**Lab Report Title: Lab 10 – Python**

Quinn Rison

4/23/2025

**-Introduction**

This lab involves developing software to search a water quality database for specific information using an LLM AI. Various Python functions will be created to efficiently locate, clean, and display data within a seemingly messy database. The Python Pandas library will be utilized for data manipulation and analysis, offering capabilities such as reading and writing data from various sources, filtering, sorting, handling missing data, and performing statistical operations. Additionally, Streamlit, an open-source Python framework, will be used to develop and deploy web-based data applications, converting Python scripts to HTML and enabling online app development and deployment.

**-Methods/Tests**

The laboratory exercise requires a computer with internet access and a Python compiler and IDE such as Anaconda Cloud or Google Colab.

The pandas Python library is used to access a .csv file downloaded from USGS. The file named station.csv is examined using Excel or Google Sheets to understand its contents, including headers and types of information. The file is uploaded to the cloud environment if using a cloud service. A Python function is generated to access the database, filter for water quality measurement site names, and display location information without repetition. A map is created to pinpoint the location of every station in the database.

A different database containing water quality test results is accessed. The file named narrowresult.csv is examined using Excel or Google Sheets to understand its contents, including headers and types of information. The file is uploaded to the cloud environment if using a cloud service. A Python function is generated to access the database, filter for a desired water quality characteristic, and plot the results, with each site represented as a separate line in different colors. The code is modified to allow filtering for two characteristics simultaneously.

A Streamlit app is developed to allow users to upload both databases used in Parts 1 and 2, search for a contaminant, and define the range of values and dates to display. The map is updated to show the location of stations with the contaminant within the specified range and timeframe, along with a trend over time. The Streamlit environment is set up by creating a GitHub account, repository, and linking it to Streamlit. The app is created, deployed, and edited to add required functionalities. Debugging is performed using the Simple Browser window, and necessary libraries are installed. A requirements.txt file is added to the repository, listing the required libraries. Prompts used to obtain help from the AI, repositories created, and the app URL are included in the lab report's results section.

**-Results**

Goal- Complete question 2 and 3 of part 1.

Model- Microsoft Copilot

Prompt- “ I have been given a csv file containing specific information from stations that the USGS monitors. I need a python function generated to access the database on the csv file, filters for the names of water quality measurement sites, and display the location information for all sites without repetition. After, create a map that can pinpoint the location of every station in the database.” After the prompt I attached the CSV file given. Here is what was produced.

Goal – complete questions 2 of part 2

Model – Microsoft Copilot

Prompt – “ Execute the following task: generate a Python function that accesses the database, filters for turbidity, and plots the results with each site represented by a separate line with a different color where the Y-axis represents the measured values and the X-axis represents time”

Goal- Complete question 2 and 3 of part 1.

Model- Microsoft Copilot

Prompt- “ I have been given a csv file containing specific information from stations that the USGS monitors. I need a python function generated to access the database on the csv file, filters for the names of water quality measurement sites, and display the location information for all sites without repetition. After, create a map that can pinpoint the location of every station in the database.” After the prompt I attached the CSV file given. Here is what was produced.

Goal – differentiate between pH and turbidity and have the same color for a single site for both measurements.

Task- make it easier to differentiate between pH and turbidity (diamonds vs solid line) and one site should possess one color

Prompt – “To differentiate between turbidity and pH lines, can you generate the pH values as small diamonds instead of a solid line? Modify the code so that the same sites possess the same color on the graph.”

Goal – create an app for StreamLit

Task – create a python code that can take parts 1 and 2 and create an app within StreamLit

Model – Microsoft Copilot

Prompt – “In part 1 you helped me create a python code that could make an interactive map. In part 2 you helped create a python code that would display two water quality parameters over time. Now develop a Streamlit app that allows the user to upload both databases used in Part 1 and 2, to search for a contaminant in the databases. Once a contaminant has been selected you should be able to define the range of values and dates that you want to show.”

Goal – create an app for StreamLit

Task – create a python code that can take parts 1 and 2 and create an app within StreamLit

Model – Microsoft Copilot

Prompt – “In part 1 you helped me create a python code that could make an interactive map. In part 2 you helped create a python code that would display two water quality parameters over time. Now develop a Streamlit app that allows the user to upload both databases used in Part 1 and 2, to search for a contaminant in the databases. Once a contaminant has been selected you should be able to define the range of values and dates that you want to show.”

Goal – fix the error regarding latitude and longitude

Model – Microsoft copilot

Prompt – “Fix the error regarding this error message: “

**Website url:** [**Streamlit**](https://official-lab-10.streamlit.app/) **or** [**https://offiial-lab-10.streamlit.app**](https://offiial-lab-10.streamlit.app)

For whatever reason it has removed information from the app to prevent data leaks.

**-Discussion**

What is to be learned from this lab is that there are efficient methods to prompt AI to get it to do tasks very fast. Although the app made in this lab isn’t currently working, I was satisfied with what it was able to do when it was. Although it was a three-hour lab and it felt as though not much had gotten done, all things considered, the app was pretty cool. Being able to tell AI to execute a task and it can do it in a few seconds is amazing. It will be wild to think about the capabilities this tool will have in the future.

-**References**

Copilot. (2025). Explanation of the Pandas Library and Streamlit Framework. Retrieved from Microsoft Copilot.

Copilot. (2025). Methods for Lab Exercise on Python and AI. Retrieved from Microsoft Copilot.