Graph Display Web Application

- The application constructs clear graphs representing silo data
  - (x,y) coordinates
  - Temperature, with high and low levels indicated by a warm color map
    - Ranging blue (cold) -> red (warm)
    - Dynamic, unique color map for each graph based on the temperature range
- The application stores each graph as an object
  - Title -> Name of the file
  - X -> Binary encoding of a numpy array containing x coordinates
  - Y -> Binary encoding of a numpy array containing y coordinates
  - T -> Binary encoding of a numpy array containing t coordinates
- Upon request, the application generates a scatter plot in matplotlib, encodes it into binary, and serving the requested url with a .png of the generated graph
  - This is very fast, and the web application can generate a fresh graph for any dataset
  - Each graph page has a distinct url based on the .csv filename
- To add new files, the admin must run the population script which generates new objects for every new .csv file in the './data' folder.

Limitations:

- As the graphs are only .png files, there is limited interaction
  - Could be fixed with the use of .js graph libraries that would work better in a web browser
- Can't view multiple graphs at once
  - Could generate a graph for each active .csv file
- Styling
  - The application is purely functional and is not styled appropriately for a userbase
- Sqlite3
  - The application runs sqlite3, this is not optimal for security
    - Employ MySQL for security.

My Thoughts:

This was challenging, I wasted a lot of time trying to set up an MySQL server, and ended up going for SQLite3 as a result. I'm not sure how I would've shipped the SQL server so that you could test the program.

I'm happy with the speed at which the graphs generate, thanks to using binary encoding (same as the population script). But they do lack interactivity, I think it would be good to zoom into the scatter plot to identify points of interest better.