

Module 4

This week we will be looking at Enterococcus levels in the Hudson River, using data from the organization Riverkeeper (<http://www.riverkeeper.org/>).

Background: Enterococcus is a fecal indicating bacteria that lives in the intestines of humans and other warm-blooded animals. Enterococcus (“ Entero”) counts are useful as a water quality indicator due to their abundance in human sewage, correlation with many human pathogens and low abundance in sewage free environments. The United States Environmental Protection Agency (EPA) reports Entero counts as colonies (or cells) per 100 ml of water.

Riverkeeper has based its assessment of acceptable water quality on the 2012 Federal Recreational Water Quality Criteria from the US EPA. Unacceptable water is based on an illness rate of 32 per 1000 swimmers.

The federal standard for unacceptable water quality is a single sample value of greater than 110 Enterococcus/100 mL, or five or more samples with a geometric mean (a weighted average) greater than 30 Enterococcus/100 mL.

Data: I have provided the data on our github page, in the folder https://github.com/charleyferrari/CUNY_DATA608/tree/master/lecture4/Data. I have not cleaned it – you need to do so.

This assignment must be done in python. It must be done using the 'bokeh', 'seaborn', or 'pandas' package. You may turn in either a . py file or an ipython notebook file.

Questions:

- Create lists & graphs of the best and worst places to swim in the dataset.
- The testing of water quality can be sporadic. Which sites have been tested most regularly? Which ones have long gaps between tests? Pick out 5-10 sites and visually compare how regularly their water quality is tested.
- Is there a relationship between the amount of rain and water quality? Show this relationship graphically. If you can, estimate the effect of rain on quality at different sites and create a visualization to compare them.

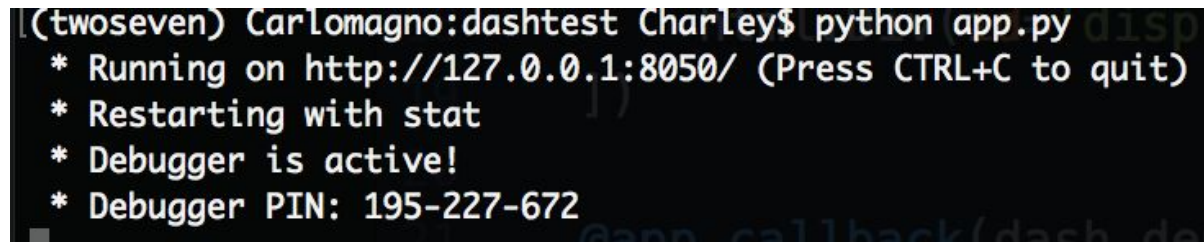
Extra Credit:

Plotly's [Dash framework](#) is awesome. It's a data driven web app development framework built in the Flask ecosystem, and built to work with interactive Plotly graphs right out of the box. Consider it the missing Shiny for Python.

While it was released for open source a couple of months ago and is growing strong, I think it is still a bit too rough around the edges to be a gradable part of this assignment. But, it will provide good practice for when we start getting into d3. Dash is slightly lower level than Shiny, so you'll need to be thinking more about the web development concepts that will come in handy with d3.

The tutorial should give everyone a clear idea of what Dash is all about. Don't worry about deployment at the moment. Just install the libraries, and run the app from the comandline:

```
python app.py
```

A terminal window with a black background and green text. The prompt is '(twoseven) Carlomagno:dashtest Charley\$'. The command 'python app.py' has been executed. The output shows four lines of status messages: '* Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)', '* Restarting with stat', '* Debugger is active!', and '* Debugger PIN: 195-227-672'.

```
(twoseven) Carlomagno:dashtest Charley$ python app.py
* Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 195-227-672
```

The 8050 is the port number. If I navigate to <http://localhost:8050>, I should see my app.

If you choose to create a dash app, all that's required is an app.py along with your Jupyter notebook.

