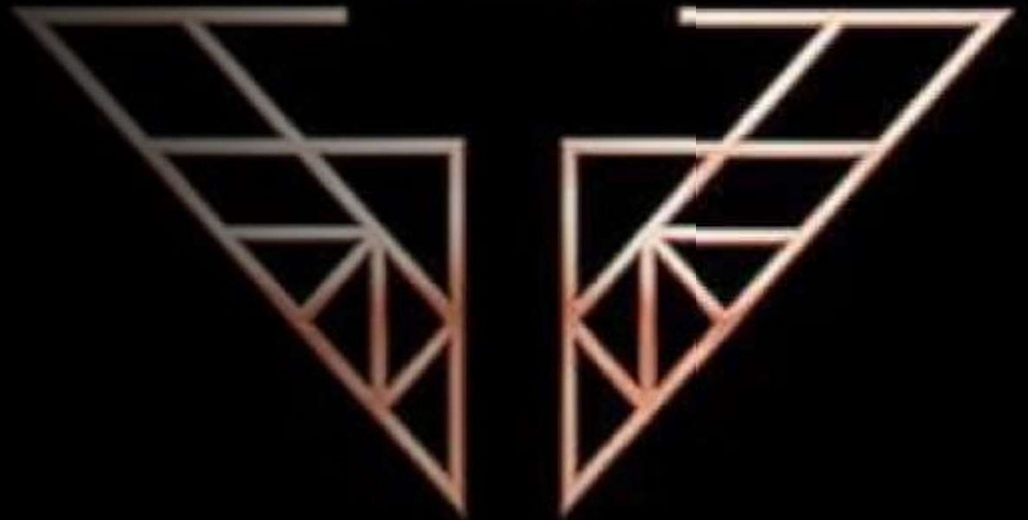


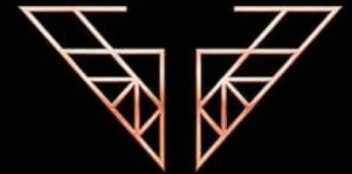
QuickVis

Charlie's Angels

PEARC 20 Hackathon



Meet the Team



Helena Coker

lenacoker4@gmail.com



Najm Mohamed

najmyemen@gmail.com



Cesar Monsalud III

cesarmonsaludiii@gmail.com



Hector Santiago III

hector.santiagomarti@gmail.com

Git Hub

https://github.com/hectorsantiago5/pearc20_quickviz



Problem Tasked



Scientists need a tool to visual data quickly in a simple and easy manner without computational science experience on their part.

Scope

Import

Users can import their own datasets

Choose

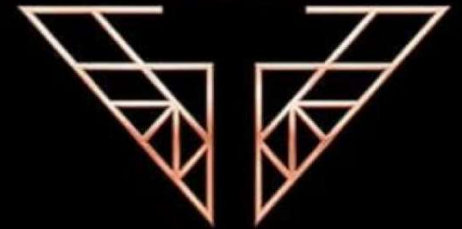
Users can select how their data is processed

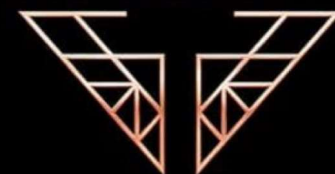
Plot

User's data is visualized geographically or graphically

Analyze

User's visualization and results are analyzed





What We Learned

How to use

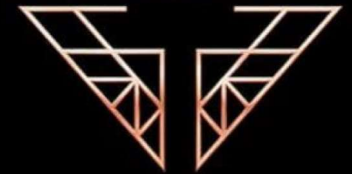
- Leaflet, Echarts, charts.js to display data on graphs and maps
- Passing data between any application and a jupyter notebook
- Learned how to make a website with 4 pages interacting
- Learned how to use google cloud
- Learned how to code in CSS
- Learned how to use GitHub and Repl.It properly



What We Started With

Started with:

- Nonfunctional static map
- Nonfunctional file uploader
- Rough single page website template



Deliverables

A fully functioning tool set that takes users data and displays it as a map or graph

Mapping Tool

Welcome to the Mapping Tool

Files must be uploaded in a .csv format.

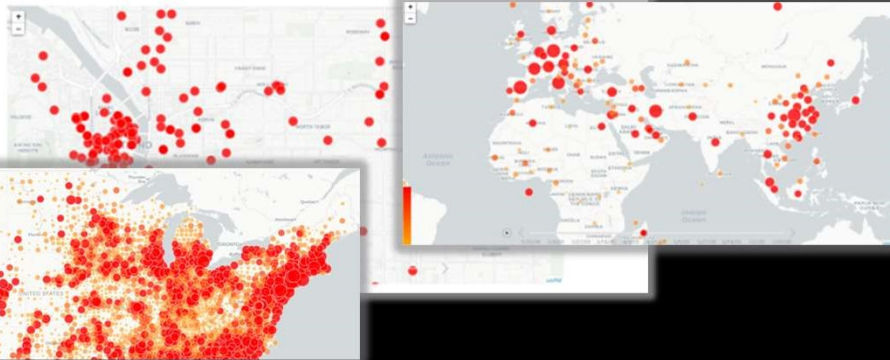
Instructions on how to save your file to the proper format is [here](#).

For .csv file formatting to ensure compatibility with the mapping tool:
Location descriptors for example country and city must be in first (1st) and second (2nd) column or column A and B.
Latitude must be in third (3rd) column or column C.
Longitude must be in fourth (4th) column or column D.
Single data classification must be in the fifth (5th) column or column E.

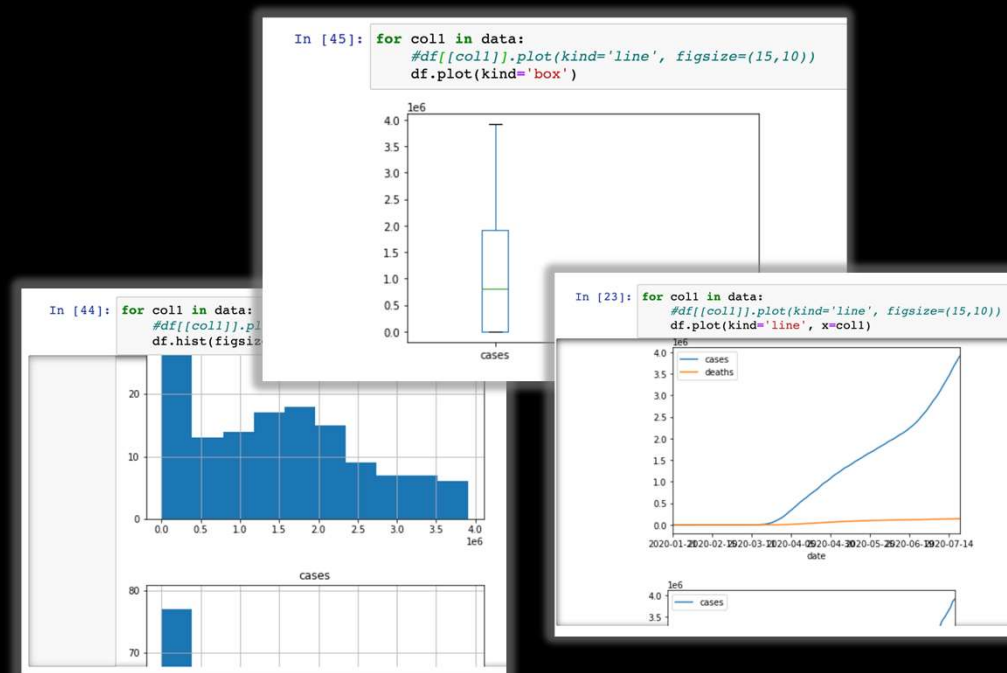
Click [here](#) for a sample file about community resources in Portland Oregon.
Click [here](#) for a sample file about confirmed COVID-19 cases in the U.S.A.
Click [here](#) for a sample file about how many people recovered from COVID-19 globally.

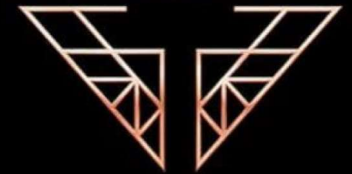
Click on the "Choose File" button to upload a file:

RoseCityResou...ormatted.csv



Graphing Tool





Future Work

- Working on expanding displaying users' data types
 - Box plots, line plots, bar plots, etc.
 - Mapping multi-variable dataset
- Statistical Analysis
 - Linear regression
 - Box plot and scatter plot and toggle bell curve feature
 - p-value with either standard alpha value (0.05) or feature to add their own
- Machine Learning and Predictive Analysis
 - One to locate areas of interest in a dataset
 - One for user assistance such as suggesting analysis types or possible formatting conflicts
 - One for user personalized such as recalling analysis they previously used on the site upon return

Our Thanks To



Charlie Dey



Linda Hayden

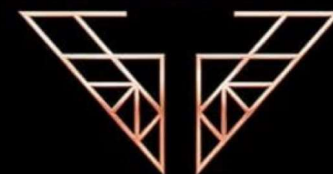


Chris Lanclos



Marlon Pierce

Our Thanks To

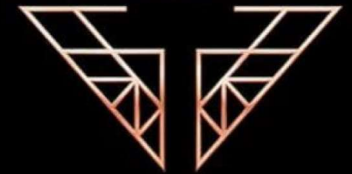


Git Hub



https://github.com/hectorsantiago5/pearc20_quickviz





Data Sources

- https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series
- <https://opendata.imspdx.org/dataset/rose-city-resource-dev2>
- <https://www.bfro.net/>
- https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series
- <https://data.world/us-doe-gov/0fd3e1b2-0e53-4e37-b822-7c3e810fe78c>



Demo Time!

The image shows a Zoom meeting interface. On the left, a PyCharm IDE window is open, displaying Python code for a web application. The code includes a Flask application with a route for '/file' that handles file uploads and a route for '/upload' that processes the uploaded file. The code is as follows:

```
from flask import Flask, request, jsonify
import os
import shutil
import uuid

app = Flask(__name__)

@app.route('/file', methods=['POST'])
def upload_file():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'})
    file = request.files['file']
    if file.filename == '':
        return jsonify({'error': 'No selected file'})
    if file and file.filename != '':
        filename = file.filename
        file.save(os.path.join(upload_path, filename))
        return jsonify({'message': 'File uploaded successfully'})

@app.route('/upload', methods=['POST'])
def process_upload():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'})
    file = request.files['file']
    if file and file.filename != '':
        filename = file.filename
        file.save(os.path.join(upload_path, filename))
        return jsonify({'message': 'File uploaded successfully'})
```

On the right, a Zoom window shows four participants in a 2x2 grid. Below the participants, a presentation slide titled 'Deliverables & Demo' is displayed. The slide contains the following text:

Deliverables & Demo

A fully functioning public facing tool set that takes users data and displays it as a map or graph

Mapping Tool **Graphing Tool**

The slide also features a map and a graph, both showing data points in red.