
bokeh_wordcloud2 Documentation

Release 1.0

joran beasley

Jun 28, 2019

Contents

1	Getting Started	3
1.1	Installation	3
1.2	Examples	3
2	Indices and tables	7

a (mostly) complete implementation of <https://wordcloud2-js.timdread.com> This package aims to provide a wordcloud extension for bokeh

CHAPTER 1

Getting Started

1.1 Installation

the easiest way to install is with pip

```
# install from pip
pip install bokeh_wordcloud2

# install from github
pip install git+https://github.com/joranbeasley/bokeh_wordcloud2.git#bokeh_wordcloud

# clone and build
git clone pip git+https://github.com/joranbeasley/bokeh_wordcloud2.git
cd bokeh_wordcloud2
pip install . # alternatively `python setup.py install`
```

1.2 Examples

1.2.1 First Word Cloud

here is a super simple word cloud to get you started

you can find this example at [Simple Example Source](#)

you can see its html here [Simple Example HTML](#)

```
1 from bokeh.io import show
2 from bokeh.models import ColumnDataSource
3 from bokeh_wordcloud2 import WordCloud2
```

(continues on next page)

(continued from previous page)

```

4
5 data = [
6     ['susan',6], ['tom',3], ['frankie',8],
7     ['roger',7], ['amy',9], ['nicole',10],
8     ['joran',5], ['mark',4], ['brianne',7],
9     ['michael',8], ['greg',4], ['adrian',6],
10    ['drew',9]
11 ]
12 names,weights = zip(*data)
13 test1 = ColumnDataSource({'names':names,'weights':weights})
14 # we will specify just "blue" for the color
15 wordcloud = WordCloud2(source=test1,wordCol="names",sizeCol="weights",colors="blue")
16 show(wordcloud)

```

when you run the application you should see a webpage open with your newly created Wordcloud

1.2.2 Using a corpus instead

if you dont specify a sizeCol, then it will extract all the words from the wordCol and use the counts as a size

you can find this example at [Example Source](#)

1.2.3 Color Options

for the color options we can pass in a single color that will be assigned to all words

```

# all the words are pink
WordCloud2(source=test2,wordCol="titles",colors='pink')

```

or we can pass in a list, and randomly select from it:

```

# pick a random color
random_colors=['pink','blue']
WordCloud2(source=test2, wordCol="titles", colors=random_colors)

```

or if our datasource has a column for colour we can pass in the name of that (only works if you specify sizeCol)

```

colors = [['red','green','blue','purple'][:4]]
for i in range(len(test1.data['words'])):
    test1.data['colorsCol'] = colors # assign a new column
WordCloud2(source=test1,wordCol="names",sizeCol="weights",
            colors="colorsCol") # use our column name instead

```

or we can supply a javascript callback that returns a string, but we name it slightly differently

```

colorFun = CustomJS(code="""
console.log("PICKING A COLOR FOR:",cb_obj)
console.log("Got Word:",cb_data['word'],cb_data['weight'],cb_data['fontSize'])
return "red" # dont forget to RETURN a value
""")

WordCloud2(source=test1,wordCol="names",sizeCol="weights",colorsFun=colorFun)

```

you can find this example at [Color Options](#)
you can see its html here [Color Options HTML](#)

1.2.4 Clicks And Hovers

you can subscribe to either clicks or hovers with a javascript object

you can find this example at [JS Callbacks](#)
you can see its html here [JS Callbacks HTML](#)

1.2.5 Python Click Callback

Note: this only applies when running bokeh server *bokeh run my_app.py*

you can also subscribe to the click handler in python if you are running with bokeh server

```
wordcloud = WordCloud2(source=test1, wordCol="names", sizeCol="weights", colors=['pink', 'blue', 'green'])

def clicked_word(evt):
    print("GOT:", evt)
    data=test1.data
    if(7 < evt.weight < 20 ):
        new_weight = evt.weight + random.choice([-1,1,1,2])
    elif evt.weight < 7:
        new_weight = evt.weight + random.choice([1,2])
    elif evt.weight < 20:
        new_weight = evt.weight - random.choice([1,2,3])
    weights = list(data['weights'])
    weights[data['names'].index(evt.word)] = new_weight
    # make sure to reassign back to data a new dict, or the difference might not be noticed
    test1.data = {'names':data['names'], 'weights':weights}

# subscribe to the click event
wordcloud.on_click(clicked_word)
curdoc().add_root(column(wordcloud))
```

you can find this example at [Bokeh Server Example](#)
since you must be running with bokeh serve app.py I cannot show you the html

Check out an interactive short tutorial in [jupyter](#)

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`