# CS-340 Project 2 README

## About the Project/Project Title

For this project the overall goal was to learn how to create and operate a dashboard using data and functions we developed in the previous project. The dashboard is meant to showcase the information and allow the user to easily sort through the data for anything they’re looking for.

## Motivation

This project is both a guide and an experience in learning how a basic dashboard is created using python and mongo. I wanted to see how interactive data charts could be used to express a multitude of information, along with formatting the information to fit different levels of comparison.

## Getting Started

Because this project takes place directly after the previous one where we developed the original login account and data accessing module, I will be focusing on the creation and usage of the dashboard.

To begin creation of your dashboard you need to setup a basic dash layout. This layout should include your initial connection to the database, an initial read of information on the database, and for the purposes of this project an image loader statement for the company logo. With these in place you can use the pulled information to establish a data table that you can then customize as needed. The data table can be made to have many selectable options or editability, however for this project we focused mainly on sorting the information.

Once the data table is working and showing the necessary information, you can begin working on the sorting options. I went with radio items for my sorting method since they had the default of only one being selected at a time. This made the creation of callbacks smoother with less possibility for wrongful sorting. The callback for these sorting options is centered around returning read requests with the desired information entered as a filter.

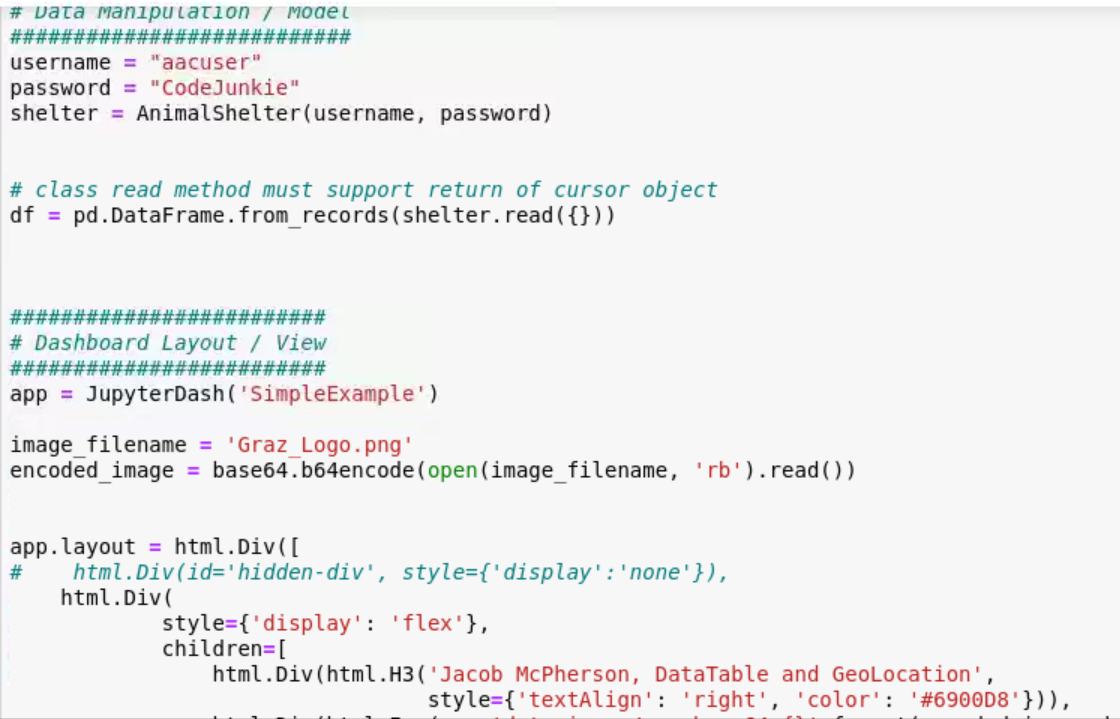
The final piece is the creation of our data graphs. The creation of the map graph was mostly given to us so that we could begin to understand how it was created. After reading through the graph a few times I was able to modify it for a bit more accuracy with output data. The bar graph shown next to the map graph also relies on the information passed by the data table after any filtering has taken place. Bar graphs pull from certain columns of information and compare them to each other. This can improve data comparison for those that get lost looking at a large table of information.

## Installation

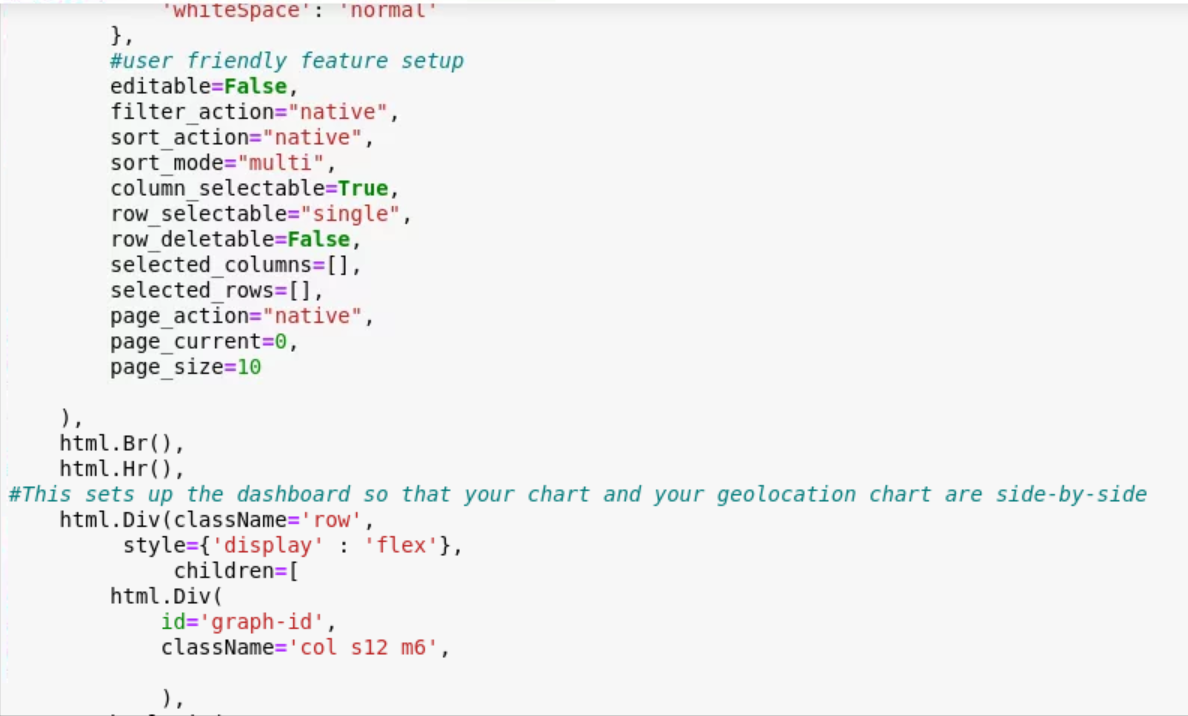
This software and its pertaining database were created using five different software tools. The first of which is MongoDB for the creation of the user account and the database we are messing with. Second you will need Jupyter Notebook for the creation of the module and test dashboard that all provided examples are pulled from. This software is made for easy management of multiple coding documents and developmental test running. Next, to make the connection work and to access the database you will need the Pymongo driver to allow the module to edit or pull information from the database. For this project specifically we will also need to use dash and plotly. Dash is the basis for which our dashboard functionality comes from, while plotly is the provider of the graphing functionality and integration. These two programs are the backbone for this project, so it’s worth mentioning that nothing will work without these two libraries.

## Usage

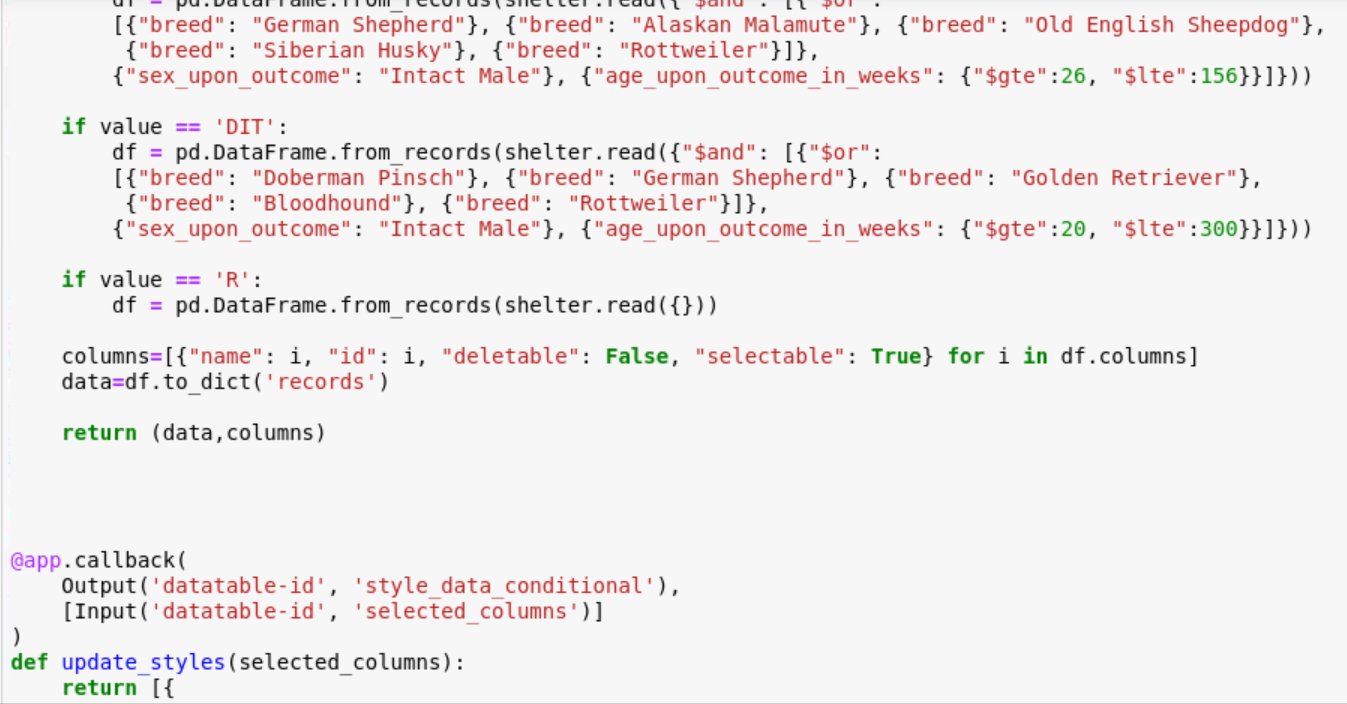
### Code Example













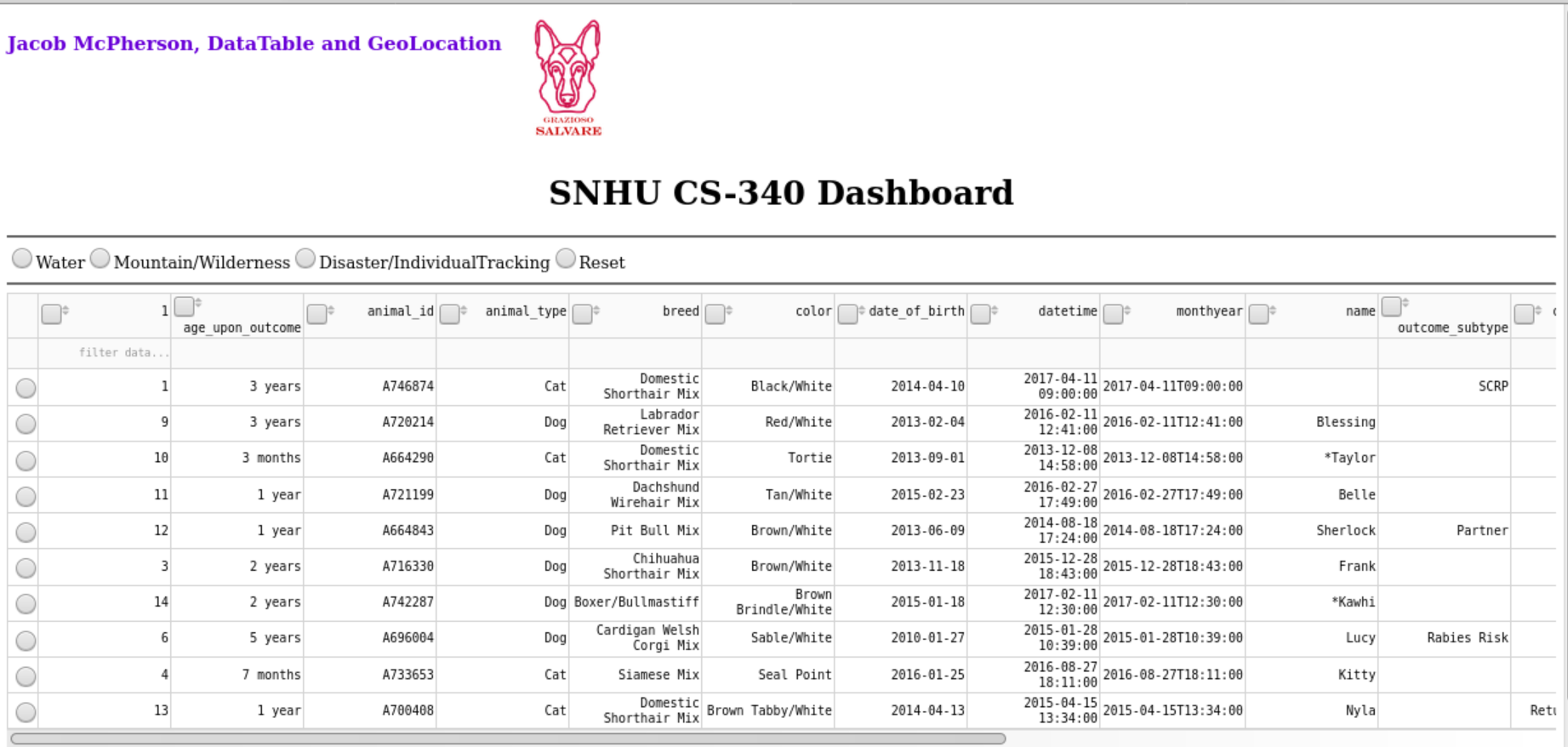


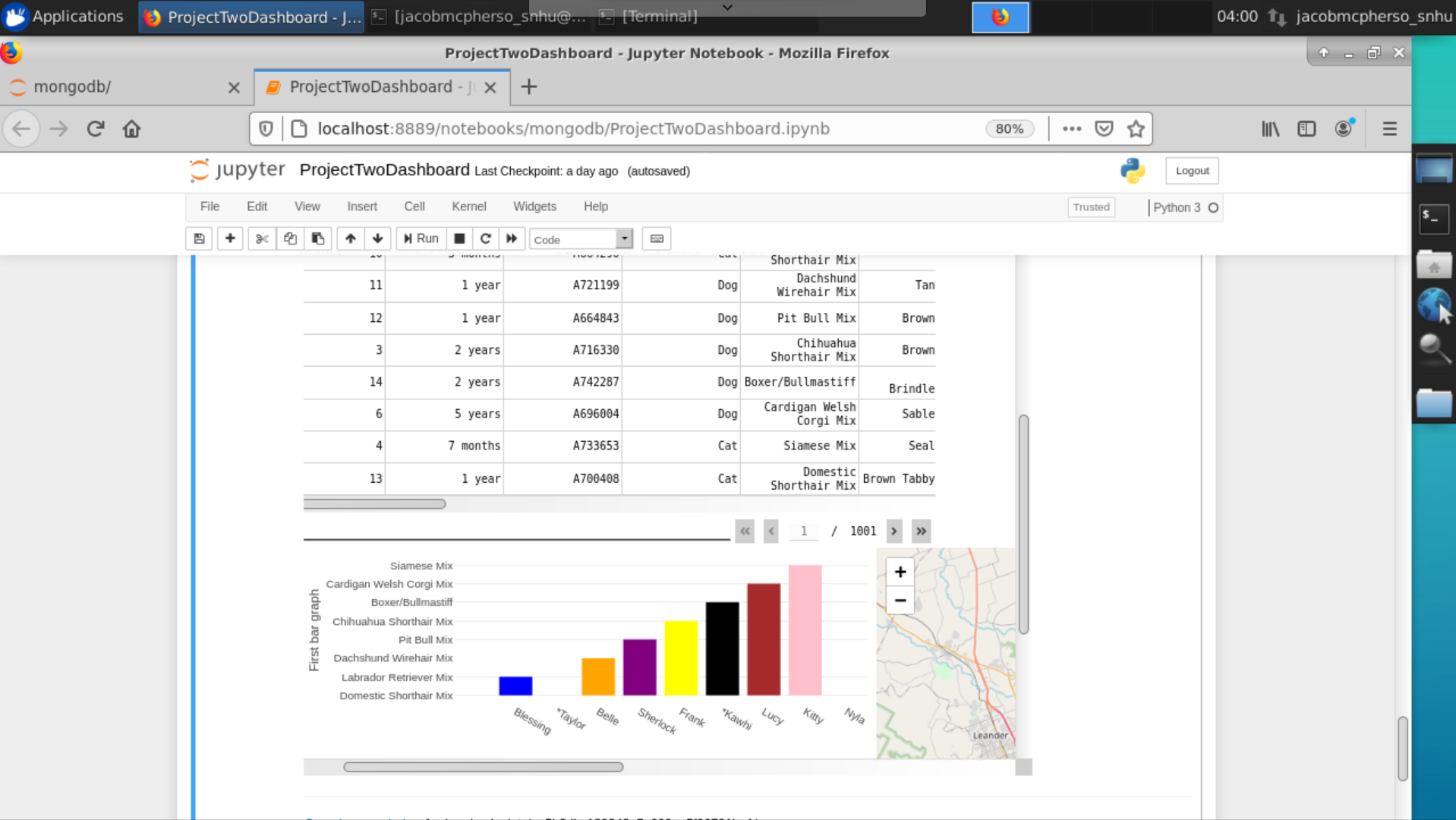
### Tests

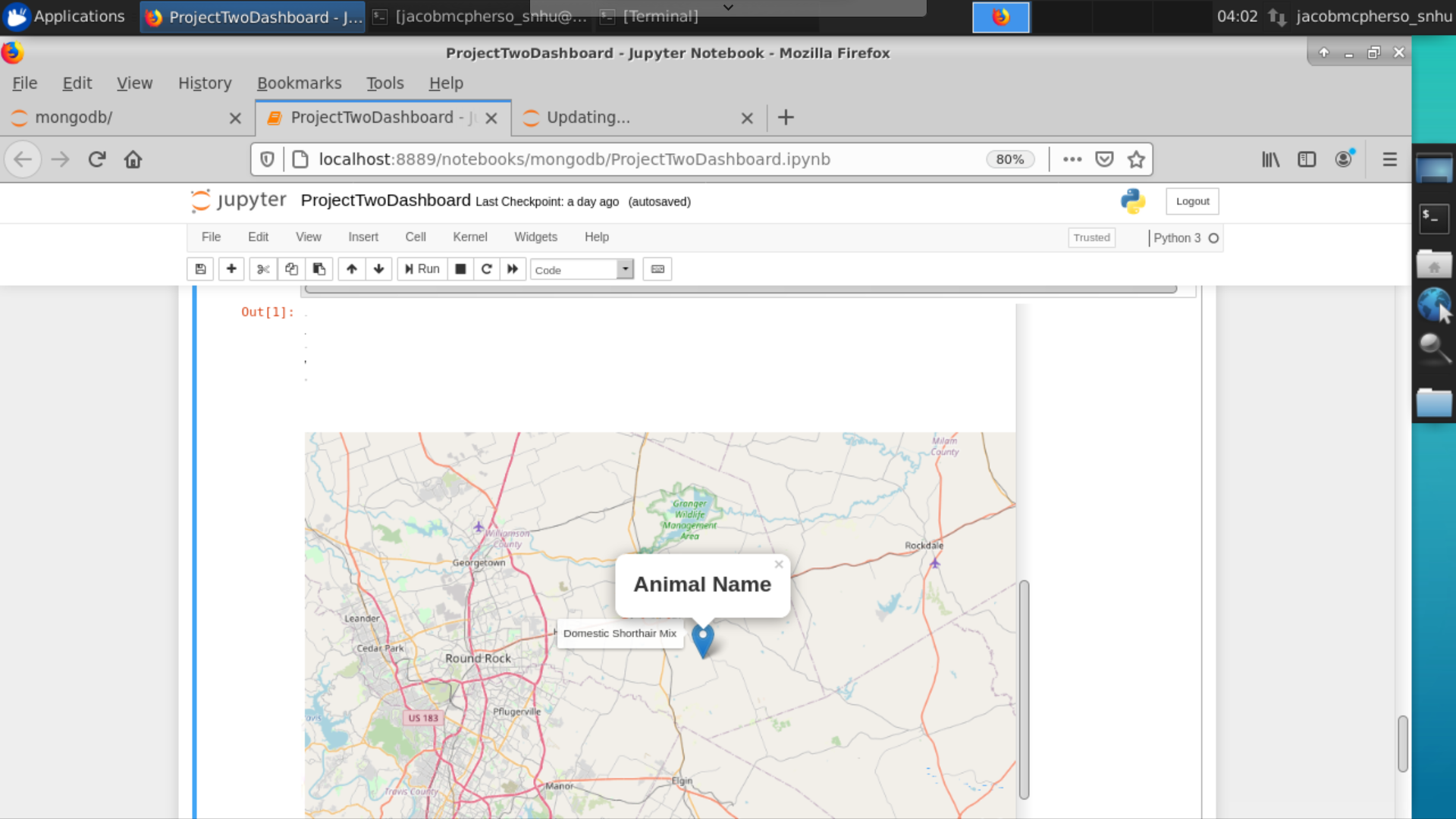
To get a test running for this dashboard you will first need to upload or create data for the dashboard to load.

1. To create a test how we have you must first connect to a server with the necessary database. In our case we connected to one using the login shown in the first screenshot above. This server gives access to the AAC database that we will be pulling from.
2. Next you need to ensure your data frame uses the correct read function provided by the module we created in the previous project. This function will read all available data and will also be used in the radio items to sort the information for our table options.
3. Once your read function is set up, all that’s left is for you to customize the options on your data table. The third screenshot shows the current options setup for the data table provided by this code. You can very easily add your own or alter the ones here.
4. With everything setup, you should see something like the screenshots provided below. The database we pull from contains information on different animals and where they are geologically.

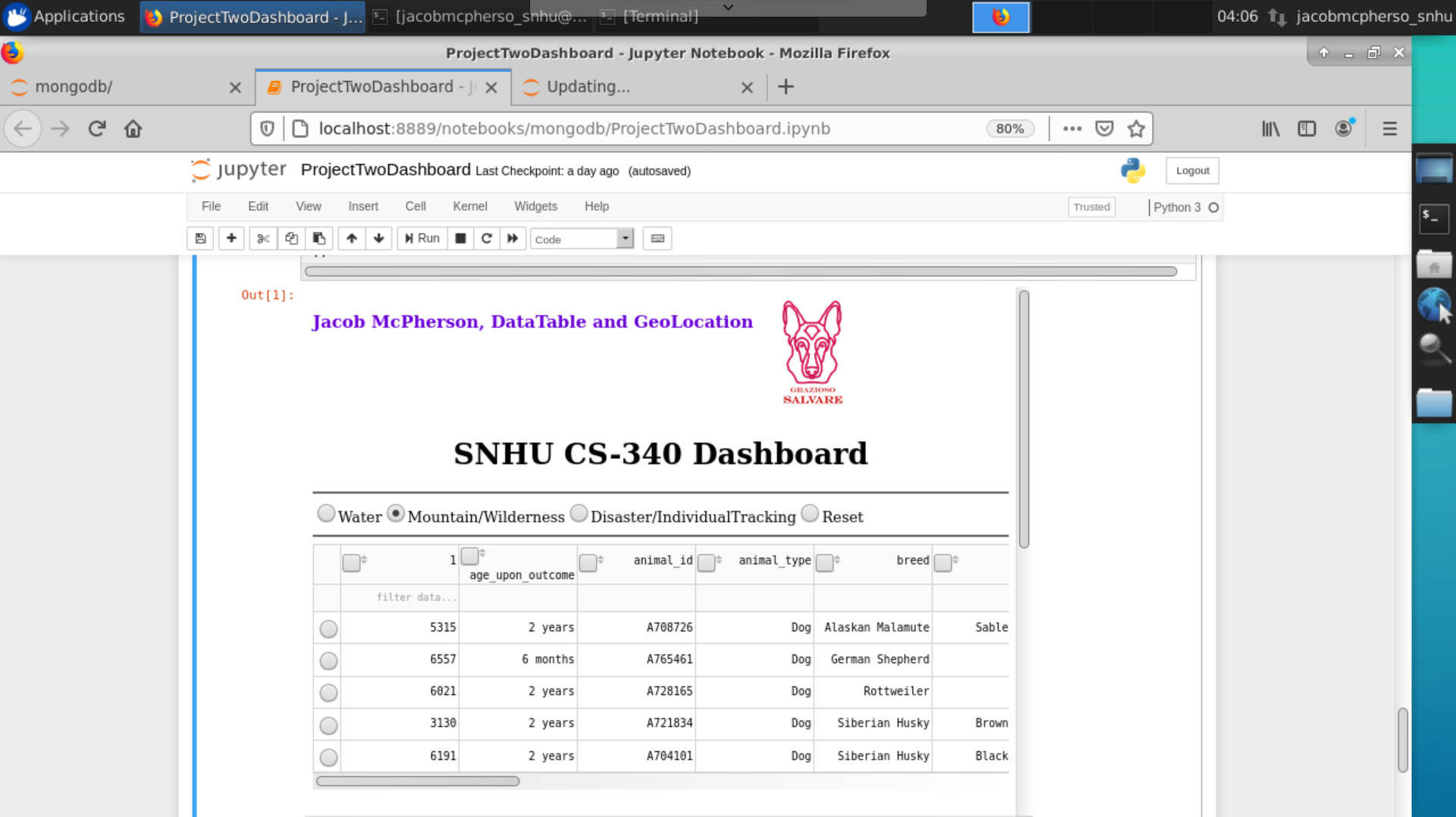
### Screenshots

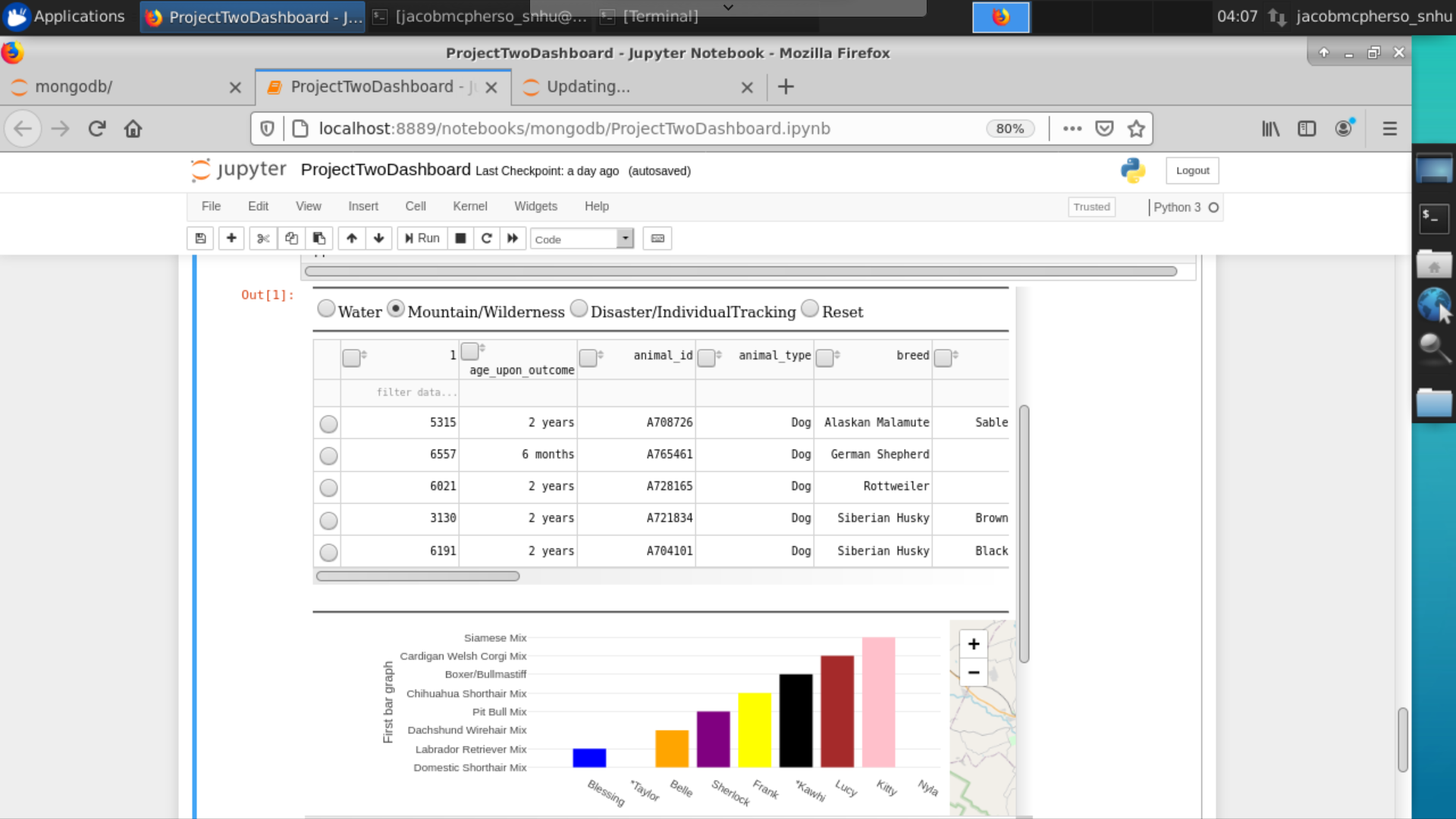


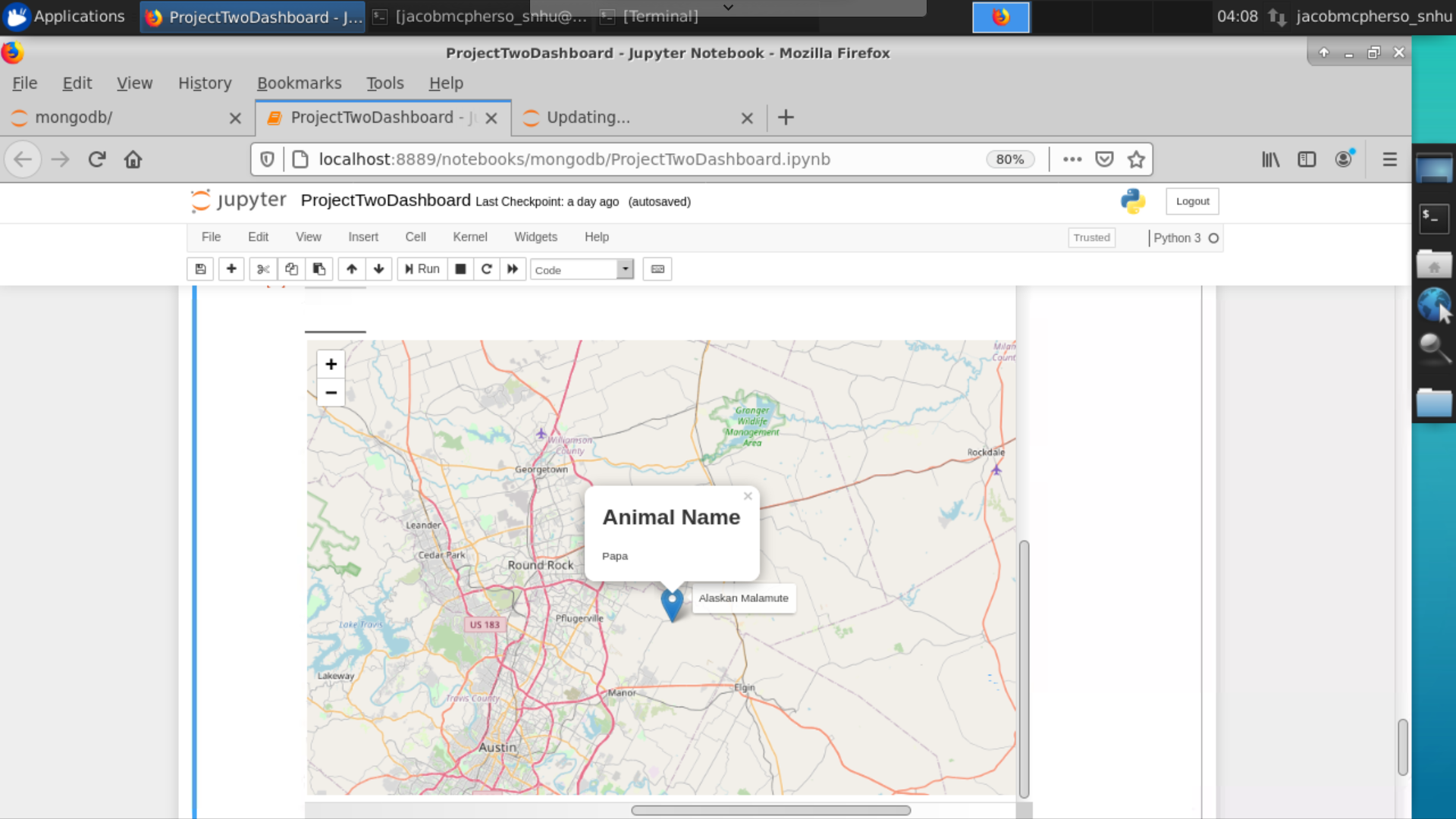




Below you can see how hitting one of the filters changes the output of ALL charts and tables shown.







## Contact

Your name: Jacob McPherson