**Me:** I want to be guided step-by-step through completing Task 4 of the Mini Project 2, starting from the beginning until everything is completed and saved.

**AI:** Sure! Let's start by loading the necessary data. You’ll need to open your Python script and load ner\_counts.tsv and NER\_gazetteer.tsv using pandas. Once they’re loaded, the goal is to merge them using the place name field and clean up any missing data to prepare it for mapping.

**Me:** I tried that, but I got an error saying: KeyError: 'placename'.

**AI:** That error usually means that the column you're trying to merge on doesn't exist in one of the dataframes. Can you check the exact column names in each file?

**Me:** The counts file has columns: ['Place', 'Count'] and the coordinates file has: ['placename', 'latitude', 'longitude'].

**AI:** Ah, there it is. The column 'Place' in your counts file doesn’t match the 'placename' column in your coordinates file. Try renaming the 'Place' column to 'placename' before merging.

**Me:** Done. Now when I try to convert the latitude column to float, I get this error: ValueError: could not convert string to float: '31,5167'.

**AI:** That suggests your data uses commas instead of periods for decimal points — a common formatting issue. Replace the commas with periods before attempting the conversion.

**Me:** I’ve replaced commas with periods and now the script runs. I can see the map in HTML, but the script doesn’t print the final confirmation message.

**AI:** Make sure you’ve included the final print() statements and the fig.write\_html() and fig.write\_image() lines. The PNG might also require an additional tool.

**Me:** I do get the HTML map, but the PNG file doesn’t appear.

**AI:** To save the Plotly figure as a PNG, you need the kaleido package. Try installing it using pip install kaleido.

**Me:** Tried that, but I got a permissions error and a message about a file being used by another process.

**AI:** That can happen if a Python process is already using the file or if the installation is corrupt. Try closing IDLE and any running Python scripts. You might also try installing with the --user flag to avoid permission issues.

**Me:** I still couldn’t install it. I took a screenshot of the map from the HTML version.

**AI:** That works too. When direct export fails, screenshots are acceptable for submission. Just make sure the image is clear and cropped properly.

**Me:** I created a second version of the map using Plotly’s density\_mapbox instead of scatter\_geo. It looks much better.

**AI:** That’s a great enhancement. Save that version separately — maybe name it task4\_density\_heatmap.html — and mention it as a bonus or alternative visualization.

**Me:** I faced more issues trying to reinstall kaleido. I even restarted my machine, but it still shows locked file errors.

**AI:** That indicates the Python environment has a corrupted kaleido install. For now, stick to the HTML map and the screenshot image. You’ve already captured the output successfully.

**Me:** So the HTML file and screenshot will be enough?

**AI:** Yes, they’re perfectly valid for submission.

**Me:** What about the README file? What should it include?

**AI:** The README should describe what Task 4 is about — that you extracted place names using both regex + gazetteer and spaCy’s NER. You should briefly describe each method, the challenges faced, the results, and include a comparison.

**Me:** Can you help me write it?

**AI:** Sure. Here’s a simplified, clear version:

Task 4 was about extracting and mapping places mentioned in a January 2024 news article. Two techniques were used. First, regular expressions were used to extract place names from text, and a gazetteer was used to find their coordinates. The results were mapped using Plotly, producing an HTML map and an optional PNG. Second, spaCy’s NER tool was used to find entities like cities or countries. These entities were visualized in an HTML file using displaCy. Both methods worked, but regex gave more accurate mapping while spaCy detected more names overall. Together, they provided a complete picture of location references in the article.

**Me:** I also need to compare the maps more clearly.

**AI:** Here's a paragraph for that:

The regex-based map gave a precise, visually appealing result by matching exact place names from the gazetteer and plotting them with size indicators. However, it missed some locations not listed or formatted differently. The spaCy-based method found a wider range of named entities, including less obvious locations, but couldn’t be mapped directly as it did not include coordinates. Instead, it highlighted place names in the text using color coding. Each approach had strengths: regex was good for mapping, and spaCy for thorough entity detection.

**Me:** What files do I need to submit?

**AI:** Your final submission should include:

* regex\_map.py – main script for regex + map
* task4\_ner\_map.html – map output file
* task4\_ner\_map.png – (if available) or a screenshot instead
* task4b\_visualize\_ner.py – script for spaCy NER visualization
* task4b\_ner\_visual.html – output from spaCy’s displaCy
* task4\_density\_heatmap.html – optional, bonus heatmap
* ner\_counts.tsv – regex output data
* NER\_gazetteer.tsv – coordinates for mapping
* README.md – your summary and reflection document

**Me:** I got everything except the PNG.

**AI:** Then submit the screenshot version with a note in your README explaining that PNG export was attempted but didn’t work due to kaleido issues.