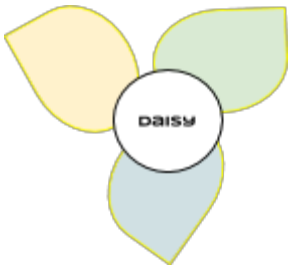


Daisy:
Data Made Easy

http://vpac00.phy.vanderbilt.edu/~piscioja/daisy.html

***Abstract*:**

Data science is an interdisciplinary field that requires a variety of complex skills. Data scientists can be divided into three groups: data engineers, data analysts, and data stewards. Project Daisy attempts to help all three types of people accomplish their research goals. On this cloud-based data portal, researchers who need datasets to answer their questions can use the data collection tools to easily gather data or use datasets made by others. Those who need help with analysis can use code written by others over their dataset. People with the technical and analysis skills can find interesting datasets to work with. Daisy brings together these people from different backgrounds into a collaborative environment in which researchers can easily and openly find and share data and code for analysis and visualization. This allows all people regardless of experience, skill, or discipline participate in research.

Background

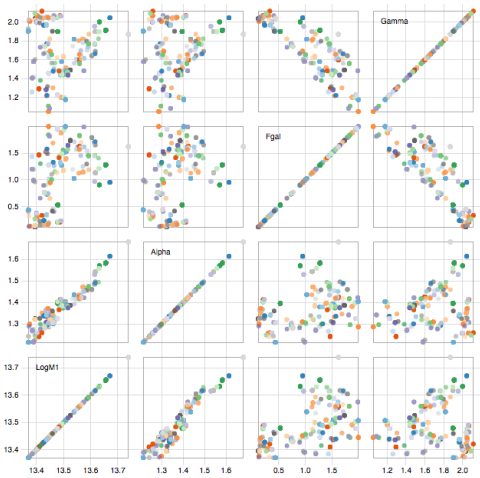
It is becoming increasingly obvious that domain experts need a more efficient way to not only access their data, but perform on the fly analysis. We are presenting a complete workflow portal that will allow a user to, in a well designed and clean user interface, perform an efficient statistical analysis on a data set queried from the open science data cloud. This workflow combines the expertise of data intensive engineers with an analysis tool meant to integrate the total user experience of a scientist.

The ultimate goal of Daisy[1] is to alleviate the complexities that researchers face when doing data retrieval, visualization, and analysis. Daisy's clean user interface combined with the accessibility to data provided by cloud computing gives even the most inexperienced users the fundamental support needed to understand their data.[2] In order to streamline the data retrieval process, Daisy contains code to easily gather data from a variety of sources. Examples include data from Twitter, Protein Database (PDB), and Astrodata. For Twitter, the API only allows for retrieval of tweets up to one month back. The benefit Daisy provides is not only streamlining the process to obtain tweets but also allow for the creation of a database to find tweets further in the past that will be easily searchable. In addition, we plan to integrate the PDB into Daisy's workflow by leveraging OSDC's cloud database to house the large dataset. One of the PDB's weaknesses is that it is a highly redundant dataset. Many of the proteins in this database are repeated submissions of one another, differing only by method obtained and resolution. This becomes an unwieldy problem for researchers, which is why we would also include code to analyze and eliminate the redundancies from the user's desired dataset.[5] Essentially, Daisy aims to help

[1] Website for Daisy: http://vpac00.phy.vanderbilt.edu/~piscioja/daisy.html
[2] The algorithm for Daisy lives here: https://github.com/gracelu/Daisy

researchers who have trouble obtaining data from complex APIs or do not want to spend the time writing data retrieval functions by providing code to perform this task and already created datasets from other users housed on a cloud acting as a data commons.

Data Analysis and Visualization:



In addition to data retrieval, Daisy will house code to perform analysis of the data. This will help researchers who are looking for easy to use analysis tools. On the portal, there would be written code for analysis including text and statistical analysis. For example, Daisy would have code for computational linguistics such as arithmetic compression. Arithmetic compression is used to better understand the structure of a language that can be run with a variety of different datasets, including tweets.[3] In addition, performing a quantitative correlation test on a large database is expensive and researchers are often not necessarily interested in the strength of the correlation but rather the structure of the correlation between parameters. Once these correlated variables have been identified, further more intensive, analysis can commence. In Daisy, we would have an interactive d3.js environment that integrates the retrieved data, plots parameters in a scatter matrix[4] and allows domain experts to cleanly visualize the correlations. By making this plot interactive, users can visualize the n-dimensional correlation at a given time. In the shown webpage we can take the output of the arithmetic compression analysis of the tweets and plot the output against each other, but the input can be any data. Daisy is not limited by the type of data it analyzes.

Future Work:

For now, we have only included one data visualization and analysis tool that does not have quantitative element. The next gen version of Daisy can include a calculation of the linear correlation coefficient between input parameters. Instead of choosing a priori the parameters plotted, Daisy will include an interface with a drop down menu for users to choose and change plotted parameters easily. The color and kind of plotted points can also be chosen to coordinate with an unplotted parameter. We also hope to expand the parameters so that they can be related across databases. For example, one could use Daisy to cross reference and visualize the parameters between a hashtag in Twitter and a phenomena from some other open dataset, such as OpenWeatherMap. We believe that by hosting Daisy's innovative workflow on a public data commons would allow for simple data reproduction and sustainability for researchers and industry workers alike.

Daisy currently exists as an idea for making data science more efficient and accessible to researchers through a cloud-based data portal and workflow. To make this project a reality, this would require integration with OSDC to house the data and code and the participation of data scientists. Daisy will make data science open to everyone, especially those new to research and those without experience with data, analysis, and coding. Daisy is the ultimate way to bridge the gaps between data engineers, data analysts, data stewards, and research enthusiasts.

---

[3] https://github.com/gracelu/Daisy/blob/master/arithmetic%20compression/arithmetic_compression.py

[4] Adapted from http://mbostock.github.io/d3/talk/20111116/iris-splom.html

[5] Adapted from https://github.com/smatlock/Python/blob/master/ids_to_nr.py