Geolocation for mapping

November 22, 2024

[3]:
```
import io
import requests

url = "https://cocl.us/Geospatial_data"
lat_long = requests.get(url).text
lat_long_df=pd.read_csv(io.StringIO(lat_long))
lat_long_df
```

[3]:

	Postal Code	Latitude	Longitude
0	M1B	43.806686	-79.194353
1	M1C	43.784535	-79.160497
2	M1E	43.763573	-79.188711
3	M1G	43.770992	-79.216917
4	M1H	43.773136	-79.239476
..	…	…	…
98	M9N	43.706876	-79.518188
99	M9P	43.696319	-79.532242
100	M9R	43.688905	-79.554724
101	M9V	43.739416	-79.588437
102	M9W	43.706748	-79.594054

[103 rows x 3 columns]

[4]:
```
import pandas as pd

# Step 1: Read the z_score_cleaned_data_rounded.csv file
z_score_path = "z_score_cleaned_data_rounded.csv"
z_score_df = pd.read_csv(z_score_path)

# Step 2: Rename 'postal_code' to 'Postal Code' in the existing DataFrame
z_score_df.rename(columns={"postal_code": "Postal Code"}, inplace=True)

# Step 3: Standardize postal code formats in both DataFrames
z_score_df["Postal Code"] = z_score_df["Postal Code"].str.strip().str.upper()
lat_long_df["Postal Code"] = lat_long_df["Postal Code"].str.strip().str.upper()

# Step 4: Merge the DataFrames on 'Postal Code'
```

```
merged_df = pd.merge(z_score_df, lat_long_df, on="Postal Code", how="left")

# Step 5: Save the merged DataFrame to the same CSV file (optional)
output_path = "z_score_cleaned_data_rounded.csv"  # Keep it the same as the␣
 ↪input file
merged_df.to_csv(output_path, index=False)

print(f"Merge complete! Updated data saved to {output_path}")
```

```
Merge complete! Updated data saved to z_score_cleaned_data_rounded.csv
```

[ ]: