Progress Evaluation - Milestone 3

Title: Customizable Analysis and Visualization Tool for COVID Cases

Team Members:

- Calvin Burns, cburns2017@my.fit.edu (Team Lead)
- Sam Hartle, shartle2017@my.fit.edu
- Stian Olsen, shagboeolsen2017@my.fit.edu
- Nicole Wright, nwright2017@my.fit.edu

Advisor: Dr. Philip Chan, pkc@cs.fit.edu

Client: Dr. Philip Chan, pkc@cs.fit.edu

Progress Matrix for Milestone 3:

Task	Completion %	Stian	Sam	Nicole	CJ	To do
1. Continue Feature 4.1 (Customizable Operations on Variables)	60%	30%	30%			Fixing UI details and interacting with more backend operations
2. Small GUI demo that integrates lockdown and mask mandate data	80%		40%	10%	30%	Add the mask mandate data to the plot
3. Consider different options for saving plots	100%	30%			70%	None

Discussion of each accomplished task and obstacles for Milestone 3:

- Task 1:

  The model is complete for an operation. The model utilizes the SymPyCharField, which will allow a user to input any operation(s) that is supported by the SymPy library. This library will do all of the heavy-lifting for processing operations and generating results.

  We have designed a UI which allows the user to create a plot based on public or private datasets. The UI allows the user to filter datasets based on columns, variables and operations. The UI is also able to select a plot type which will visualize the filtered data.

- Task 2:

  This task is about 80% done. That's because we have decided on a charting software and we have created a csv importer which is able to import any csv file. The backend functions needed to format the data in the data frames are completed. So, what's left for this task is to send the data frame for the mask mandate plot to the backend functions for the selected plot type and render it on the page.

- Task 3:

  We considered a few different options for saving the plots. A good solution for saving plots is important in this project as we are dealing with large amounts of data. Throughout the semester, we have been looking into the pros and cons of storing the data in a database, json/csv storage and storing the data as binary. We also had to look into storing the data in memory versus storing it on disk. After comparing the different options, we ended up storing the data in a pickle file. When we store the data in a pickle file we store the data as binary. Pickle is a python way to serialize data. The pickle is stored on disk, so we don't need to worry about running out of memory. The time it takes to save and load the data is also improved a lot. The Florida Department of Health (FDOH) data, which was originally xxx MB, was compressed to 44 MB with the binary format.

  A new idea we have been looking into is storing the data frames based on user sessions. So, we store the data in a cache and keep it there while the user is active. Data not used will be taken away from the cache after a certain time. This idea was brought up to lower the time it takes to load data. Some logic is needed here to determine how much data we will store in the cache and for how long we will keep it there to make sure we don't risk running out of memory.

Discussion of contribution of each team member to Milestone 3:

- Stian: Did research on three different tools for visualizations: ChartJS, ApexCharts and Highcharts. Did demos on all of them, but chose to go with Highcharts as they have all the charts we would need and they had a free licence for non-profit organizations. Did time and space comparisons on the different file formats for saving dataframes. Used the View Plot page which CJ created and created reusable components for the different chart options we currently have (line, scatter, bar, pie). Created python functions for each of these plot types which uses the plot's data frame to format the data the way Highcharts need it to be in order to render it. Looked into highcharts_panda which CJ found and informed me about. Highcharts_panda serializes the data frame and organizes it so Highcharts can use it. When creating a demo with the library, we found out that it was outdated and that the serialized data could not be used.

- CJ: Researched new table options after original database setup proved to be inefficient. Found Python Pandas. Researched Pandas and created proof of concepts for table/data manipulation. Created generic CSV importer that converts a CSV to a compressed DataFrame saved as a pickled object. Write documentation explaining the basics of Pandas along with links to resources for teammates. Work with Stian for alternative charting softwares. Found Highcharts. Documented conversion to Pandas. Met with the team to walk through the setup. Set up an external storage server for holding pickle files via Amazon S3. Developed our program to save files to the S3 server if that option is enabled. Setup auto delete. Created the following pages, Public Data, Private Data, Upload Dataset, Create Plot, and View Plot.

- Sam: Worked with CJ to research different operations to support those listed in our requirement document. Made bug fixes to the login and registration pages. This involved making the "Incorrect Username or Password" banner message show up on the same page that the data was entered instead of incorrectly redirecting back to the homepage and displaying both a successful login message and an incorrect login message. In addition, added buttons to both of these pages to allow the user to return to the homepage if they so desire. Wrote the majority of the progress evaluation and worked with Stian on the presentation. Uploaded and was able to query the "Florida Lab Testing" CSV dataset as a Pandas dataframe object on my local machine. At the end of the milestone, I was working on making a demo which would carry out linear regression on a time series from the lab testing dataset. However, I needed to take a step back and start with a simpler problem first. I am currently working on a plot which will show the positivity rate at each date from the dataset. Most of the work is through the operations. Ideally, this work will be able to be integrated with the *Create Plot* GUI that CJ is working on.

- Nicole: Researched and fixed local environment issues that blocked me from running Django commands (issues with OSGeo4W and gdal libraries.) Solution was to download pip wheel and use pip to install OSGeo4W and apply some configurations. Next, I was able to import mask mandate data into Postgres with the script I wrote during the previous milestone, however, this was before we decided to switch to using Pandas instead. Once we switched to Pandas, I successfully imported my mask data csv file through the “upload dataset” page. I also worked a lot on designing our plot creation page. Me and CJ met several times to brainstorm ideas for the interface of the plot creation page. We researched links to other Pandas GUI’s and picked out what we wanted to be displayed and how to do it. I drew a sketch of our initial design with all of the different pieces of the interface.

Task Matrix for Milestone 4:

Task	Stian	Sam	Nicole	CJ
1. Continue work on scatter plot	Example for Dr. Chan on why we would use a scatter plot for our application Finish development on plot type (85%)	Assist as needed (5%)	Assist as needed (5%)	Assist as needed (5%)
2. Continue work on male/female pie chart	Assist with different chart types (10%)	Assist with various operations as needed (10%)	Add additional “cards” to Create Plot interface which shows general plot details (20%)	Work on Create Plot GUI and using specified filters/operations instead having to “program” (60%)
3. Continue work on creating lab testing plot	Assist with different chart types and operations research (10%)	Plot positivity rate vs. date Focus on operations: count, sum, division (60%)	Assist as needed (5%)	Ideally this will be able to done through GUI as well Continue to build Create Plot GUI abstractly (25%)

Discussion of each planned task for Milestone 4:

- Task 1:

  This task will mainly revolve around Stian continuing work on determining if the scatter plot would be a good choice for our application. He needs to consider the variables that would be plotted on each axis. If so, he would be finishing development for support for this plot type. He was the sole initial developer for this task, so other team members will assist him as he needs.

- Task 2:

  This task will mainly revolve around CJ continuing work on the male/female pie chart. His work on this task will involve finishing the functionality of the Create Plot GUI and allowing users to specify specific filters/operations instead of having to “program” their own plot. Nicole will add additional “cards” to the Create Plot GUI which will allow the user to input general plot details such as name, type, etc. Stian will assist with different chart types as CJ and Nicole see fit. Sam will assist with various operations as CJ and Nicole see fit.

- Task 3:

  This task will mainly revolve around Sam continuing work on creating the lab testing line plot. This plot will show positivity rate over time. He will be focusing on the different backend operations needed to display this plot properly (count, sum, and division). Stian will assist with different chart types and operations research. CJ will be continuing to build the Create Plot GUI abstractly. This is so that the plot will ideally be able to be created through the interface once it has been initially programmed. Nicole will assist any of the other three group members as needed with this task.

Date(s) of meeting(s) with Client/Advisor (same) during Milestone 3:

- November 6th
- November 20th

Client feedback on the current milestone: See Faculty Advisor Feedback below

Faculty Advisor feedback on each task for Milestone 3:

- Task 1:
    1. Abstract CSV importer
        - Look at the input data file to see if it conforms to what pandas is expecting
        - FDOH case line data vs. lab testing data
- Task 2:
    1. Scatter plot for Stian. Make it more purposeful. Come up with an example of why you would like a scatter plot as it's a different thing than line plots.
    2. At least make the male female example work using GUI and specified filters/operations
        - Hopefully as well line graph of cases GUI
    3. For lab testing plot positivity rate vs date
        - Hopefully also through the GUI
- Task 3:
    1. Look into downsides with using pandas
        - Tradeoffs between speed and memory capacity
    2. “Have you considered using a database and use sql to query for results, which are usually much smaller than the datasets? Sql supports many of the operations we are considering including sum/count..., group by, filter ("where" in sql). The query results are (usually much smaller) tables, which can be stored in python/panda and further operations can be applied if needed before plotting. If you haven't considered it, I suggest you think about it. If you have, I would like to hear the pros and cons compared to your current approach.” - Email from Dr.Chan on Nov 21, 2020
    3. Exporting to pickle to save memory
        - Rely on garbage collection or potentially explicitly releasing memory
            - If relying on garbage collection, you don't have control what to keep or not in the memory; ie implementing your caching policy would be difficult
        - Caching policies, look up some memory management policies in OS
            - What to cache can be at least two different levels:
                - dataset level
                - commonly calculated results (e.g. case count vs dates)
            - https://en.wikipedia.org/wiki/Cache_replacement_policies