**Documentation for Berlin Electric Charging Station Demand Analysis**

**Program Structure**

**1. Purpose**

The program visualizes the demand for electric charging stations in Berlin using geospatial data. It combines datasets for population size and charging station distribution by postal code (PLZ) to identify areas with high demand for additional charging stations.

**2. Program Workflow**

1. **Import Required Modules**:
   * pandas: For data manipulation.
   * core.methods: Contains processing and visualization functions.
   * core.HelperTools: Includes utility functions like a timer decorator.
   * config.pdict: Stores configuration variables such as file paths.
2. **Core Workflow (main.py)**:
   * **Data Loading**:
     + Geographical data (file\_geodat\_plz) is loaded for postal codes.
     + Charging station data (file\_lstations) and resident data (file\_residents) are imported.
   * **Data Processing**:
     + Charging station data is preprocessed and analyzed to count the number of stations per postal code (preprop\_lstat and count\_plz\_occurrences).
     + Resident data is processed similarly to prepare for visualization (preprop\_resid).
   * **Visualization**:
     + The processed data is passed to a method (make\_streamlit\_electric\_Charging\_resid) to generate a Streamlit app for geospatial visualization.
3. **Execution**:
   * The main() function orchestrates the entire process. The program measures execution time using the @ht.timer decorator.

**3. Key Methods**

**From methods.py:**

* **preprop\_lstat(df\_lstat, df\_geodat\_plz, pdict)**:
  + Preprocesses the charging station data (e.g., cleansing, merging with geographical data).
* **count\_plz\_occurrences(df\_lstat2)**:
  + Aggregates the number of charging stations per postal code.
* **preprop\_resid(df\_residents, df\_geodat\_plz, pdict)**:
  + Preprocesses resident data, aligning it with geographical postal code data.
* **make\_streamlit\_electric\_Charging\_resid(gdf\_lstat3, gdf\_residents2)**:
  + Generates a Streamlit app with geospatial heatmaps visualizing charging stations and resident data.

**From HelperTools.py:**

* **@timer**:
  + A decorator function to log the execution time of the main() function.

**Interpretation of Results**

**1. Data Sources**

* **Charging Station Data**:
  + Number of charging stations per postal code (PLZ).
  + Essential for identifying supply levels.
* **Resident Data**:
  + Population distribution across postal codes.
  + Indicates potential demand for charging stations.

**2. Visualization Insights**

The final visualization consists of geospatial heatmaps overlaid on a Berlin map:

* **High-Demand Areas**:
  + Postal codes with a high population and a low number of charging stations appear as areas of high demand.
* **Low-Demand Areas**:
  + Regions with fewer residents or a high number of charging stations show less demand.

**3. Observations**

* **Urban Areas**:
  + Likely to have higher demand due to dense populations and limited private charging solutions.
* **Suburban Areas**:
  + Typically show lower demand as residents may have personal charging setups.