POL 487 (Name searching Code) (Spring 2025)

This code was useful in matching names between data sets from the FEC and from the professor's research. < You will have to change key pieces of information in the code for it to work for you, because your User ID isn't mine.>

The general Idea behind this code, is upload the data to your computer first,
Then check if it can be uploaded into your platform. I used a Jupyter notebook, which you can open through your terminal.

To open a Jupyter Notebook, navigate to the desired directory in your terminal or command prompt and type jupyter notebook. This will launch the notebook server and open a web browser with the notebook interface. Alternatively, if you are using an IDE like VS Code, you can install the Jupyter extension and open notebooks directly from the IDE.

Steps to open a Jupyter Notebook:

1. Open a terminal or command prompt:

.
Opens in new tab
This is where you will type the command to launch the notebook server.

2. Navigate to the desired directory:

.
Opens in new tab
Use the cd command (change directory) to navigate to the folder where your Jupyter Notebooks are located.

3. Type jupyter notebook:

.
Opens in new tab
This will launch the Jupyter Notebook server and open a browser window with the notebook interface.

4. Select a notebook:

.
Opens in new tab
In the browser window, you will see a list of files and folders. Navigate to and click on the notebook file (identified by its .ipynb extension) to open it in a new tab.

Alternative methods:

Using Anaconda Navigator:

.
Opens in new tab
If you have Anaconda installed, you can launch Jupyter Notebook through the Anaconda Navigator application by clicking on the "Launch" button for JupyterLab.

Using VS Code:

.
Opens in new tab
Install the Jupyter extension for VS Code and open notebooks directly from the IDE.

After opening the notebook you can run this code in it as long as the files are uploaded into your computer and you change your username to your actual username. You Can use a File pathway to create a " link" in the notebook for the Nobook to access the data.

Next you create multiple file pathways to open multiple datasets at the same time.

Then I had to change the names of the columns so it would make sense to an A.I. to write code for this data set, which is edited by me <you>.

POL 487 (Name searching Code) (Spring 2025)

For example ESEFirst_name Should be changed to “First name”. So the large language model isn't confused.

Next you can detailedly explain what is happening to Chat GTP or another A.I. to help you write code to search for names in the files. You can double check if the code works by using it in the Jupyter notebook on the exact files you have already < mind you when you get Code from A.I. it often needs to be edited, it’s trial and error if you have a lack of experience like me>
Lastly, you then can also check the data sets by opening each individual file and Control (f) Find, and search to see if those name matches or “ pullings” are accurate. “ it should be because A.I. isn't going to hallucinate in Jupiter because Jupiter isn't A.I.

Remember your job is to leverage A.I. in the research , not let A.I. do the research.

Code below>>>>

```
import os

file_path = "/Users/raymondlizzolii/Documents/exe_test_3.csv"  # Make sure this matches exactly

# Check if the file exists
if os.path.exists(file_path):
    print("✅ File found!")
else:
    print("❌ File NOT found! Check the file path.")
```

```
import os

file_path = "/Users/raymondlizzolii/Downloads/exe_test (1).csv"

if os.path.exists(file_path):
    print("yes file exists!")

else:
    print("NArh? No filefound!")
```

```
import pandas as pd

file_path = r"/Users/raymondlizzolii/Downloads/exe_test (1).csv"
```

POL 487 (Name searching Code) (Spring 2025)

```
df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/exe_test (1).csv")
print(df.head())
```

```
import pandas as pd  # Import the pandas library

# Define the file path
file_path = r"/Users/raymondlizzolii/Downloads/exe_test (1).csv"  # Update this if needed

# Load the CSV file into a DataFrame
try:
    df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/exe_test (1).csv")
    print("File loaded successfully!")
    print(df.head())  # Display the first few rows of the dataset
except FileNotFoundError:
    print("Error: File not found. Check the file path and try again.")
except Exception as e:
    print(f"An error occurred: {e}")
```

```
import pandas as pd  # Import the pandas library

# Define the file path
file_path = r"/Users/raymondalizzolii/Downloads/exe_test (1).csv"  # Update this if needed

# Load the CSV file into a DataFrame
try:
    df = pd.read_csv(file_path)
    print("File loaded successfully!")
    print(df.head())  # Display the first few rows of the dataset
except FileNotFoundError:
    print("Error: File not found. Check the file path and try again.")
except Exception as e:
    print(f"An error occurred: {e}")
```

```
import pandas as pd  # Import pandas

# Define the file name (assumes it's in the current working directory)
```

POL 487 (Name searching Code) (Spring 2025)

```
file_name = "exe_test (1).csv"

try:
    # Load the CSV file into a DataFrame
    df = pd.read_csv(file_name)
    print("File loaded successfully!")
    print(df.head())  # Show the first few rows
except FileNotFoundError:
    print("Error: File not found. Make sure the file is in the correct directory.")
except Exception as e:
    print(f"An error occurred: {e}")
```

```
import os
print(os.getcwd())  # Shows the directory Python is looking for files in
```

```
import os
os.chdir("/Users/raymondlizzolii/Downloads")  # Change to the correct folder

# Verify the change
print(os.getcwd())  # Should now show /Users/raymondalizzolii/Downloads
```

```
import os
print(os.path.expanduser("~"))  # This prints your actual home directory
```

```
import pandas as pd

df = pd.read_csv("exe_test (1).csv")  # No path needed if you're in the right directory
print(df.head())
```

POL 487 (Name searching Code) (Spring 2025)

```
import pandas as pd
from tkinter import Tk
from tkinter.filedialog import askopenfilename

Tk().withdraw()  # Hide the Tkinter root window
file_path = askopenfilename(title="Select CSV File")  # Open file dialog

df = pd.read_csv(file_path)
print(df.head())
```

```
import pandas as pd

# Load the CSV file into a DataFrame
file_name = "exe_test (1).csv"  # Make sure this is the correct file name
df = pd.read_csv(file_name)

#Search the correct part of the name
filtered_df = df[df["EXEC_FULLNAME"].str.contains("Harold Wagner|Leonard H. Lavin",
case=False, na=False)]
# Define the names to keep
names_to_keep = [" Harold A. Wagner","Leonard H. Lavin"]

# make the serch cacesensitive
filtered_df = df[df["EXEC_FULLNAME"].str.contains("Harold Wagner|Leonard H. Lavin",
case=False, na=False)]

# Filter the DataFrame
filtered_df = df[df["EXEC_FULLNAME"].isin(names_to_keep)]

# Display the filtered data
print(filtered_df.head())

# Optionally, save the filtered data to a new CSV file
filtered_df.to_csv("filtered_data.csv", index=False)
```

```
import pandas as pd

# Load the CSV file into a DataFrame
```

POL 487 (Name searching Code) (Spring 2025)

```
file_name = "exe_test (1).csv"  # Make sure this is the correct file name
df = pd.read_csv(file_name)

#Search the correct part of the name
filtered_df = df[df["EXEC_FULLNAME"].str.contains("Harold Wagner|Leonard H. Lavin",
case=False, na=False)]
# Define the names to keep
names_to_keep = [" Harold A. Wagner","Leonard H. Lavin"]

# make the serch cacesensitive
filtered_df = df[df["EXEC_FULLNAME"].str.contains("Harold Wagner|Leonard H. Lavin",
case=False, na=False)]

# Filter the DataFrame
filtered_df = df[df["EXEC_FULLNAME"].isin(names_to_keep)]

# Display the filtered data
print(filtered_df.head())

# Optionally, save the filtered data to a new CSV file
filtered_df.to_csv("filtered_data.csv", index=False)
```

```
import pandas as pd

# Load the CSV file into a DataFrame
file_name = "exe_test (1).csv"  # Make sure this is the correct file name
df = pd.read_csv(file_name)

#Search the correct part of the name
filtered_df = df[df["EXEC_FULLNAME"].str.contains("Harold Wagner|Leonard H. Lavin",
case=False, na=False)]
# Define the names to keep
names_to_keep = ["Harold A. Wagner","Leonard H. Lavin"]

# make the serch cacesensitive
filtered_df = df[df["EXEC_FULLNAME"].str.contains("Harold Wagner|Leonard H. Lavin",
case=False, na=False)]

# Filter the DataFrame
filtered_df = df[df["EXEC_FULLNAME"].isin(names_to_keep)]
```

POL 487 (Name searching Code) (Spring 2025)

```
# Display the filtered data
print(filtered_df.head())

# Optionally, save the filtered data to a new CSV file
filtered_df.to_csv("filtered_data.csv", index=False)
```

```
import os

file_path = "/Users/raymondlizzolii/Downloads/donor_test_4.csv"

if os.path.exists(file_path):
    print("yes file exists!")

else:
    print("NArh? No filefound!")
```

```
import pandas as pd

file_path = r"/Users/raymondlizzolii/Downloads/exe_test (1).csv"
file_path = r"/Users/raymondlizzolii/Downloads/donor_test_2.csv"
file_path = r"/Users/raymondlizzolii/Downloads/donor_test_3.csv"
file_path = r"/Users/raymondlizzolii/Downloads/donor_test_4.csv"

df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/exe_test (1).csv")
print(df.head())
df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/donor_test_2.csv")
print(df.head())
df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/donor_test_3.csv")
print(df.head())
df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/donor_test_4.csv")
print(df.head())
```

```
import pandas as pd  # Import the pandas library

# Define the file path
file_path = r"/Users/raymondlizzolii/Downloads/exe_test (1).csv"  # Update this if needed
file_path = r"/Users/raymondlizzolii/Downloads/donor_test_2.csv"  # Update this if needed
file_path = r"/Users/raymondlizzolii/Downloads/donor_test_3.csv"  # Update this if needed
```

POL 487 (Name searching Code) (Spring 2025)

```
file_path = r"/Users/raymondlizzolii/Downloads/donor_test_4.csv"  # Update this if needed

try:
    df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/exe_test (1).csv")
    df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/donor_test_2.csv")
    df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/donor_test_3.csv")
    df = pd.read_csv(r"/Users/raymondlizzolii/Downloads/donor_test_4.csv")

    print("File loaded successfully!")
    print(df.head())  # Display the first few rows of the dataset
except FileNotFoundError:
    print("Error: File not found. Check the file path and try again.")
except Exception as e:
    print(f"An error occurred: {e}")




import pandas as pd
import os

# List of CSV files to process (you can also use os.listdir() to automatically find all CSVs)
file_names = ["exe_test (1).csv", "donor_test_2.csv", "donor_test_3.csv", "donor_test_4.csv"]  #
Add more file names as needed

# Define the names to keep
names_to_keep = ["Harold A. Wagner", "Leonard H. Lavin"]

# Create an empty list to store DataFrames
filtered_dfs = []

# Loop through each file
for file_name in file_names:
    try:
        # Load the CSV file into a DataFrame
        df = pd.read_csv(file_name)

        # Ensure "EXEC_FULLNAME" column exists in the CSV
        if "EXEC_FULLNAME" in df.columns:
            # Filter the DataFrame based on case-sensitive matching
            filtered_df = df[df["EXEC_FULLNAME"].isin(names_to_keep)]
            filtered_dfs.append(filtered_df)
        else:
            print(f"Column 'EXEC_FULLNAME' not found in {file_name}")
```

```
    except Exception as e:
        print(f"Error processing {file_name}: {e}")

# Combine all filtered DataFrames
final_filtered_df = pd.concat(filtered_dfs, ignore_index=True)

# Display the combined results
print(final_filtered_df.head())

# Save the final filtered data to a new CSV file
final_filtered_df.to_csv("exe_test (1).csv", "donor_test_2.csv", "donor_test_3.csv",
"donor_test_4.csv", index=False)
```