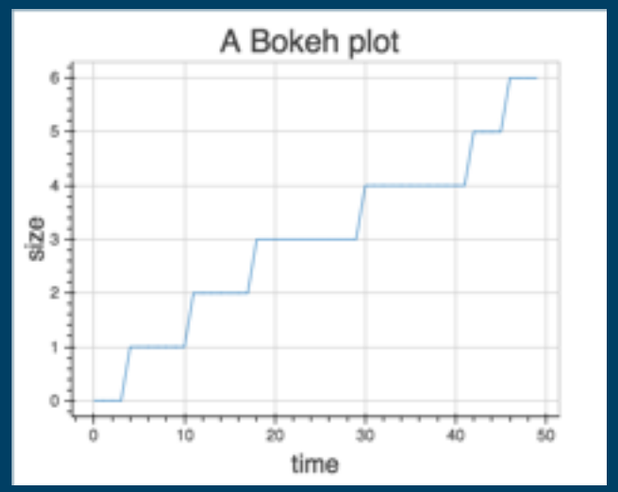
Embedding plots into Flask pages

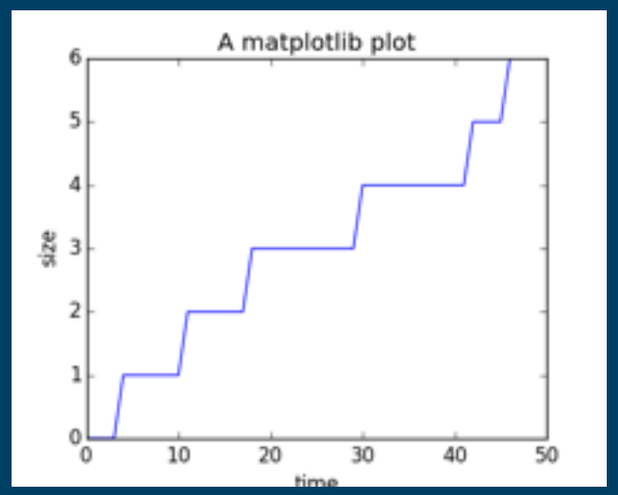
- Sample app at: http://cfss.uchicago.edu/flaskPlotExample.zip
- Approaches:
  - ***Bokeh*** (pure HTML/Javascript)
    - Creates a Javascript program that runs in the browser
    - Embed this whole program directly in the page HTML
  - ***matplotlib*** (temporary image file)
    - Creates a temporary PNG image
    - HTML uses `<img>` tag to reference file

Sample app

- Single page/route that:
  - Generates some random numbers
    `[0,0,0,1,1,1,2,2,3,4,4,4,4,…]`
  - Makes and displays a plot with Bokeh



- Makes and displays a plot with matplotlib



Random numbers (*partial code*)

app.py

```python
from flask import Flask, render_template
import random

app = Flask(__name__)

@app.route('/')
def indexPage():
    # generate some random integers, sorted
    exponent = .7+random.random()*.6
    dta = []
    for i in range(50):
        rnum = int((random.random()*10)**exponent)
        dta.append(rnum)
    y = sorted(dta)
    x = range(len(y))
    return(render_template('figures.html',y=y))


if __name__ == '__main__':
    app.debug=True
    app.run()
```

figures.html

```
<div class="rawData">
    <p>
        Data Points:<br>
        {% for i in y %}
            {{ i }},
        {% endfor %}
    </p>
</div>
```

Bokeh plotting (*partial code*)

app.py

```
# imports for Bokeh plotting
from bokeh.plotting import figure
from bokeh.resources import CDN
from bokeh.embed import file_html, components


...

@app.route('/')
def indexPage():


    ...

    # generate Bokeh HTML elements
    # create a `figure` object
    p = figure(title='A Bokeh
plot',plot_width=500,plot_height=400)
    # add the line
    p.line(x,y)
    # add axis labels
    p.xaxis.axis_label = "time"
    p.yaxis.axis_label = "size"
    # create the HTML elements
    figJS,figDiv = components(p,CDN)


    ...

    return(render_template(
        'figures.html',
        y=y,
        figJS=figJS,figDiv=figDiv,
        plotPng=plotPng))


...
```

figures.html

```
<!doctype html>
<html>
<head>
        <title>Figure examples</title>

        <link rel="stylesheet"
href="http://cdn.pydata.org/
bokeh-0.7.1.min.css" type="text/css" />
        <script type="text/javascript"
src="http://cdn.pydata.org/
bokeh-0.7.1.min.js"></script>
        {{ figJS|safe }}

</head>
<body>

...

<div class='bokeh'>
    {{ figDiv|safe }}
</div>


...


</body>
</html>
```

matplotlib plotting (*partial code*)

app.py

```
# imports for matplotlib plotting
import tempfile
import matplotlib
matplotlib.use('Agg') # this allows PNG plotting
import matplotlib.pyplot as plt

• • •
@app.route('/')
def indexPage():

     • • •
    # generate matplotlib plot
    fig = plt.figure(figsize=(5,4),dpi=100)
    axes = fig.add_subplot(1,1,1)
    # plot the data
    axes.plot(x,y,'-')
    # labels
    axes.set_xlabel('time')
    axes.set_ylabel('size')
    axes.set_title("A matplotlib plot")
    # make the temporary file
    f = tempfile.NamedTemporaryFile(
        dir='static/temp',
        suffix='.png',delete=False)
    # save the figure to the temporary file
    plt.savefig(f)
    f.close() # close the file
    # get the file's name
    # (the template will need that)
    plotPng = f.name.split('/')[-1]

    return(render_template(
        'figures.html',
        y=y,
        figJS=figJS,figDiv=figDiv,
        plotPng=plotPng))
```

figures.html

```
<div class='matplotlib'>
    <img src="static/temp/{{plotPng}}">
</div>
```

Complete Python code

app.py

```
from flask import Flask, render_template
import random

# imports for Bokeh plotting
from bokeh.plotting import figure
from bokeh.resources import CDN
from bokeh.embed import file_html, components

# imports for matplotlib plotting
import tempfile
import matplotlib
matplotlib.use('Agg') # this allows PNG plotting
import matplotlib.pyplot as plt


app = Flask(__name__)

@app.route('/')
def indexPage():
    # generate some random integers, sorted
    exponent = .7+random.random()*.6
    dta = []
    for i in range(50):
        rnum = int((random.random()*10)**exponent)
        dta.append(rnum)
    y = sorted(dta)
    x = range(len(y))


    # generate Bokeh HTML elements
    # create a `figure` object
    p = figure(title='A Bokeh plot',
        plot_width=500,plot_height=400)
    # add the line
    p.line(x,y)
    # add axis labels
    p.xaxis.axis_label = "time"
    p.yaxis.axis_label = "size"
    # create the HTML elements to pass to template
    figJS,figDiv = components(p,CDN)


    # generate matplotlib plot
    fig = plt.figure(figsize=(5,4),dpi=100)
    axes = fig.add_subplot(1,1,1)
    # plot the data
    axes.plot(x,y,'-')
```

app.py (continued)

```
    # labels
    axes.set_xlabel('time')
    axes.set_ylabel('size')
    axes.set_title("A matplotlib plot")
    # make the temporary file
    f = tempfile.NamedTemporaryFile(
        dir='static/temp',
        suffix='.png',delete=False)
    # save the figure to the temporary file
    plt.savefig(f)
    f.close() # close the file
    # get the file's name (rather than the whole path)
    # (the template will need that)
    plotPng = f.name.split('/')[-1]


    return(render_template(
        'figures.html',
        y=y,
        figJS=figJS,figDiv=figDiv,
        plotPng=plotPng))



if __name__ == '__main__':
    app.debug=True
    app.run()
```

Complete HTML template

figures.html

```
<!doctype html>
<html>
<head>
        <title>Figure examples</title>

        <link rel="stylesheet" href="http://cdn.pydata.org/bokeh-0.7.1.min.css" type="text/css" />
        <script type="text/javascript" src="http://cdn.pydata.org/bokeh-0.7.1.min.js"></script>
        {{ figJS|safe }}

</head>
<body>

<div class="rawData">
    <p>
        Data Points:<br>
        {% for i in y %}
            {{ i }},
        {% endfor %}
    </p>
</div>


<div class='bokeh'>
    {{ figDiv|safe }}
</div>


<div class='matplotlib'>
    <img src="static/temp/{{plotPng}}">
</div>


</body>
</html>
```