**AI Documentation For DH Mini Project 2**

**Conversation 1**

You said  
I’m a student doing a python project in DH course. Act as a good tutor and give me advice and teach me without giving me the full answers.

I’m trying to clean up place names, and I want to remove “The” when it appears at the start—like turning “The Netherlands” into just “Netherlands.” I wrote this line:

python

place = re.sub("^the ", "", place)

But it doesn’t seem to work with “The Netherlands”.

ChatGPT said:  
Good attempt! The logic is right, but the issue is that regex is **case-sensitive by default**, so "^the " won’t match "The " with a capital T.

To fix that, just add the flags=re.IGNORECASE argument so it becomes case-insensitive:

python

CopyEdit

place = re.sub(r"^the\s+", "", place, flags=re.IGNORECASE)

Also, using \s+ instead of a single space ensures it catches any extra whitespace too.

**Conversation 2**

You said  
I’m trying to write the contents of normalized\_places to a .tsv file, one row per place name. So I added this part:

python

for entity, count in normalized\_places:

row = f"{entity}\t{count}\n"

But it throws an error.

ChatGPT said:  
Ah, I see the issue. When you loop over a dictionary directly like that—for entity, count in normalized\_places:—Python expects each item to be a tuple. But normalized\_places by itself only gives you the **keys**, not the key-value pairs.

Try using .items() to get both the name and frequency like this:

python

for entity, count in normalized\_places.items():

You said  
Right! That fixed the error. Now I’m writing the rows with:

python

row = "{entity}\t{count}\n"

But when I open the file, it literally says {entity} and {count} instead of the values.

ChatGPT  
Classic mistake—happens to everyone! You're using regular quotes instead of an **f-string**, which means Python won’t evaluate the variables inside the string. Update that line to:

python

row = f"{entity}\t{count}\n"

That little f before the string is what tells Python to insert the variable values.

**Conversation 3**

You said  
I’m trying to take all the place names from my ner\_counts.tsv file, look up their coordinates, and write them to a new file called ner\_gazetteer.tsv. I wrote this part:

python

coordinates = get\_coordinates(name)

if coordinates:

lat = coordinates['latitude']

lon = coordinates['longitude']

out\_file.write(f"{name}\t{lat}\t{lon}\n")

else:

out\_file.write(f"{name}\tNA\tNA\n")

But I’m getting an error that says 'NoneType' object is not subscriptable. Not sure what’s wrong.

ChatGPT said  
That error usually means get\_coordinates(name) is returning None, and you're trying to access it like a dictionary (coordinates['latitude']). You're checking if coordinates, which is good, but maybe get\_coordinates() doesn't return None—it might return something else like False, or maybe it returns a tuple instead of a dictionary?

Can you show me or check what the get\_coordinates() function returns when a location isn’t found?

You said  
 I printed it out, turns out it's returning None for some places that aren’t found.

ChatGPT said  
Okay, so the logic is mostly good, but to be extra safe, you might want to explicitly check:

python

if coordinates is not None:

Also, make sure you’re not assuming 'latitude' and 'longitude' exist unless you’re really getting a dictionary.

You said  
I changed it to:

python

if coordinates is not None:

lat = coordinates['latitude']

lon = coordinates['longitude']

Now it works for most places! But for a few, I get a KeyError: 'latitude'.

ChatGPT said  
Ah—that means get\_coordinates() returned something, but maybe not with the expected keys. You could guard against that by using .get():

python

lat = coordinates.get('latitude', 'NA')

lon = coordinates.get('longitude', 'NA')

Then, even if the key’s missing, it’ll write “NA”

**Conversation 4**

You said  
I’m trying to merge the ner\_counts DataFrame with the gazetteer one on the "name" column, so I did:

python

merged\_df = pd.merge(ner\_counts, gazetteer, on="name")

But it throws a KeyError: 'name'. What am I doing wrong?

ChatGPT said  
That error means at least one of your DataFrames—ner\_counts or gazetteer—doesn’t have a column called "name". Could you check the column names using ner\_counts.columns and gazetteer.columns?

You said  
Oh… one of them has "Name" with a capital N, and the other has "name" lowercase.

ChatGPT said  
There it is! Column names are case-sensitive in pandas. You can either rename the column in one of the DataFrames like this:

python

gazetteer.rename(columns={"Name": "name"}, inplace=True)

Then your merge line will work just as you wrote it.

You said  
Got it. I renamed the column, and now the merge works. But I also noticed some rows are missing from the result.

ChatGPT said  
By default, pd.merge() does an **inner join**, meaning it only keeps rows that match in both datasets. If you want to keep all entries from ner\_counts, even if they don't have coordinates, try:

python

pd.merge(ner\_counts, gazetteer, on="name", how="left")

That way, unmatched rows will still appear—with NaN for the coordinates.