# CS 340 README

## About the Project/CS340 Dashboard

I have developed a Web Dashboard application to aid users with displaying the information and filtering the document to show the best fit for service dogs. This functionality will allow users to easily find a list of dogs that fit many different criteria and show on a map where they were found near the Austin Animal Center along with two graphs to show the distribution of dogs and their ages. To power this application I am using a python module called AnimalShelter.py which can be found in the project files along with the main dashboard file.

## Motivation

The dashboard was motivated by my initial learning of Dash and HTML and the understanding that a terminal readout of the animals list may be difficult to read for some users. Furthermore, for services that will be needing different breeds of service dogs for different purposes creating the specific queries to find these animals may be cumbersome to type out. So I created a simple to use web application that provides an easy to read table of animals as well as multiple displays of data to aid users with their search for specific animals. Dash was chosen as it provides a simple method to use HTML and Python together to seamlessly create a well organized and easy to use web application.

The main challenges I faced when working on this application were establishing the filters for each type of rescue operation. To get past this I worked on reading through my code one line at a time and checking that my code was correctly running the queries I wanted it to run. I also read a guide for the pandas dataframe method to aid my searches at <https://www.geeksforgeeks.org/python-pandas-dataframe-isin/>.

To complete this project I also needed to create a simple CRUD module in python to work with the application which meant I also needed to learn more about MongoDB and PyMongo to make sure that all of my code would work together and function correctly.

## Getting Started

To get this dashboard working with your system first the system must be set up to work with MongoDB.

Also, the Mongo Host and Port in the initialization function of the program must also be set to the system’s Host and Port to allow the program to access the database. This initialization can be found in the python module AnimalShelter.py. Modification of the included module code will be able to be done through any text editor such as notepad or Linux nano.

After these steps the system will be able launch the dashboard and read through the database.

## Installation

First, the Python language must be installed onto the system for which there are multiple methods to install such as a standard installation package from Python.org for windows or if the system is using Linux the terminal can install it through the method sudo apt-get install python3.

The pymongo library will be needed to use this module so this will need to be installed with the terminal after installing Python through the method !pip install pymongo.

MongoDB will also be necessary which can likewise be installed through a standard installation package from MongoDB.com for windows or using the terminal sudo apt-get install -y mongodb-org method.

The Dash program will also need to be installed onto the system through the method “pip install dash” along with the pandas data analysis tool which can be installed through the method “pip install pandas”.

Once these are installed your system should be ready to test out the dashboard and start working with the database.

## Usage

This program should be used by users that need to be able to quickly read through a large database to find specific breeds and age ranges of dogs to be trained in specific situations.

This dashboard provides a dropdown menu of established use cases of animals such as ones able to be trained for water rescue. This is used by simply clicking on the dropdown menu and selecting your specific use case and the dashboard will automatically update to show relevant animals based on established criteria.

### Code Example

Dashboard Update Functionality:

@app.callback(Output('datatable-id','data'),

[Input('filter-type', 'value')])

def update\_dashboard(filter\_type):

if filter\_type == 'Reset':

dff = df

elif filter\_type == 'Water Rescue':

dff = df[df.breed.isin(['Labrador Retriever Mix', 'Chesapeake Bay Retriever', 'Newfoundland'])

& (df.sex\_upon\_outcome == 'Intact Female')

& ((df.age\_upon\_outcome\_in\_weeks >=26) & (df.age\_upon\_outcome\_in\_weeks <= 156))]

elif filter\_type == 'Mountain or Wilderness Rescue':

dff = df[df.breed.isin(['German Shepherd', 'Alaskan Malamute', 'Old English Sheepdog', 'Siberian Husky', 'Rottweiler'])

& (df.sex\_upon\_outcome == 'Intact Male')

& ((df.age\_upon\_outcome\_in\_weeks >= 26) & (df.age\_upon\_outcome\_in\_weeks <= 156))]

elif filter\_type == 'Disaster or Individual Tracking':

dff = df[df.breed.isin(['Doberman Pinscher', 'German Shepherd', 'Golden Retriever', 'Bloodhound', 'Rottweiler'])

& (df.sex\_upon\_outcome == 'Intact Male')

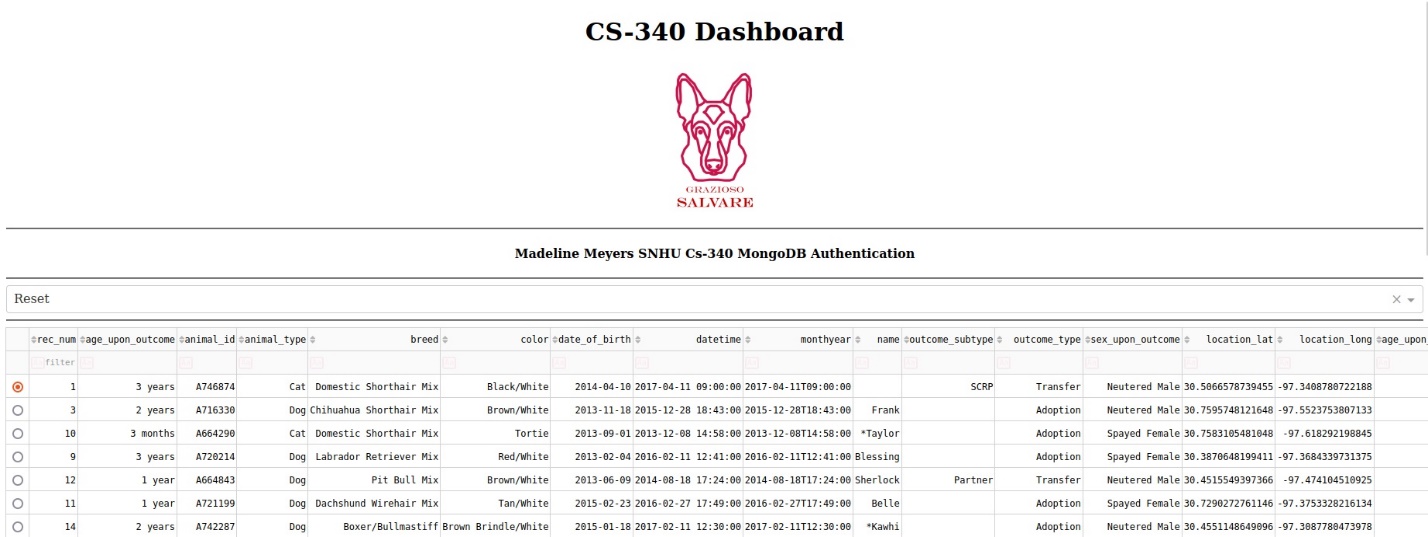
& ((df.age\_upon\_outcome\_in\_weeks >= 20) & (df.age\_upon\_outcome\_in\_weeks <= 300))]

return dff.to\_dict('records')

### Tests

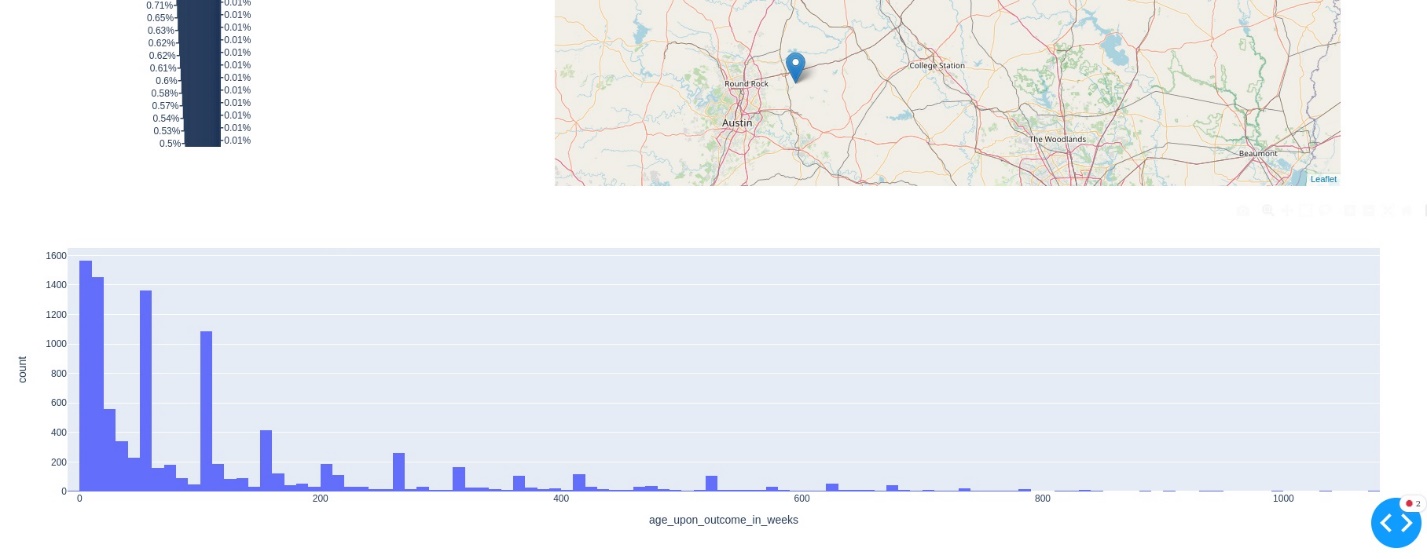
To test this module, I recommend simply running the dashboard and selecting any use case to see what results are available. This information can be double checked by using a terminal to use MongoDB with the Austin Animal Center outcomes database and performing a query that would match the expected parameters of the animal use case being searched for. The terminal readout should be an identical match to the animals listed in the table created by the dashboard. In the code example I show these queries and they should be simple to replicate within a terminal query using MongoDB.

### Screenshots

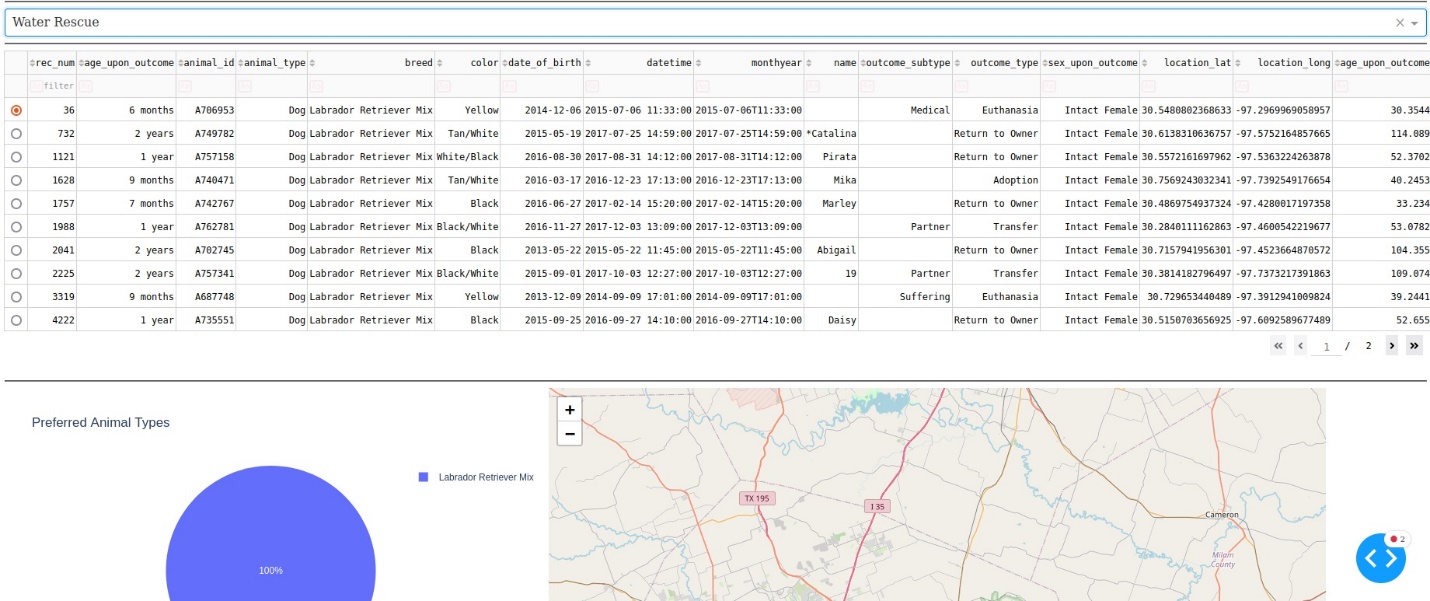
**

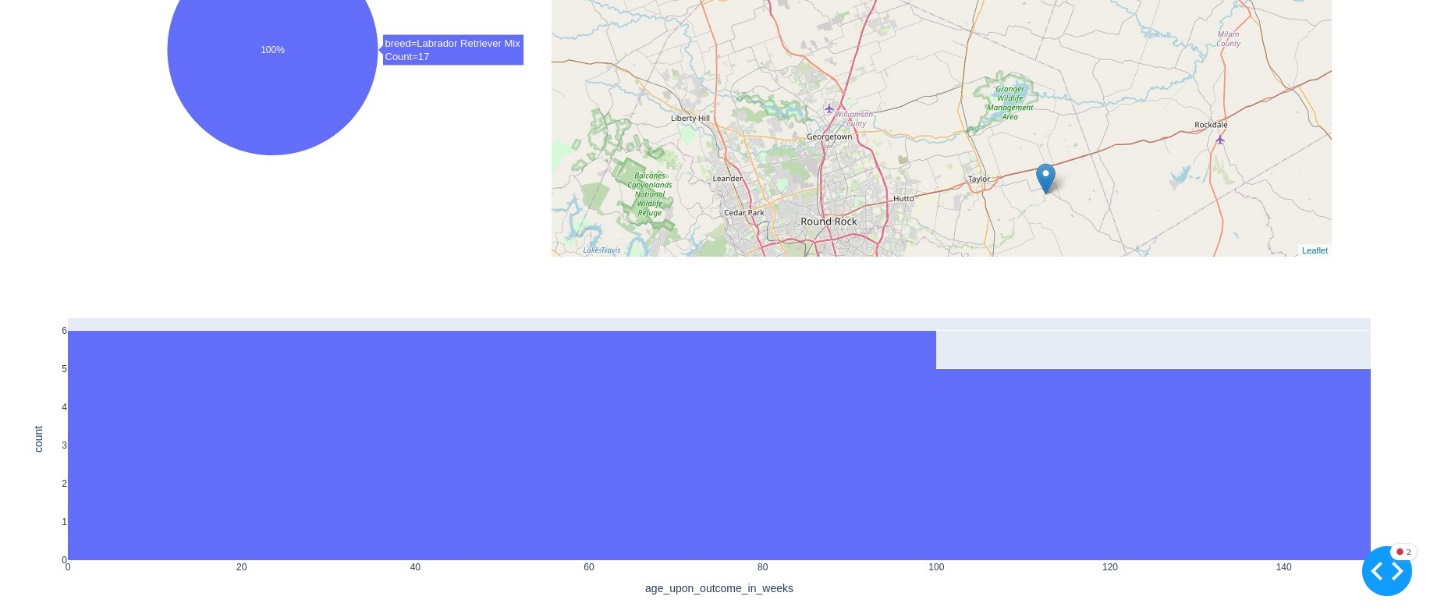
A screenshot of a map

Description automatically generated

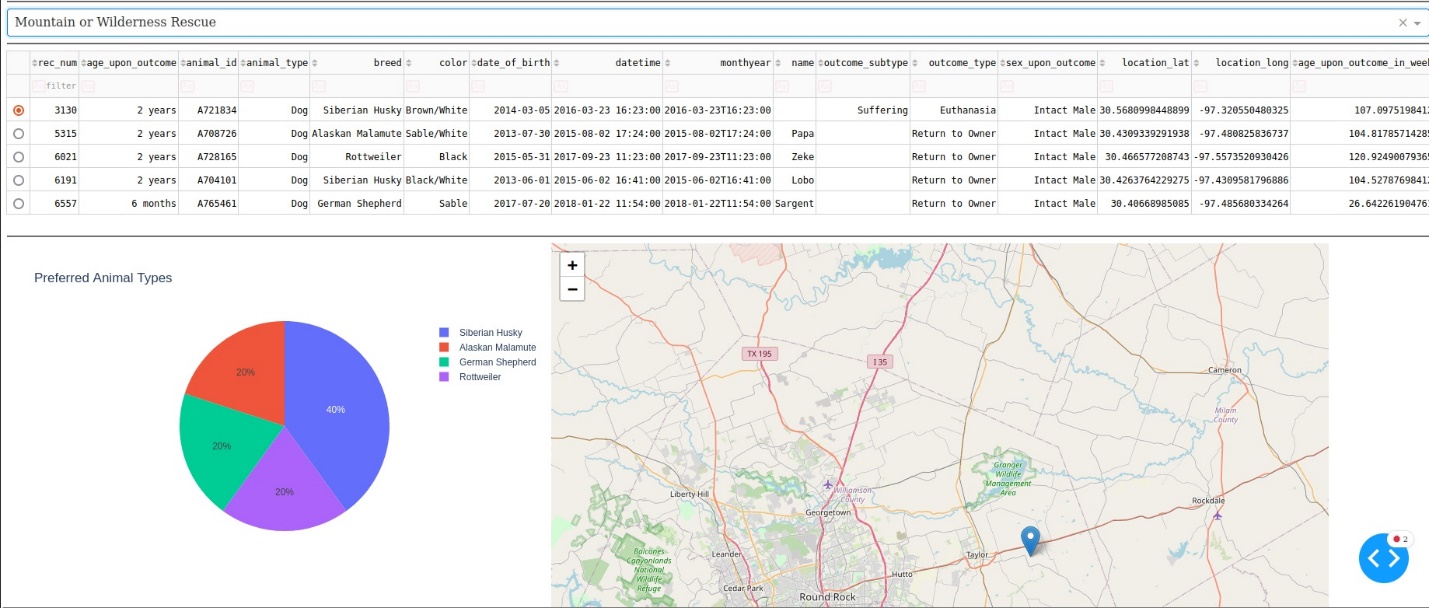
**

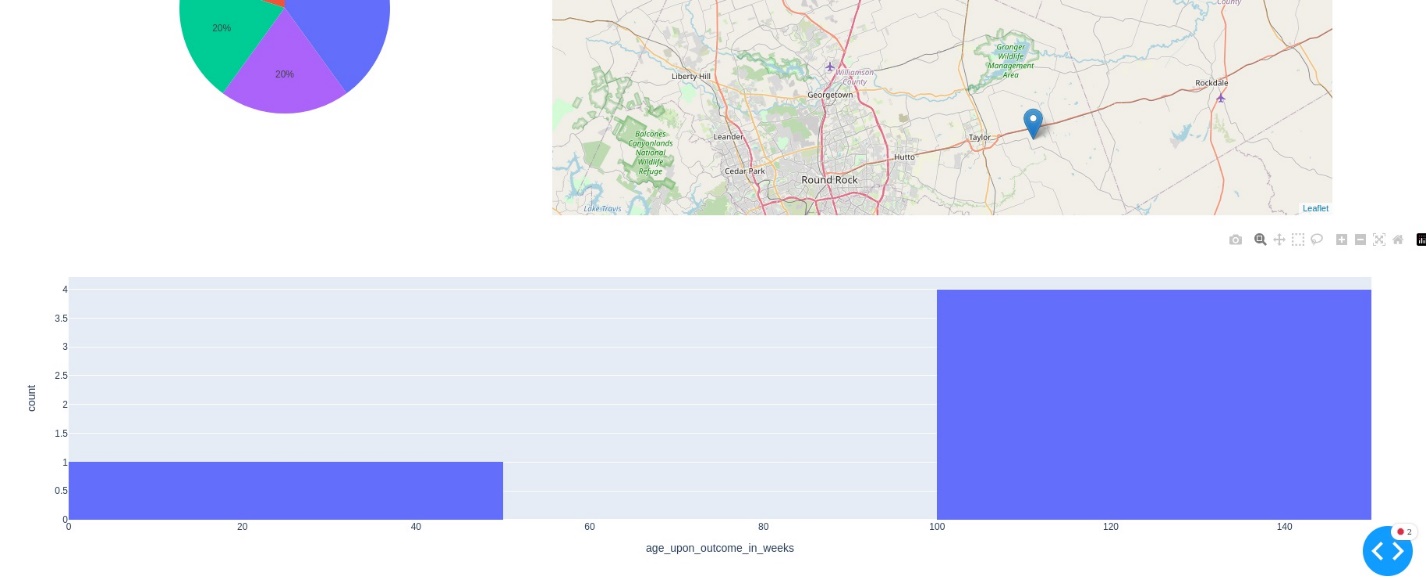
These screenshots show the result of the default search for animals along with the logo for the company I am working with.

**

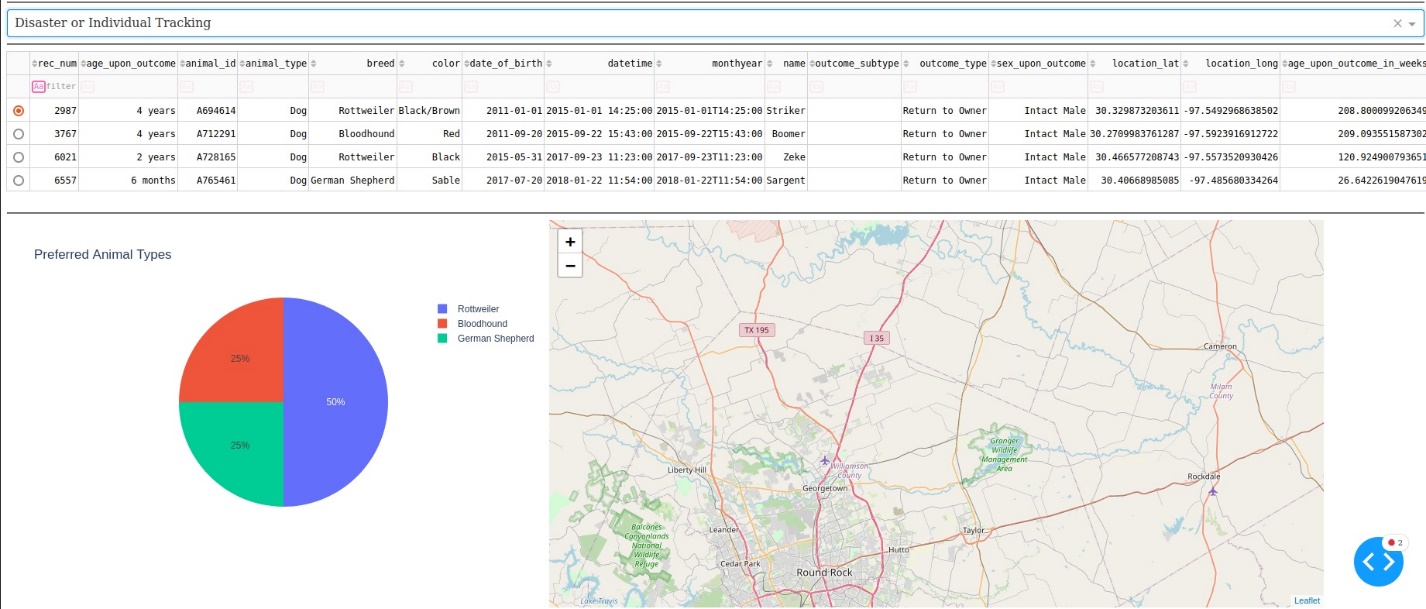
**

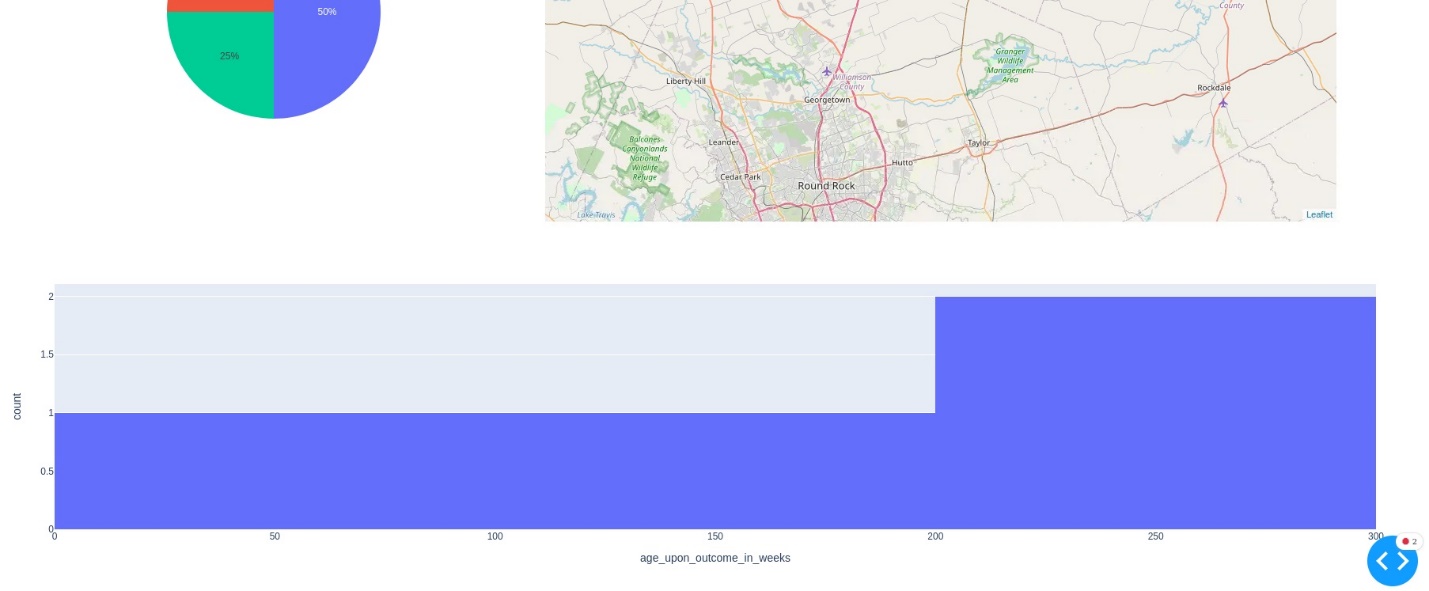
These screenshots show the results of searching for dogs that can be trained for Water Rescue.

**

**

These Screenshots show the results of searching for dogs that can be trained for Mountain or Wilderness Rescue.

**

**

These screenshots show the results of searching for dogs that can be trained for Disaster or Individual Tracking.

Each of these screenshots use the dropdown menu to select each type of search.

## Contact

Your name: Madeline Meyers