Geospatial analysis of the distribution of environmental and social status indicators at district level for the City of Hamburg

1 Introduction

This script uses python to analyse and classify different environmental and social status indicators to assess questions in relation to the topic of environmental justice within the city of Hamburg, Germany. The analysis is informed by the debate on environmental justice. The concept of "environmental justice" examines the relationship between the exposure to environmental hazards or the quality of the environment and social structure, focusing primarily on distributional issues and the negative impacts of environmental burdens (State of Berlin, 2023). A disproportional high exposure to environmental hazards or low quality of the environment in relatively poor or socially vulnerable areas would be an example of environmental injustice. (Paul Mohai & Pellow David Roberts, 2009).

The python code calculates statistics for four core indicators: Social Status, Green Area Supply, Noise Pollution and Thermal Burden. It explores at first the spatial distribution of the individual indicators and compares these across the different districts for Hamburg. In a second step, through a combination of the individual indicators, the code aims to explore the relation between social status and environmental conditions. Level of spatial analysis are the districts of the City of Hamburg. To visualize the result, the code creates and saves 5 choropleth maps, detailed statistics, and graphs for the visualization of the distributions of the different indicators. The outputs will be automatically saved to the “output” folder available on GitHub when running the script.

2 Set-Up / Installation

The script is made available on GitHub. The GitHub repository with the name “egm722_project” can be accessed under the following link: https://github.com/CharlotteGIS/egm722_project . The repository contains a Jupyter Notebook with the code. The data files used in the script are available in the “data” folder on GitHub.

The steps to set up the environment and to run the code are:

1. Fork the repository. To fork the repository, click on the 'Fork' button at the top right corner of the screen.
2. Clone the repository to create a local version on your computer.
3. Conda environment: To use the script, it is recommended to install conda and to set up a dedicated project environment.
4. Install required packages and dependencies: To recreate the project environment and automatically install all the required packages and dependencies, import the **environment.yml** file available in the repository. Make sure to change the name provided in the *yml -file* to the name of your environment. The packages and dependencies available in the yml-file are:

- Python
- Geopandas
- Cartopy>=0.21
- Notebook
- Folium
- Rasterio
- Pyepsg

The current script doesn't require the use of Rasterio or Folium. These packages have however been installed with the aim to extend the script further in the future. The code has been designed using Jupyter Notebook. To run the code the setup of an Integrated Development Environment (IDE) such as PyCharm is not required. Further explanations of the functions used in the script and the different steps performed are provided in the script itself.

2.1 Running the code

The code can be executed using the data files available in the data folder in Jupyter Notebook. For the noise data, the file "Laermkarten.zip" first needs to be *unzipped* after the download to be able to read the data file. The other data files can be used as such. The user can make some changes to the title of the maps or colour palette used as well as the columns used for classification. When doing so, the change of columns and column names must be reflected throughout the script for the script to work. When using different data files from the files provided in the "data" folder, naming conventions in the script need to be updated.

Table 1: Packages & Python Libraries used in the script

Packages	**Function**
Geopandas	Working with spatial data
Matplotlib	Plotting and visualizations

Cartopy	Creating visualisations
Libraries / Modules imported (not named above)	
SciPy	Used for the calculation of the z-score to normalize the data
Contextly	Adding basemap to the maps
Pandas	To perform calculations

3 Methodology

The script analyses four environmental and one social status indicator. The overall methodological workflow has been informed by a recent publication on Environmental Justice published by the city of Berlin for the year 2021/2022 (State of Berlin, 2023). The methodology has been adapted to the specific use case for Hamburg and considering the available data sources.

3.1 Data sources

The following data sources have been used.

Table 2: Date sources used in the script

Name	Description	Source
Indicator: Social Status		
Stadtteile_Hamburg.shp	Shapefile with the administrative boundaries of the districts for Hamburg	ESRI Data Hub
Statistics_HH21.csv	Data table with social statistics for the year 2021 broken down by district	Statistisches Amt für Hamburg und Schleswig-Holstein (Statistical Office for the Federal States Hamburg and Schleswig Holstein
Indicator: Green Area Supply		
Oeffentliche Gruenanalge_Hamburg.shp	Shapefile with publicly maintained green areas and parks for Hamburg	ESRI Data Hub
Statistics_HH_21.csv	Integrated information about the number of inhabitants per district for the calculation of statistics	See above
Indicator: Noise Pollution		
Residential_buildings_HH.shp	A shapefile with residential buildings in Hamburg	ESRI Data Hub
Laermkarten_HH_2018-11-19.zip	A shapefile of noise maps for Hamburg. The data is a classified shapefile distinguishing between 4 noise classes in relation to the noise level. The noise level is determined considering noise from e.g., streets, rail.	City of Hamburg
Indicator: Thermal Burden		
• Temp_stats.csv • Landsat 8, Band10 (Date of acquisition: 2022-17-07)	Table containing the output of zonal statistics for Band10 of a Landsat image. The statistics have been derived from Band 10. For the Landsat image, brightness temperature has been calculated.	• USGS – Earth Explorer (Landsat Image) • Own calculation of Radiance and Brightness Temperature in QGIS using Band 10.

3.2 Methods

The script begins by importing all the required libraries and by loading the required data. Next, all the functions used in the script are defined. Detailed explanation of the functions, the parameters and return are provided in the Docstrings in the Notebook. The functions are:

- A function to reproject the data to the same coordinate system (EPGS:25832) for Hamburg. Function name: ***<reproject_to_local_epsg>.***
- A function to classify the values of one single column into 5 equal percentiles using pandas *pd.qcut* command. Function name: ***percentile_5.***
- A function to classify the sum of the values of multiple columns into 5 equal percentiles using pandas pd.*qcut* command. Function name: ***<percentile_multi>.***
- A function to create a text column called *status* and assign text labels from very low to very high based on the values of the classified column, output of the percentile_5 function. Function name: ***<add_status_column>.***
- A function to produce maps visualizing the percentile ranking of the indicators based on the status column and save them to the output folder. Function name: ***<plot_stats>*.**

The script first analysis each of the 4 indicators individually and groups the results by district. The indicators are then re-classified into 5 equal percentiles and assigned values from 0-4, whereas 0 represents the lowest 20th percentile, and 4 the highest 80th percentile of the mean. Subsequently, the values are then allotted the corresponding labels [“very low ”, “low”, “medium”, “high”, “very high”]. The text values are assigned based on the results of the classification. This is followed by the combined analysis of the individual indicators. The combined analysis first aggregates the 3 environmental indicators (noise, green area supply and thermal burden) and classifies the sum into 5 equally percentiles to derive comprehensive statistics about the multiple environmental burden per district. In a last step, this is combined with the results of the percentile ranking of the social status indicator to derive an integrated view of environmental burdens and social status. The results are equally classified into 5 equal percentiles.

Prior to any geospatial operation, all of the data sources are reprojected into the same coordinate system using the *reproject_to_local_espg* function.

3.2.1 Social Status Indicator

For analysis of the social status indicator, the script uses the shapefile with the administrative boundaries and a file in csv format with social statistics for Hamburg. To determine the social status, four social statistics available in the csv- file were considered. All represent a degree of social vulnerability. The script considers the values with equal weighting. These are:

- *%unemployed* : % of unemployed population per district
- *%social_benefits*: % of population receiving social benefits per district
- *%social_housing*: % of population in social housing per district
- *%unemployed>65*: % of unemployed population older than 65 per district.

The code first cleans and subsets the dataframe so that only the relevant columns remain. The column names are further translated to English. To normalize the data, the SciPy Library is used on the four columns mentioned above to calculate the z-score. For the social indicator, the *percentile_multi* function is applied to classify the data. It takes as input the sum of the four variables defined above after these have been normalized. Subsequently, a status column is added by applying the *add_status_column* function. Finally, the social statistics are merged with the administrative boundary dataset to enable plotting and map creation.

3.3 Green Area Supply

Access to green areas is considered a positive and potentially mitigating factor. Data source is the shapefile with publicly maintained green areas for Hamburg. The script first filters the data to remove irrelevant columns according to the ALKIS convention (Cadastre Information System, Germany). This is followed by the calculation of different statistics. These are:

- The totally available area (ha) of greenspace per district (*green_area_total_ha*) and the mean area (ha) available *(green_area_mean_ha)*
- The count of green areas per district (*green_space_count)*
- The percentage of green areas of the total area per district in Sqm (*perc_green_area*)
- The available green area per inhabitant and district (*area_per_inhbt).* The information on the inhabitants per district are derived from the administrative boundary dataset.

To determine the indicator, the available green area per inhabitant (*area_per_inhbt*) is used as the central metric and normalized by calculating the z-score. The classification is performed by using the *percentile_5* function which takes 1 variable as input using the *area_per_inhabitant* column. Status description in text format is added by applying the *add_status_column* function. Finally, the statistics for green spaces are merged with the admin data set to add geometry information and allow map creation.

3.4 Noise pollution

Date sources used for this indicator are the housing data set and the noise data set for Hamburg. For the noise indicator, the code aims to identify the area of houses affected by noise and noiseclass per district. The noise data is a classified polygon layer with different noise classes form 0 (low) to 4 (high). To assess noise pollution per district, the code determines the number of affected houses and of affected area per noiseclass. After the data is filtered to remove irrelevant columns and relevant column names are translated into English, the script then applies the *overlay method* to intersect the housing layer and the noise layer. Unnecessary columns were dropped from the dataset. To prepare for plotting the intersected layer is joined with the admin dataset. The code then calculates the following statistics:

- Number of affected houses per noiseclass and district
- The area (*total_area*) of affected houses by multiplying the housing area with the number of houses per district
- The total affected area of residential buildings (*area floors*) by multiplying the *total_area* with the number of floors per house and per district converted into Sqm.

To take into consideration the differing severity of impact depending on the noiseclass, the script then assigns weights: {0: 0.5, 1: 1, 2: 1.5, 3: 2, 4: 2.5} to each noiseclass in view of severity from low to high with an increase by 0.5. The weights are then mapped to the noise affected housing area (*weigthed_area)*. Finally, the *percentile_5* and the *add_status_column* functions are applied to classify the data. The classification is performed on the *weighted_area* column.

3.5 Thermal Burden

The script uses a greatly simplified calculation to derive Land Surface Temperature following a tutorial conducted by the NASA ARSET Training program (NASA, 2022). Pre-processing was done outside of Python in QGIS on a Landsat Image. The Landsat B10 Surface Temperature Band was rescaled and then converted from Kelvin into Celsius. The formula used is as follows :

$$LST\_C \Rightarrow B10(Landsat8) * 0.00341802 + 149 - 273.15$$

Zonal Statistics were than calculated. The output is represented in the *temp_stats* file used in this script. The script applies the *percentile_5* and the *add_status_column* functions to classify the data. Classification is performed on the *MEDIAN* column representing the median value per district . The classified data is then merged to the admin dataset.

3.6 Integrated Environmental Burden and Social status Index

Using the classified columns of the individual indicators, the code subsequently calculates overall statistics of multiple environmental burdens based on the sum of the percentile ranking of the 3 environmental indicators per district. To enable calculations, the script first converts the categorial values to integers. All indicators are considered with equal weighting. The output is saved in the column *env_multiple_burden.* Finally, the classified social status indicator is added to the sum of 3 environmental indicators to receive combined statistics of social status and environmental burdens (*combined_indicators)*. Like for the individual indicators, the script classifies the sum operations for both into 5 percentiles using pandas *pd.qcut* and the *add_status_column* function.

3.7 Visualization

All the map visualizations created by the code are based on the *status* column , derived from the classification of the data into equal percentiles by applying the *plot_stats* function. See the Docstrings in the notebook for an explanation on how to use the function. In addition, the code calculated different plots to visualize the spatial distribution of the indicators.

4 Expected Results

The script will produce different outputs. Statistics with the calculations of the indicators are saved to the output folder in csv format. The files include:

- *stats_merged.csv*: A csv file with the combined statistics calculated across the individual indicators and the combined analysis of the indicators. In addition to the results of the classification of indicators, the file equally includes further statistics calculated for the individual indicators not used in the visualization to provide additional context and information.
- The files*: green_stats*.csv, *noise_stats.csv* and *social_stats.csv* contain only the statistics specific to the relevant indicator.

The script produces 5 choropleth maps and saves them in the “output” folder. These maps include one map for each of the individual indicator as well as a map to visualize the results of the combined analysis. The outputs are presented on the following pages.

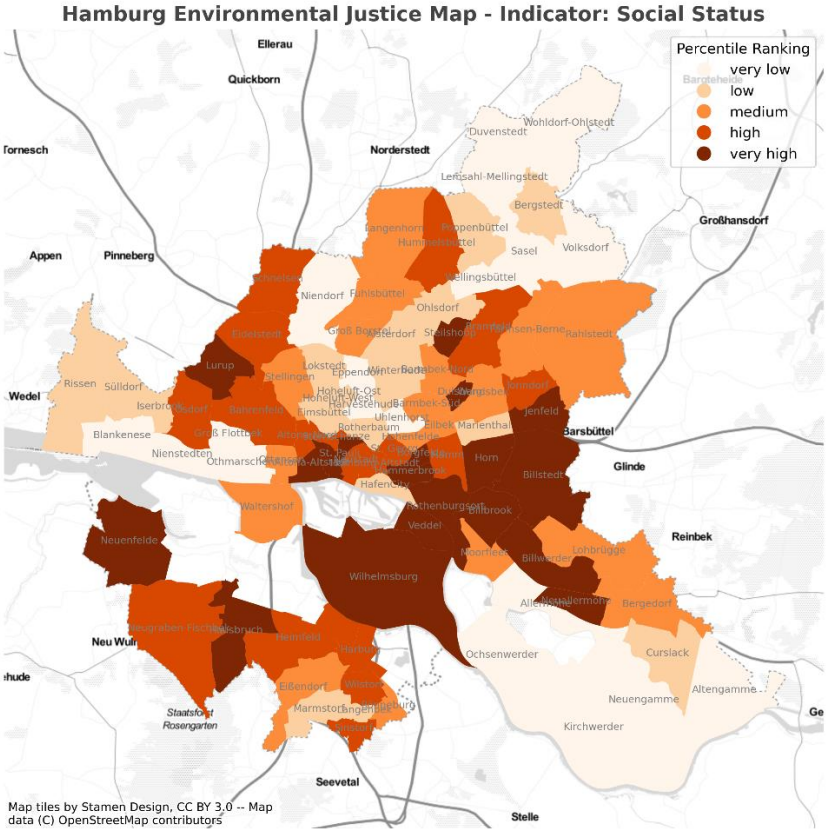

Figure 1: Map - Social Status. The map visualizes the results of the social status indicator, classified by percentiles. The classification is based on the following individual statistics of: % of unemployed population, % of population receiving social benefits, % of population in social housing and % of unemployed population older than 65 per district.

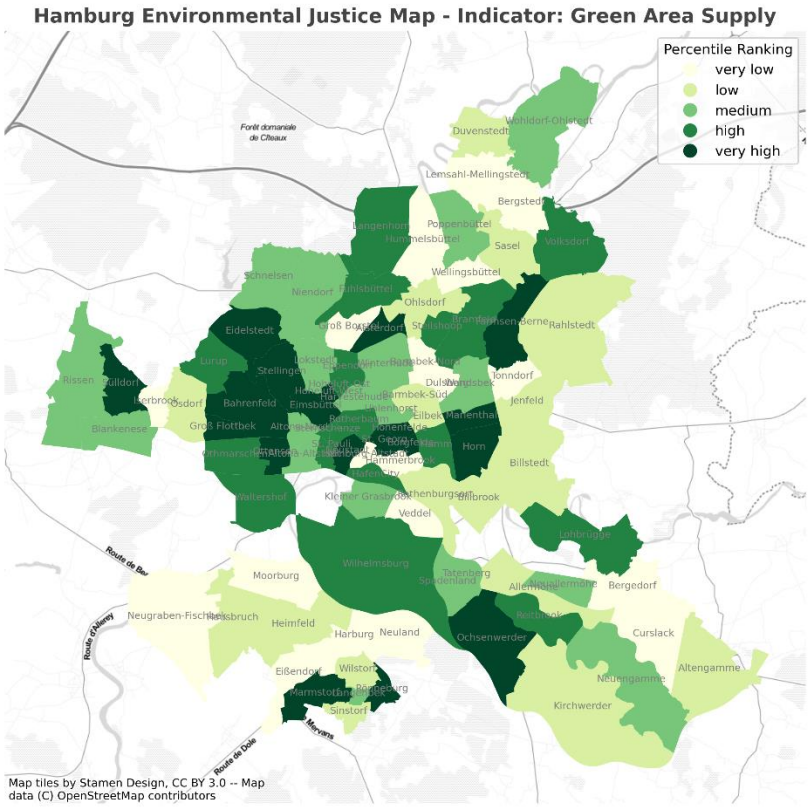

Figure 2 : Map – Green Area Supply. The map visualizes the results of the Green Area Supply Indicator, classified by percentiles. The classification is based on statistics of "green areas / inhabitant in Sqm per district.

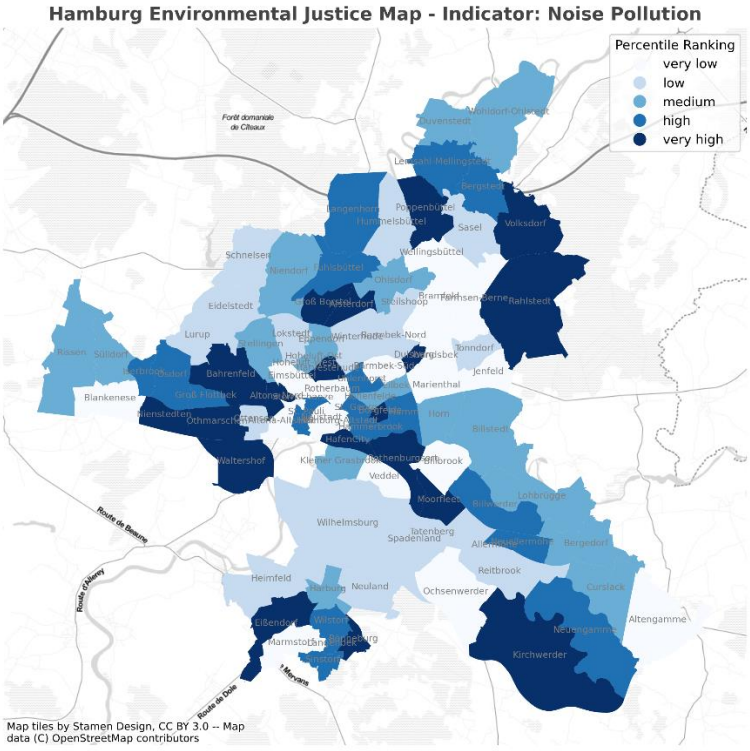

Figure 3 : Map – Noise Pollution. The map visualizes the results of the Noise Indicator, classified by percentiles. The classification is based on the weighted residential housing area affected by noise class per district. The housing area affected was weighted by noise class in view of the severity.

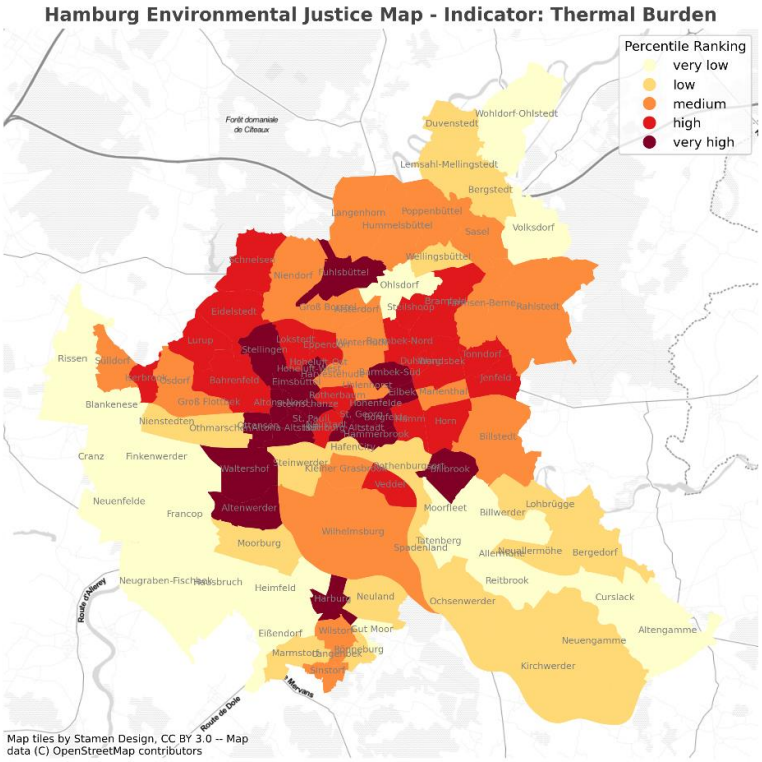

Figure 4: Map – Thermal Burden. The map visualizes the results of the Thermal Burden Indicator, classified by percentiles. The classification is based on statistics of the mean brightness temperature per district.

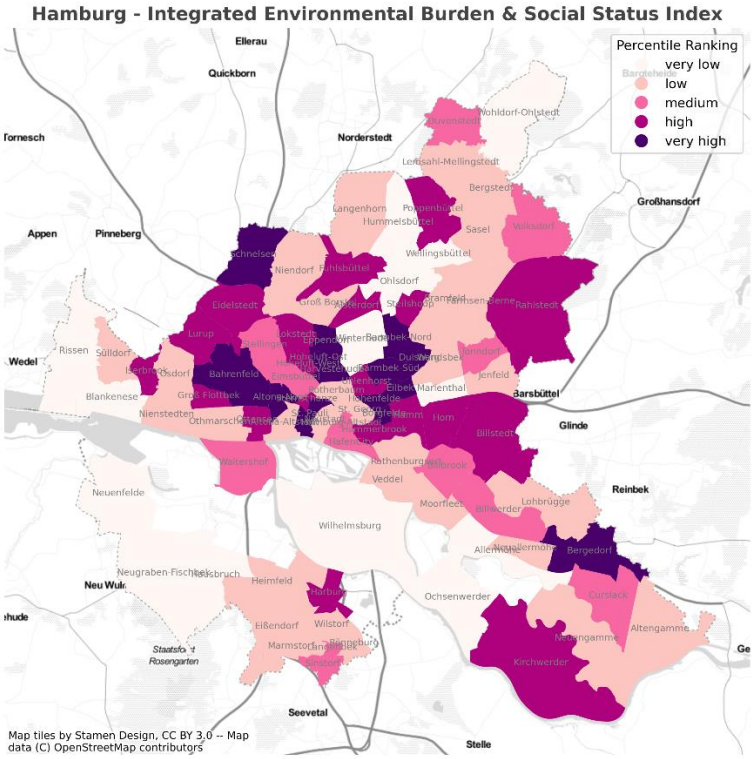

Figure 5: Map – Integrated Environmental Burden & Social Status index. The map visualizes the results of the combination of the individual indicators, classified by percentiles. The classification is based on sum of the environmental burdens (green area supply / inhabitant, noise pollution , thermal burden) and the social status per district.

Finally, the script produces a plot showing the number of counts of the different percentile rankings for each of the four individual indicators The plot is saved in the “output” folder.

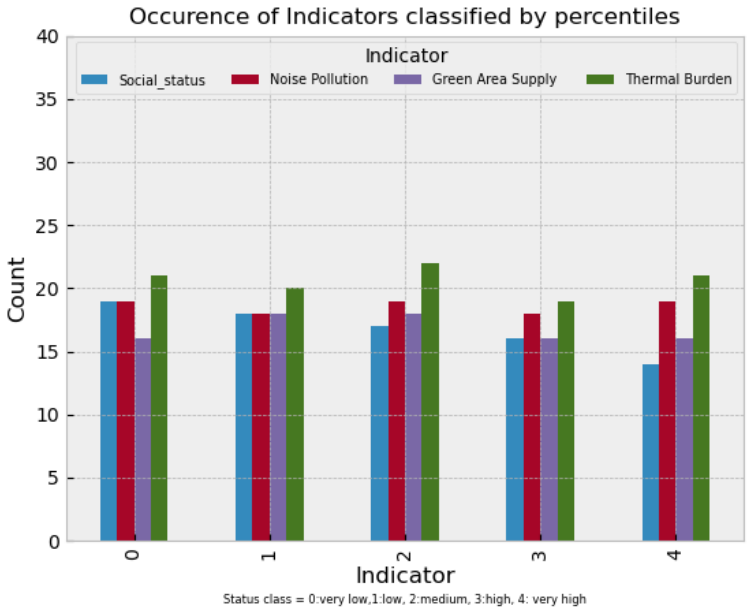

Figure 6: Occurrence Plot - Number of counts the four status classes per indicator

A correlation matrix of the main indicators. The correlation matrix is not saved automatically in the output folder.

	weighted_area_percentile	qt_soc_stats	z_area_per_inhbt	MEDIAN_percentile	%unemployed	%social_benefits	env_multiple_burd
weighted_area_percentile	1.000000	-0.027378	-0.306882	-0.184897	-0.111438	-0.052056	
qt_soc_stats	-0.027378	1.000000	-0.077380	0.396643	0.932606	0.917629	
z_area_per_inhbt	-0.306882	-0.077380	1.000000	0.339968	-0.034500	-0.088439	
MEDIAN_percentile	-0.184897	0.396643	0.339968	1.000000	0.382876	0.222293	
%unemployed	-0.111438	0.932606	-0.034500	0.382876	1.000000	0.955546	
%social_benefits	-0.052056	0.917629	-0.088439	0.222293	0.955546	1.000000	
env_multiple_burden_percentile	-0.082469	0.027421	0.102558	0.108422	-0.007721	0.002205	

Figure 7: Correlation matrix - selected statistics

The scatter plot is derived by using the environmental burden statistics and the social statistics and saved.

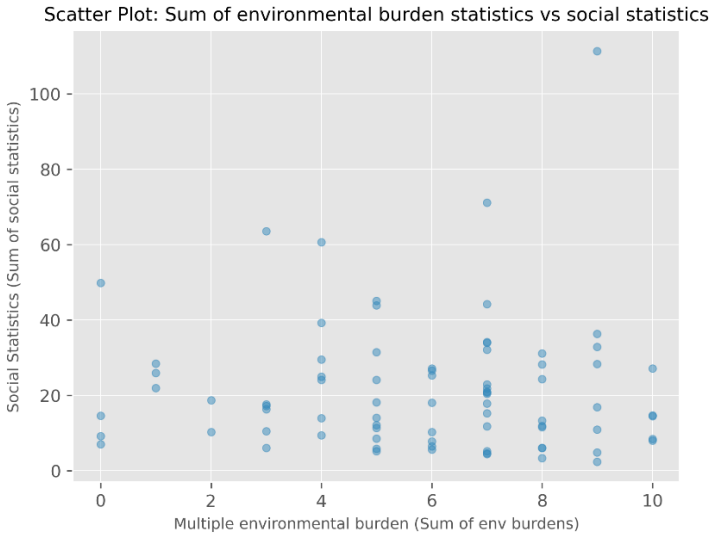

Figure 8: Scatter Plot : Social Statistics vs sum of environmental burden (sum of noise, pollution, thermal indicator)

To quickly visually assess the status across all indicators a cleaned and further reduced dataframe grouping the status columns of the individual indicators and the status ranking of the combined analysis is available in the *status_ counts* dataframe.

	status_noise					status_greens					status_soc				
	high	low	medium	very high	very low	high	low	medium	very high	very low	high	low	medium	very high	very low
district															
Allermöhe	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
Alsterdorf	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Altengamme	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
Altona-Altstadt	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Altona-Nord	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0

Figure 9: Status counts grouped by district (extract)

5 Troubleshooting

Some troubleshooting options for possible issues that might arise are described here.

- Reading of csv files: The script reads csv files. Issues with reading the csv files might arise when a different regional setting is used for the "separators" in excel. Update the code specifying the delimiter used for reading the files when loading the data. Issues with the csv files might further arise due to different language settings in excel and treatment of special characters such as “ä”, “ö” not available in the English language. This issue might arise with the “statistics_HH21.csv” file which contains several special characters in its original version. The script renames these columns, but the appearance of these column names might slightly differ when reading the data. To troubleshoot, make sure to either change the setting, or adapt the column names referred to.
- An error will occur, if the noise data: *Laermkarten* is not unzipped prior to reading the data. The data need to be unzipped and the shapefile with its extensions stored in the “data” folder.
- Some of the files processed in the script are quite heavy in size and the whole code will most likely run for several minutes (5-10 minutes) depending on the capacity of the computer used. In case the script hangs, Restart the Kernel and run again. It is not recommended to “*out_comment*” the housing data due to the size of the file.
- Case sensitivity, naming conventions and naming error: An error might occur after columns have been dropped from the script and only specific cells are re-run. To troubleshoot, it is recommended to re-load the data. Further, naming conventions in the script need to be observed.

- In case of issues with the functions, call the help function to access the Docstring.
- To launch the notebook, make sure that the project environment is activated. The code will otherwise not work.
- Errors will occur if the designated order of the script isn't observed. For example, the *plot_stats* function requires the existence of a categorial column called *status* derived from the classification of the data.

6 References

NASA, A. (. R. S. T., 2022. *NASA - ARSET - Satellite Remote Sensing for Measuring Urban Heat Islands and Constructing Heat Vulnerability Indices, Part 3 (Integrating Socioeconomic Data with Satellite Imagery for Constructing Heat Vulnerability).* [Online] Available at: https://appliedsciences.nasa.gov/join mission/training/english/arset-satellite-remote-sensing-measuring-urban-heat-islands-and [Accessed: May 2023].

Paul Mohai & Pellow David Roberts, J., 2009. Environmental Justice. *Annual Review of Environment and Resources,* 34(1), pp. 405-430.

State of Berlin, G., 2023. *Senate Department for Mobility, Transport, Climate protection and the Environment, Berlin.* [Online] Available at: https://www.berlin.de/sen/uvk/en/environment/sustainability/environmental-justice/ [Accessed on 15.04.2023 April 2023].