SESAME Documentation

SESAME software tools for integrating Human - Earth System data

Abdullah Al Faisal[1,*], Maxwell Kaye[2], Maimoonah Ahmed[1], Eric Galbraith[1]

[1]Department of Earth and Planetary Sciences, McGill University, Montreal, Québec, H3A 0E8, Canada.
[2]Department of Mathematics and Statistics, McGill University, Montreal, Québec, H3A 2K6, Canada.

Corresponding author: Abdullah Al Faisal (abdullah-al.faisal@mail.mcgill.ca)

SESAME documentation and user manual

This section describes the functions in the initial release of SESAME. For each function, we give a description of the input arguments, the expected output, and sometimes usage examples.

Dependencies

- Python version >= 3.9.15
- Geopandas version >= 1.0.1
- Xarray version >= 2024.6.0
- Cartopy version >= 0.23.0

Functions descriptions

grid_2_grid(raster_path, fold_function, variable_name, long_name, units="value/grid-cell", source=None, time=None, cell_size=1, netcdf_variable=None, output_directory=None, output_filename=None, zero_is_value=None, value_per_area=False, verbose=False)

Input Argument

- raster_path: str – string describing the location path of the raster file. This function automatically understands the extension of the raster (e.g., netcdf or geotiff) and executes the conversion.
- fold_function: str – describing how a user wants to aggregate the grid values to a coarser resolution. A user can either sum, mean, STD or take max values of the high-resolution grids and store them onto the larger grids. If the raw raster has different latitude and longitude resolution (e.g. MERRA-2), or from a dataset with latitude and longitude resolutions that do not evenly divide the target resolution, the tool automatically adjusts the grid value by calculating the weighted fraction to be distributed to each grid cell at the specified resolution.
- variable_name, str – variable name of the final dataset.
- long_name: str - description of the variable to be added to the attributes metadata.
- units: str, optional – optional argument to be added to the attributes metadata. The default is “value/grid-cell”.
- source: str, optional – optional source related description to be added to the attributes metadata.

- time: str, optional – optional argument describing the time period of the variable. If the raw raster is a netcdf variable and a user wants to re-grid the data for a specified time, then the time should be mentioned under time argument as a string variable.
- cell_size: float, optional – optional argument of the desired cell resolution. The default is 1.
- netcdf_variable: str, optional – netcdf variable name if raw raster data is a netcdf file.
- zero_is_value: bool, optional – optional argument describing zero means existence of data or not. If the value is "yes", then the function will consider zero as an existent value and 0 values will be considered while calculating mean or STD.
- output_directory: str, optional – optional argument describing the path of the final netcdf dataset.
- output_filename: str, optional – optional argument describing the name of the final netcdf file without needing the extension of ".nc".
- value_per_area: str, optional – optional argument if input "yes" then the value will be transformed from "value per grid-cell" to "value m^{-2}".
- verbose: bool, optional – by default the input is False, but if it is True, the global gridded statistics (e.g., sum or mean) of before and after re-gridding operation will be printed.

Output and behavior

- The output is a netcdf file transformed from a comparatively high-resolution raster grid. The function automatically checks and converts the spatial resolution and coverage of the raw data. The final netcdf file is a standardized netcdf file where latitude, longitude and time dimensions are standardized and useful for additional operations.

point_2_grid(points_gdf=None, variable_name=None, long_name=None, units="value/grid-cell", source=None, time=None, cell_size=1, fold_field=None, fold_function="sum", attr_field=None, shapefile_path=None, output_directory=None, output_filename=None, value_per_area=False, zero_is_value=False, verbose=False)

Input Arguments

- points_gdf: GeoDataFrame, optional - GeoDataFrame containing point data to be gridded. If not provided, `shapefile_path` must be specified.
- variable_name: str, optional – variable_name description to be added to netcdf attributes metadata. If attr_field is specified, then the variable_name will be the same as the corresponding entries in the unique field column in the shapefile. If neither variable_name nor attr_field is given, variable_name defaults to the input file name without the extension.
- long_name: str, optional - long_name description to be added to netcdf attributes metadata. It behaves the same as variable_name if attr_field is specified, and if neither are specified, defaults to the input filename without the extension.
- units: str, optional – optional string describing units of the data variables to be added to the netcdf file. All data variables created from the input will be assigned this unit description. Default: value/grid_cell.
- source: str, optional – optional argument, string describing the original source of the input data, this will be added to the metadata stored in the netcdf attributes automatically.

- time: str, optional – By default the output is a spatial, 2 dimensional (lat lon) netcdf file. If time is specified, the output will be a 3-dimensional netcdf file with an additional time dimension, for which there is only one value coordinate, given by the input to this argument.
- cell_size: float, optional – optional argument of the desired cell resolution. The default is 1.0.
- fold_field: str, optional – optional argument, by default the output netcdf values will be the number of points per grid cell. If field_name is given, then the values will be derived from the corresponding values in the shapefile column called field_name, combined by the fold_function (which defaults to sum).
- fold_function: str, optional - optional argument describing how to combine data (if field_name is given, otherwise points will simply be counted). The default is to sum. Options include sum, max, min and std. For example, if each point in the shapefile is associated with a value (and the field_name argument is given pointing to which column in the shapefile stores that value), then the new grid cell value will be the sum of all the point values, if fold_function equals 'sum'.
- attr_field: str, optional – optional argument, if the input shapefile has multiple types of data that the user wants to be stored as separate data variables in the netcdf file, specify the column name that stores strings to be used as the data variable names. If no input is given, only one data variable will be created, and named variable_name, if no variable_name is given, it will be named long_name, if no long_name is given, it will be named filename.
- shapefile_path: str, optional – a shapefile that has point data, where the units of the data stored at each point are all the same.
- output directory: str, optional – path to desired netcdf file location.
- output_filename: str, optional - By default, it will save the netcdf file as the input filename by adding .nc as extension. Otherwise, a user can specify the file name.
- value_per_area: str, optional – optional argument if input "yes" then the value will be transformed from "value per grid-cell" to "value m^{-2}".
- zero_is_value : bool, optional - if True, treats zero values as valid data rather than as no-data. Default is False.
- verbose: bool, optional – by default the input is False, but if it is True, the global gridded statistics (e.g., sum or mean) of before and after re-gridding operation will be printed.

Output and behavior

- The output is a netcdf file transformed from the input point shapefile data into a gridded netcdf file. If points lie exactly on a grid boundary, they will be shifted by 0.0001 degree in the positive latitude and positive longitude directions and added to the corresponding grid cell.

line_2_grid(lines_gdf=None, variable_name=None, long_name=None, units="meter/grid-cell", source=None, time=None, cell_size=1, fold_field=None, fold_function="sum", attr_field=None, shapefile_path=None, output_directory=None, output_filename=None, value_per_area=False, zero_is_value=False, verbose=False)

Input Arguments

- lines_gdf : GeoDataFrame, optional - GeoDataFrame containing line data to be gridded. If not provided, `shapefile_path` must be specified.
- variable_name – variable_name description to be added to netcdf attributes metadata. If attr_field is specified, then the variable_name will be the same as the corresponding entries in

the unique field column in the shapefile. If neither variable_name nor attr_field is given, variable_name defaults to the input file name without the extension.

- long_name - long_name description to be added to netcdf attributes metadata. It behaves the same as variable_name if attr_field is specified, and if neither are specified, defaults to the input file_name without the extension.
- units – optional string describing units of the data variables to be added to the netcdf file. All data variables created from the input will be assigned this unit description. Default: meter/grid_cell.
- source – string describing the original source of the input data, this will be added to the metadata stored in the netcdf attributes automatically.
- time – By default the output is a spatial, 2 dimensional (lat lon) netcdf file. If time is specified, the output will be a 3-dimensional netcdf file with an additional time dimension, for which there is only one value coordinate, given by the input to this argument.
- cell_size – optional argument of the desired cell resolution. The default is 1.
- field_field – optional argument, by default the output netcdf values will be the summation of length of the intersected lines per grid cell. If field_name is given, then the values will be derived from the corresponding values in the shapefile column called field_name, combined by the fold_function (which defaults to sum).
- fold_function - optional argument describing how to combine data (if field_name is given, otherwise points will simply be summation of intersected lines per grid cell). The default is to sum. Options include sum, max, min and stddev. For example, if each line in the shapefile is associated with a value (e.g., width of a road) and the field_name argument is given pointing to which column in the shapefile stores that value, then the new grid cell value will be the sum all the values, if fold_function equals 'sum'.
- attr_field – optional argument, if the input shapefile has multiple types of data that the user wants to be stored as separate data variables in the netcdf file, specify the column name that stores strings to be used as the data variable names. If no input is given, only one data variable will be created, and named variable_name, if no variable_name is given, it will be named long_name, if no long_name is given, it will be named file_name.
- shapefile_path: str, optional – a shapefile that has line data, where the units of the data stored at each line are all the same (e.g., width of road in meter).
- output directory – path to desired netcdf file location.
- output_filename - By default, it will save the netcdf file as the input filename by adding .nc as extension. Otherwise, a user can specify the file name.
- value_per_area: str, optional – optional argument if input "yes" then the value will be transformed from "value per grid-cell" to "value m^{-2}".
- zero_is_value : bool, optional - if True, treats zero values as valid data rather than as no-data. Default is False.
- verbose: bool, optional – by default the input is False, but if it is True, the global gridded statistics (e.g., sum or mean) of before and after re-gridding operation will be printed.

Output and behavior

- The output is a netcdf file transformed from the input line shapefile data into a gridded netcdf file.

poly_2_grid(poly_gdf=None, variable_name=None, long_name=None, units="m2/grid-cell", source=None, time=None, cell_size=1, attr_field=None, shapefile_path=None, fraction=False, fold_function="sum", output_directory=None, output_filename=None, value_per_area=False, zero_is_value=False, verbose=False)

Input Arguments

- poly_gdf : GeoDataFrame, optional - GeoDataFrame containing polygon data to be gridded. If not provided, `shapefile_path` must be specified.
- variable_name, str, optional – variable_name description to be added to netcdf attributes metadata. If attr_field is specified, then the variable_name will be the same as the corresponding entries in the unique field column in the shapefile. If neither variable_name nor attr_field is given, variable_name defaults to the input file name without the extension.
- long_name, str, optional - long_name description to be added to netcdf attributes metadata. It behaves the same as variable_name if attr_field is specified, and if neither are specified, defaults to the input file_name without the extension.
- units: str, optional – optional string describing units of the data variables to be added to the netcdf file. All data variables created from the input will be assigned this unit description. Default: m2/grid_cell.
- source: str, optional – string describing the original source of the input data, this will be added to the metadata stored in the netcdf attributes automatically.
- time: str, optional – By default the output is a spatial, 2 dimensional (lat lon) netcdf file. If time is specified, the output will be a 3-dimensional netcdf file with an additional time dimension, for which there is only one value coordinate, given by the input to this argument.
- cell_size: float, optional – optional argument of the desired cell resolution. The default is 1.
- attr_field: str, optional – optional argument, if the input shapefile has multiple types of data that the user wants to be stored as separate data variables in the netcdf file, specify the column name that stores strings to be used as the data variable names. If no input is given, only one data variable will be created, and named variable_name, if no variable_name is given, it will be named long_name, if no long_name is given, it will be named file_name.
- shapefile_path: str, optional – a shapefile that has polygon data.
- fraction: bool, optional – by default False, if yes then the fraction each polygon in each grid will be generated. The data range would be 0 to 1.
- fold_function : str, optional - aggregation method for combining values in each grid cell. Default is 'sum'. Options include: ‘sum’, ‘max’, ‘min’, ‘mean’.
- output directory: str, optional – path to desired netcdf file location.
- output_filename: str, optional - By default, it will save the netcdf file as the input filename by adding .nc as extension. Otherwise, a user can specify the file name.
- value_per_area: bool, optional – optional argument if input “yes” then the value will be transformed from “value per grid-cell” to “value m^{-2}”.
- zero_is_value : bool, optional - if True, treats zero values as valid data rather than as no-data. Default is False.
- verbose: bool, optional – by default the input is False, but if it is True, the global gridded statistics (e.g., sum or mean) of before and after re-gridding operation will be printed.

Output and behavior

- The output is a netcdf file transformed from the input polygon shapefile data into a gridded netcdf file.

table_2_grid(netcdf_variable, tabular_column, netcdf_file_path=None, csv_file_path=None, input_ds=None, input_df=None, variable_name=None, long_name=None, units="value/grid-cell", source=None, time=None, output_directory=None, output_filename=None, zero_is_value=None, value_per_area=None, verbose=False)

Input Arguments

- netcdf_variable: str – a dataset variable which acts as the surrogate variable of the dasymmetric operation. The spatial resolution of the variable must be 1 or 0.5 degrees.
- tabular_column: str **–** a string variable representing the column name of the tabular data.
- netcdf_file_path, str, optional – optional argument, a netcdf variable data path as string.
- csv_file_path: str, optional – a tabular csv file where data is stored based on their jurisdiction or ISO3 code. The csv file must hold a column named "ISO3". If not, then users must use country_2_ISO3 function to convert the country name to their corresponding ISO3 code.
- input_ds: str, optional – optional argument of a xarray dataset if available. The function needs a mandatory surrogate variable either in netcdf or xarray dataset format.
- input_df: str, optional– optional argument of a pandas dataframe if available. The function needs mandatory tabular data either in csv or pandas dataframe format.
- variable_name: str, optional – optional argument of the final netcdf variable name. If variable_name is not mentioned, then the tabular_column name will be used for the ultimate variable name.
- long_name: str, optional – optional argument to be added to the netcdf variable's attributes.
- units: str, optional – optional argument to be added to the variable's attributes. The default is "value/grid-cell".
- source: str, optional - optional argument to be added to the netcdf variable's attributes.
- time: str, optional – optional argument representing the variable's timeframe. A string time needs to be indicated, and based on the input time, the country fraction and surrogate variable (if available) time will be selected.
- output_directory: str, optional – optional argument of output path location where the final netcdf file will be stored.
- output_filename: str, optional – optional argument describing the final netcdf file name without needing the extension of ".nc".
- value_per_area: bool, optional – optional argument, if input "yes" then the value will be transformed into "value m-2".
- zero_is_value: bool, optional - if True, treats zero values as valid data rather than as no-data. Default is False.
- verbose: bool, optional – by default the input is False, but if it is "yes", the global gridded sum of before and after re-gridding operation will be printed. If any jurisdiction where surrogate variable is missing and tabular data is evenly distributed over the jurisdiction, the ISO3 codes of evenly distributed countries will also be printed.

Output and behavior

- The output is a netcdf file transformed from tabular data to gridded format. The function converts the tabular data proportional to a surrogate variable onto global spatial grids.

grid_2_table(netcdf_path=None, dataset=None, variable=None, time=None, grid_area=None, cell_size=1, aggregation=None, method='sum', verbose=False)

Input Arguments

- netcdf_path : str, optional - Netcdf path containing path location.
- dataset : xarray Dataset, optional - Gridded dataset containing spatial information.
- variable : str, optional - variable name to be processed. If None, all variables in the dataset (excluding predefined ones) will be considered.
- time : str, optional - time slice for data processing. If provided, the nearest time slice is selected. If None, a default time slice is used.
- grid_area : str, optional - indicator to consider grid area during processing. If 'YES', the variable is multiplied by grid area.
- aggregation : str, optional - aggregation level for tabular data. If 'continent', the data will be aggregated at the continent level.
- method : str, optional - aggregation method. Options: 'sum', 'mean', 'max'.

Output and behavior

Returns a pandas DataFrame Tabular data for different jurisdictions, including ISO3 codes, variable values, and optional 'Year' column.

plot_histogram(variable, dataset=None, bin_size=30, color='blue', plot_title=None, x_label=None, remove_outliers=False, log_transform=None, output_dir=None, filename=None, netcdf_directory=None)

Input Arguments

- dataset: xarray.Dataset, the dataset containing the variables.
- variable: str, the name of the variable to plot.
- bin_size: int, optional, the number of bins in the histogram.
- color: str, optional, the color of the histogram bars.
- remove_outliers: bool, optional, whether to remove outliers.
- log_transform: str, optional, the type of log transformation ('log10', 'log', 'log2').
- netcdf_directory: str, optional: directory where netcdf file is located.

Output and behavior

Creates a histogram for an array variable in an xarray dataset. Optionally remove outliers and apply log transformations.

plot_scatter(variable1, variable2, dataset=None, dataset2=None, color='blue', x_label=None, y_label=None, plot_title=None, remove_outliers=False, log_transform_1=None, log_transform_2=None, equation=False, output_dir=None, filename=None, netcdf_directory=None, netcdf_directory2=None)

Input Arguments

- dataset: xarray.Dataset, the dataset containing the variables for the x-axis.
- variable1: str, the name of the first variable to plot on the x-axis.
- variable2: str, the name of the second variable to plot on the y-axis. If dataset2 is provided, this variable is from dataset2.
- dataset2: xarray.Dataset, optional, a second dataset containing the variable for the y-axis.
- remove_outliers: bool, optional, whether to remove outliers.
- log_transform: str, optional, the type of log transformation ('log10', 'log', 'log2').
- color: str, optional, the color of the scatter plot points.
- netcdf_directory: str, optional: directory where netcdf file of the variable is located.
- netcdf_directory2: str, optional: directory where netcdf file of the 2nd variable is located.

Output and behavior

Creates a scatter plot for two variables in an xarray dataset. Optionally remove outliers and apply log transformations.

plot_time_series(variable, dataset=None, fold_function='sum', plot_type='both', color='blue', plot_label='Area Plot', x_label='Year', y_label='Value', plot_title='Time Series Plot', smoothing_window=None, output_dir=None, filename=None, netcdf_directory=None)

Input Arguments

- dataset: xarray.Dataset, the dataset containing the variable to plot.
- variable: str, the name of the variable to plot.
- fold_function: str, the operation to apply ('sum', 'mean', 'max', 'std').
- smoothing_window: int, optional, the window size for rolling mean smoothing.
- plot_type: str, optional, the type of plot ('line', 'area', 'both'). Default is 'both'.
- color: str, optional, the color of the plot. Default is 'blue'.
- plot_label: str, optional, the label for the plot. Default is 'Area Plot'.
- x_label: str, optional, the label for the x-axis. Default is 'Year'.
- y_label: str, optional, the label for the y-axis. Default is 'Value'.
- plot_title: str, optional, the title of the plot. Default is 'Time Series Plot'.
- output_dir: str, optional, the directory to save the plot.
- filename: str, optional, the filename to save the plot.
- netcdf_directory: str, optional: directory where netcdf file is located.

Output and behavior

Creates a line plot and/or area plot for a time series data variable.

plot_hexbin(variable1, variable2, dataset=None, dataset2=None, color='pink_r', grid_size=30, x_label=None, y_label=None, plot_title=None, remove_outliers=False, log_transform_1=None, log_transform_2=None, output_dir=None, filename=None, netcdf_directory=None, netcdf_directory2=None)

Input Arguments

- dataset: xarray.Dataset, the dataset containing the variables for the x-axis.
- variable1: str, the name of the first variable to plot on the x-axis.
- variable2: str, the name of the second variable to plot on the y-axis. If dataset2 is provided, this variable is from dataset2.
- dataset2: xarray.Dataset, optional, a second dataset containing the variable for the y-axis.
- color: str, optional, the color map of the hexbin plot.
- grid_size: int, optional, the number of hexagons in the x-direction.
- x_label: str, optional, the label for the x-axis.
- y_label: str, optional, the label for the y-axis.
- plot_title: str, optional, the title for the plot.
- remove_outliers: bool, optional, whether to remove outliers from the data.
- log_transform_1: str, optional, the type of log transformation for variable1 ('log10', 'log', 'log2').
- log_transform_2: str, optional, the type of log transformation for variable2 ('log10', 'log', 'log2').
- netcdf_directory: str, optional: directory where netcdf file of the variable is located.
- netcdf_directory2: str, optional: directory where netcdf file of the 2nd variable is located.

Output and behavior

Creates a hexbin plot for two variables in an xarray dataset.

plot_map(variable, dataset=None, color='hot_r', title='', label='', color_min=None, color_max=None, levels=10, output_dir=None, filename=None, netcdf_directory=None)

Input Arguments

- variable : str. The name of the variable to plot from the dataset.
- dataset : xarray.Dataset or None, optional. The dataset containing the variable to be plotted. If None, the dataset is read from `netcdf_directory`.
- color : str, optional. The colormap to use for the plot. Default is 'hot_r'.
- title : str, optional. The title of the plot. Default is an empty string.
- label : str, optional. The label for the colorbar. Default is an empty string.
- color_min : float or None, optional. The minimum value for the color scale. If None, the minimum value is determined automatically. Default is None.
- color_max : float or None, optional. The maximum value for the color scale. If None, the maximum value is determined automatically. Default is None.
- levels : int, optional. Number of levels for the color scale. Default is 10.
- output_dir : str or None, optional. Directory to save the output plot. If None, the plot is not saved. Default is None.
- filename : str or None, optional. The name of the output file for the plot. If None, no file is saved. Default is None.

- netcdf_directory : str or None, optional. Directory containing the netCDF file to load the dataset from, if `dataset` is not provided. Default is None.

Output and behavior
Plots a variable from a dataset as a 2D map with customizable options.

sum_variables(variables=None, dataset=None, new_variable_name=None, time=None, netcdf_directory=None)

Input Arguments

- variables: list of str, the names of the variables to sum. If None, sum all variables except those starting with 'grid_area'.
- dataset: xarray.Dataset, optional, the dataset containing the variables.
- new_variable_name: str, optional, the name of the new variable to store the sum.
- time: optional, a specific time slice to select from the dataset.
- netcdf_directory: str, optional: directory where netcdf file is located.

Output and behavior
Returns an xarray.Dataset, with the summed variable. Sum specified variables in the xarray dataset. If no variables are specified, sum all variables except those starting with 'grid_area'. Fill NaNs with zero before summing, and convert resulting zeros back to NaNs.

subtract_variables(variable1, variable2, dataset=None, new_variable_name=None, time=None, netcdf_directory=None)

Input Arguments

- variable1: str, the name of the variable to subtract from.
- variable2: str, the name of the variable to subtract.
- dataset: xarray.Dataset, optional, the dataset containing the variables.
- new_variable_name: str, optional, the name of the new variable to store the result.
- time: optional, a specific time slice to select from the dataset.
- netcdf_directory: str, optional: directory where netcdf file is located.

Output and behavior
Returns an xarray.Dataset, with the resulting variable. Subtract one variable from another in the xarray dataset. Fill NaNs with zero before subtracting, and convert resulting zeros back to NaNs.

divide_variables(variable1, variable2, dataset=None, new_variable_name=None, time=None, netcdf_directory=None)

Input Arguments

- variable1: str, the name of the variable to be divided (numerator).
- variable2: str, the name of the variable to divide by (denominator).
- dataset: xarray.Dataset, optional, the dataset containing the variables.
- new_variable_name: str, optional, the name of the new variable to store the result.
- time: optional, a specific time slice to select from the dataset.
- netcdf_directory: str, optional: directory where netcdf file is located.

Output and behavior

Returns an xarray.Dataset, with the resulting variable. Divide one variable by another in the xarray dataset. Fill NaNs with zero before dividing, and convert resulting zeros back to NaNs.

multiply_variables(variables=None, dataset=None, new_variable_name=None, time=None, netcdf_directory=None)

Input Arguments

- variables: list of str, the names of the variables to multiply. If None, multiply all variables.
- dataset: xarray.Dataset, optional, the dataset containing the variables.
- new_variable_name: str, optional, the name of the new variable to store the product.
- time: optional, a specific time slice to select from the dataset.
- netcdf_directory: str, optional: directory where netcdf file is located.

Output and behavior

Returns an xarray.Dataset, with the resulting variable. Multiply specified variables in the xarray dataset. If no variables are specified, multiply all variables. Fill NaNs with one before multiplying, and convert resulting ones back to NaNs.

average_variables(variables=None, dataset=None, new_variable_name=None, time=None, netcdf_directory=None)

Input Arguments

- variables: list of str, the names of the variables to average. If None, average all variables except those starting with 'grid_area'.
- dataset: xarray.Dataset, optional, the dataset containing the variables.
- new_variable_name: str, optional, the name of the new variable to store the average.
- time: optional, a specific time slice to select from the dataset.
- netcdf_directory: str, optional: directory where netcdf file is located.

Output and behavior

Returns an xarray.Dataset, with the resulting variable. Average specified variables in the xarray dataset. If no variables are specified, average all variables except those starting with 'grid_area'. Fill NaNs with zero before averaging, and convert resulting zeros back to NaNs.

get_netcdf_info(netcdf_path, variable_name=None)

Input Arguments

- netcdf_path : str. The file path to the NetCDF dataset.
- variable_name : str, optional. The prefix or complete name of the variable to filter. If not provided, all variables are included.

Output and behavior

Returns a tuple containing lists of dimensions, short names, long names, units, & time values (if 'time' exists). Extract information about variables and dimensions from a NetCDF dataset.

country_2_iso3(df, column)

Input Arguments

- df (pandas.DataFrame): The DataFrame containing a column with country names.
- column (str): The name of the column in the DataFrame that contains country names.

Output and behavior

Returns a pandas.DataFrame. The original DataFrame with an additional 'ISO3' column containing the ISO3 country codes. FileNotFoundError: If the JSON file containing country mappings cannot be found. KeyError: If the specified column is not present in the DataFrame. The resulting ISO3 codes are added as a new column named 'ISO3'.