TROUBLESHOOTING

The user may experience errors when running the scripts. If these occur when running a function, the user is advised to show the function docstring by calling help(), the docstring will explain the required inputs and data types. See below for more information on some errors the user may encounter and how to fix them.

Error	**Description/Solution**
IndexError: list index out of range	One (or more) of the lists within the script does not contain the correct number of values. Solution: check and update all lists to the correct number of unique values. If the number of unique values is unknown the user can run the following code adding the appropriate GeoDataFrame and column label. num_uv = len(<*GeoDataFrame*>.<*Column*>.unique()) print(num_uv)
KeyError:	This error message occurs when the wrong column label is being used, a column label list is of an incorrect length or two GeoDataFrames used within a function have the same column label for their uniquely identifying naming column. Solution: inspect and edit column labels in script and, if needed, rename column labels.
NameError: <> is not defined	A variable has not been defined such as the buffer distance or a module has not been imported. Solution: inspect script and ensure variables have been specified and all modules have been imported under their desired aliases.
ValueError: Invalid RGBA argument	An incorrect colour value is being used. Solution: colour names must be from the matplotlib colour list found here.
Data plotting outside study extent in Locator_map.py	The user's input data is not to the study extent. Solution: repeat line 198 in the code to clip the new input data to the extent of the study.
Legend plotting over map in Locator_map.py	Solution: the user should edit the figure size or extent of the map in lines 165 and 195.
Site_characteristics.py ZeroDivisionError: float division by zero	This error occurs specifically when running the find percentage underlying raster values function when the sites GeoDataFrame has not been transformed to the same CRS as the input raster. Solution: ensure that the sites GeoDataFrame is transformed to the raster CRS before running the function.
Site_characteristics.py ValueError: zero-size array	This error occurs specifically when running the find underlying raster statistics function when the sites GeoDataFrame has not been transformed to the same CRS as the input raster. Solution: ensure that the sites GeoDataFrame is transformed to the raster CRS before running the function.
Interactive_map.ipynb map not plotting to study area when creating folium map	The GeoDataFrame to find the central location has an inappropriate CRS. Solution: The GeoDataFrame CRS to find the central location must have lat/long coordinates. Transform the GeoDataFrame using the following code: <*GeoDataFrame*>.to_crs(epsg=4326, inplace=True).
Interactive_map.ipynb FileNotFoundError	This error occurs when the user tries to run the Interactive_map.ipynb without running the Proxomity_analysis.py and Site_characteristics.py scripts first. Solution: The scripts should be run in the correct order.
CRS issues	Solution: Naive geometries should not be used within the script and all data should have the same CRS to enable merges and plotting of data. Solution: To set a CRS for a naive geometry the user should use <*GeoDataFrame*>.set_crs(epsg=<*ESPG Code*>, inplace=True), the user should set all data to the same CRS by using <*GeoDataFrame*>.to_crs(epsg=<*ESPG Code*>, inplace=True). See here for a list of EPSG codes.