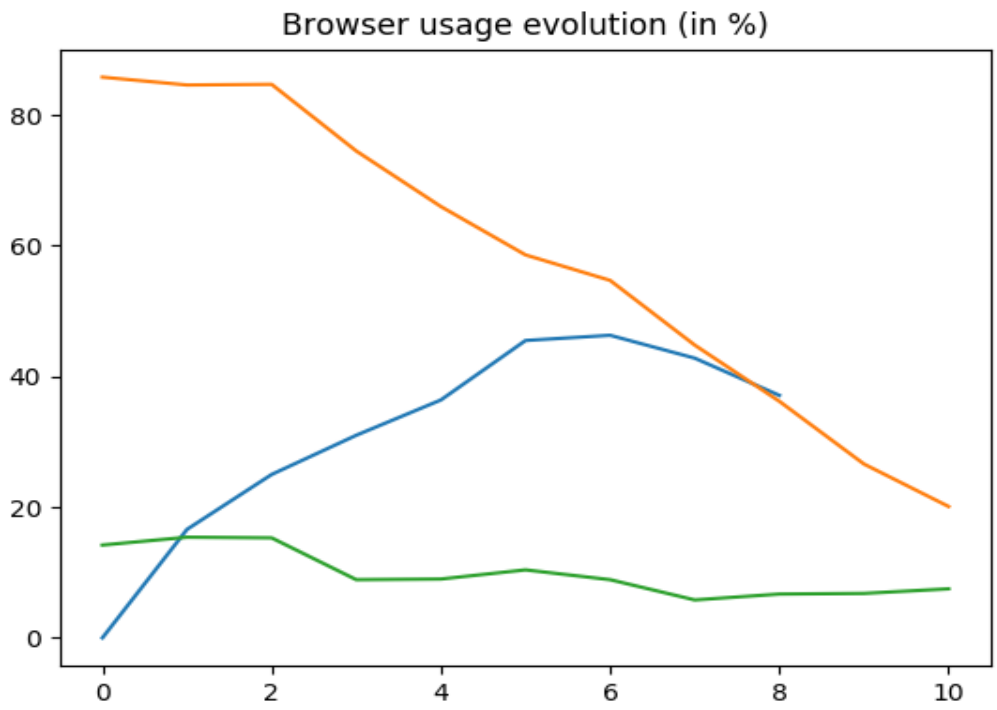
In order to draw XY graph, x, y data should be placed vertically at first and second column. In order to draw multiple lines, their data should be placed in individual sheets.

Here is the source code using pyexcel

```
book = pyexcel.get_book(file_name='xy.xlsx')
svg = book.plot(chart_type='xy',
     width=600, height=400, explicit_size=True)
```

Save pyexcel data as svg chart

Line

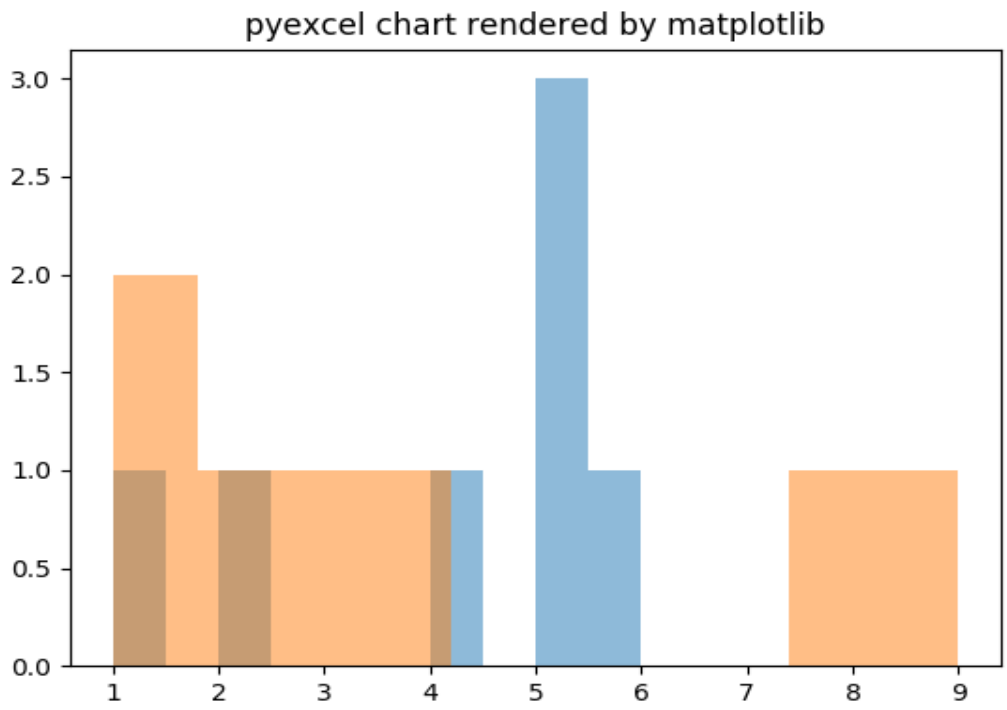


Here is the source code using pyexcel:

```
title = 'Browser usage evolution (in %)'
x_labels = map(str, range(2002, 2013))
data = {
    'Firefox': [None, None, 0, 16.6, 25, 31, 36.4, 45.5, 46.3, 42.8, 37.1],
    'Chrome': [None, None, None, None, None, None, 0, 3.9, 10.8, 23.8, 35.3],  #␣
↪flake8: noqa
    'IE': [85.8, 84.6, 84.7, 74.5, 66, 58.6, 54.7, 44.8, 36.2, 26.6, 20.1],
    'Others': [14.2, 15.4, 15.3, 8.9, 9, 10.4, 8.9, 5.8, 6.7, 6.8, 7.5]
}
pe.save_as(
```

```
    adict=data,
    dest_title=title,
    dest_x_labels=x_labels,
    dest_chart_type='line',
    dest_file_name='line.png',
)
```

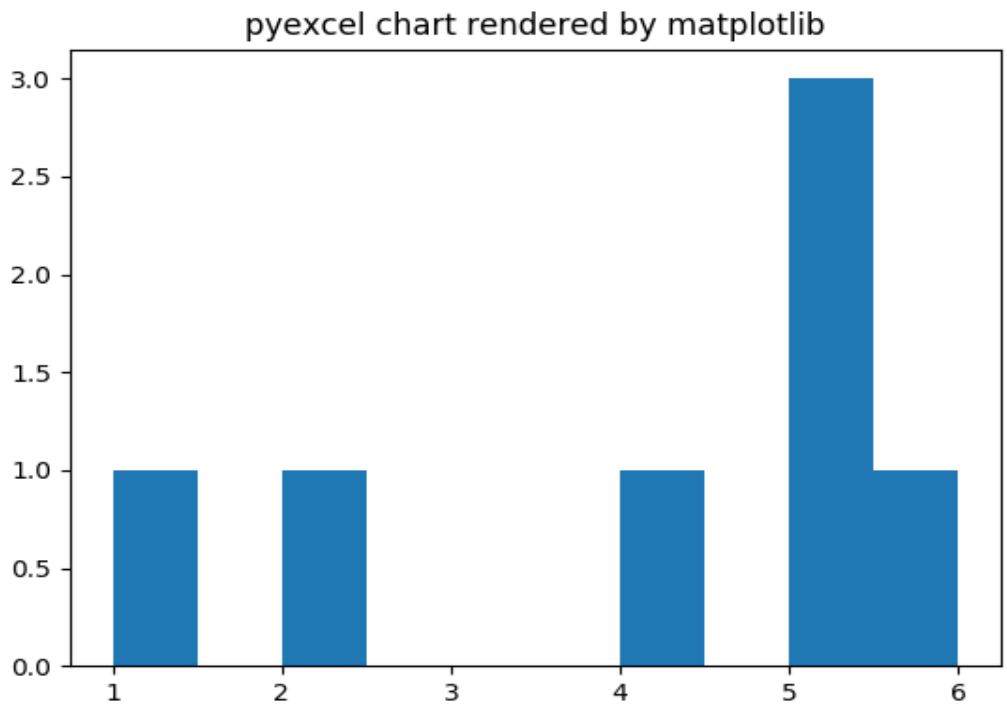
Histogram



Here is the source code using pyexcel:

```
sheet_name = 'Wide bars'
data = [[5, 5, 2, 1, 4, 5, 6],
        [1, 1, 2, 3, 4, 8, 9]]
pe.save_as(
    array=data,
    sheet_name=sheet_name,
    dest_chart_type='histogram',
    dest_file_name='histogram.png',
    dest_alpha=0.5
```

Single histogram



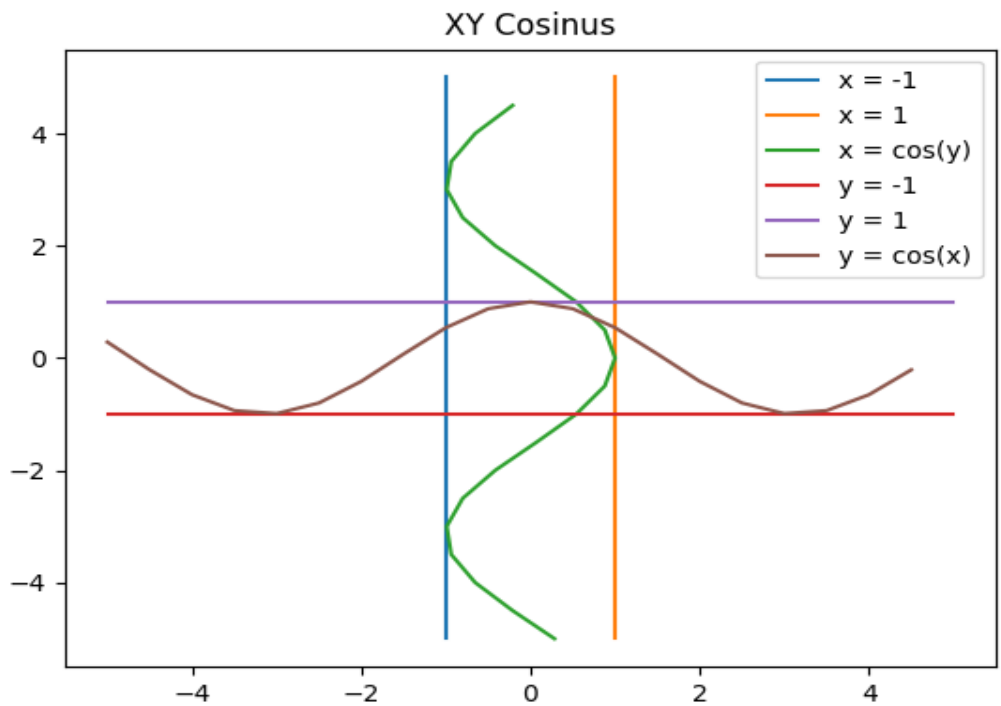
Here is the source code to draw single sheet histogram:

```
sheet_name = 'Wide bars'
data = [[5, 5, 2, 1, 4, 5, 6]]
pe.save_as(
    array=data,
    sheet_name=sheet_name,
    dest_chart_type='histogram',
    dest_file_name='single_histogram.png',
)
```

XY

BASIC

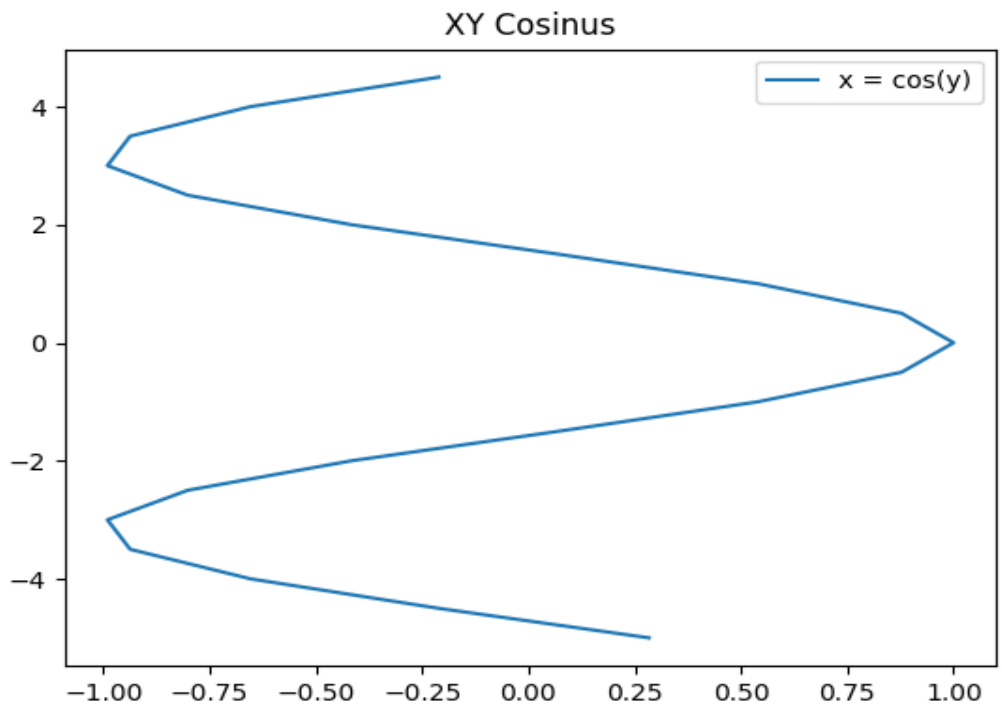
Basic XY Lines, drawing cosinus:



Here is the source code using pyexcel:

```
from math import cos
data = {
    'x = cos(y)': [(cos(x / 10.), x / 10.) for x in range(-50, 50, 5)],
    'y = cos(x)': [(x / 10., cos(x / 10.)) for x in range(-50, 50, 5)],
    'x = 1':  [(1, -5), (1, 5)],
    'x = -1': [(-1, -5), (-1, 5)],
    'y = 1':  [(-5, 1), (5, 1)],
    'y = -1': [(-5, -1), (5, -1)]
}
pe.save_book_as(
    bookdict=data,
    dest_chart_type='xy',
    dest_title='XY Cosinus',
    dest_file_name='xy_cosinus.png',
)
```

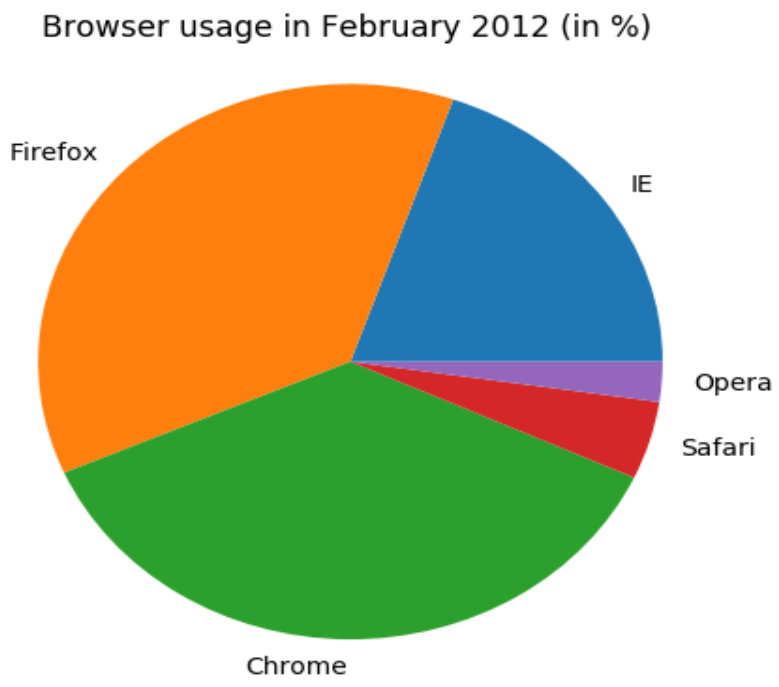
Single xy line



Here is the source code to draw single sheet histogram:

```
from math import cos
sheet_name = 'x = cos(y)'
data = [(cos(x / 10.), x / 10.) for x in range(-50, 50, 5)]
pe.save_as(
    array=data,
    sheet_name=sheet_name,
    dest_chart_type='xy',
    dest_title='XY Cosinus',
    dest_file_name='single_xy_cosinus.png',
)
```

Pie chart



Here is the source code using pyexcel:

```
title = 'Browser usage in February 2012 (in %)'
data = OrderedDict()
data['IE'] = [19.5]
data['Firefox'] = [36.6]
data['Chrome'] = [36.3]
data['Safari'] = [4.5]
data['Opera'] = [2.3]
pe.save_as(
    adict=data,
    dest_title=title,
    dest_chart_type='pie',
    dest_file_name='pie.png'
)
```

Radar chart

_static/radar.png

Here is the source code using pyexcel:

```
    array=data,
    sheet_name=sheet_name,
    dest_chart_type='histogram',
    dest_file_name='single_histogram.png',
)
_validate_and_remove('single_histogram.png')


test_histogram():
sheet_name = 'Wide bars'
data = [[5, 5, 2, 1, 4, 5, 6],
        [1, 1, 2, 3, 4, 8, 9]]
pe.save_as(
    array=data,
    sheet_name=sheet_name,
    dest_chart_type='histogram',
    dest_file_name='histogram.png',
    dest_alpha=0.5
```

Box chart

_static/box.png

Here is the source code using pyexcel:

```
_validate_and_remove(file_name):
from filecmp import cmp
status = cmp(file_name, _fixture_file(file_name))
assert status is True
os.unlink(file_name)


_get_svg_graph_element(file_name):
import lxml.etree as etree

svg = etree.parse(file_name)
root = svg.getroot()
g = root.find('.//g', namespaces=root.nsmap)
```

Dot chart

```
_static/dot.png
```

Here is the source code using pyexcel:

```
_fixture_file(file_name):
return os.path.join("docs", "source", "_static", file_name)
```

Funnel chart

_static/funnel.png

Here is the source code using pyexcel: