# The SESAME Software Manual

Document Version 1.1

Release date – 12 May 2025

**Research Group**
Integrated Earth System Dynamics (iESD)
Department of Earth and Planetary Sciences
McGill University
Canada
Website: https://earthsystemdynamics.org/

**Contact**
Abdullah Al Faisal – abdullah-al.faisal@mail.mcgill.ca
Maxwell Kaye – maxwell.kaye@mail.mcgill.ca
Eric Galbraith – eric.galbraith@mcgill.ca

**Specific Publications**

Faisal, A.A., Kaye, M., Ahmed, M. *et al.* The SESAME Human-Earth Atlas. *Sci Data* **12**, 775 (2025). https://doi.org/10.1038/s41597-025-05087-5

**Human-Earth Atlas**

Faisal, A. A., Kaye, M., Ahmed, M. & Galbraith, E. The SESAME Human-Earth Atlas. *figshare* https://doi.org/10.6084/m9.figshare.28432499 (2025).

**Software Package:**

GitHub: https://github.com/A2Faisal/SESAME

**Revision History**

| Version | Date | Key Changes | Notes |
|---|---|---|---|
| 1.0 | 22 April 2025 | Initial document | SI examples and atlas generation scripts(genscripts) are based on this version |
| 1.1 | 12 May 2025 | Updated functions to support a unified input style — accepting both in-memory objects and file paths (e.g., NetCDF/xarray, CSV/DataFrame, shapefile/GeoDataFrame). | This version is not backward compatible with v1.0. Please review the updated documentation for input requirements. |

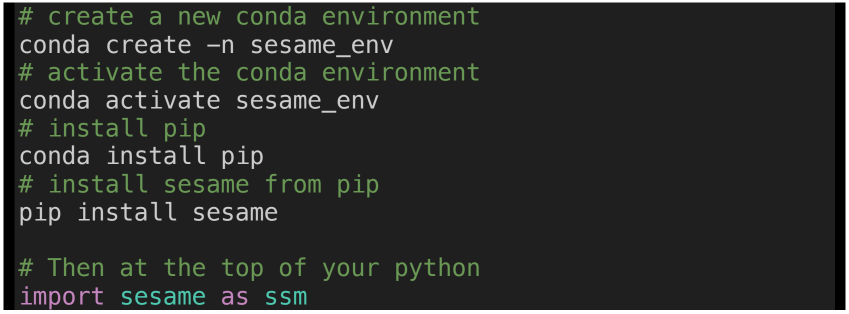
## Table of Contents

## SESAME Software Package

All NetCDF files contain grid_area and land_frac variables if the resolution is either 0.25, 0.5, or 1-degree. If the desired resolution does not fall under the mentioned resolution, then the SESAME software package automatically creates a grid_area variable (where the variable represents the area of each grid cell in m²) at the desired resolution. The features of the SESAME software package are subject to enhancement over time. The capabilities described here reflect the state of the software at the time of publication. Users are encouraged to consult the latest software documentation on GitHub for updates (https://github.com/A2Faisal/SESAME).

## Installation Instructions

If you want to use conda to manage your environment:

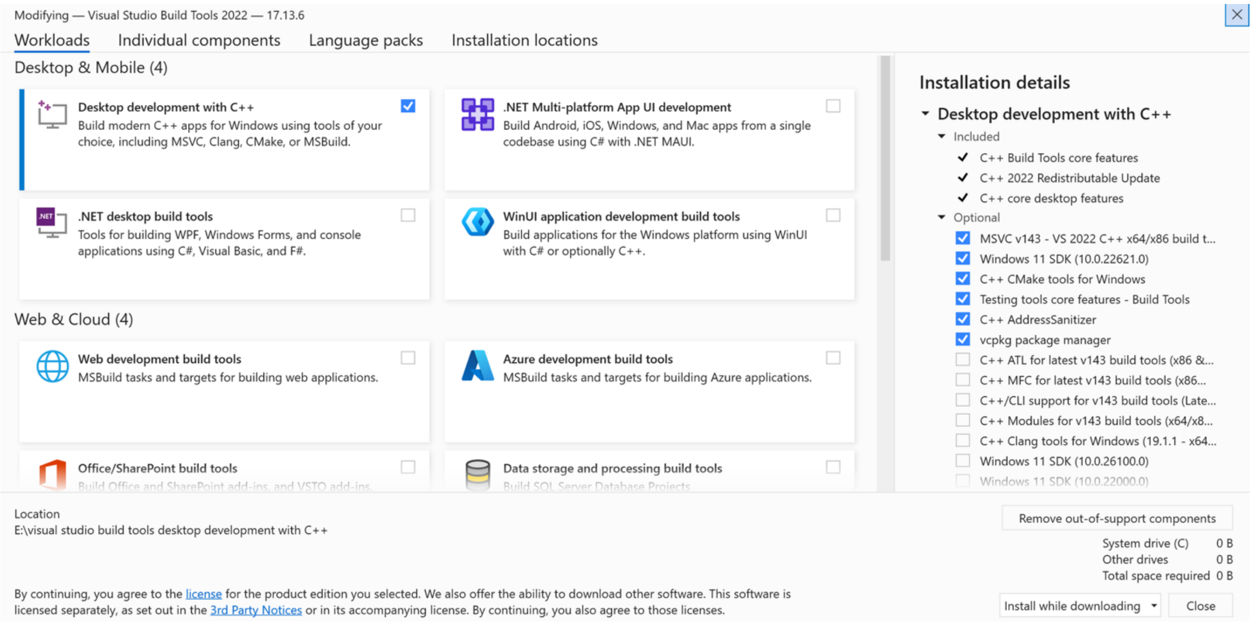


```
# create a new conda environment
conda create -n sesame_env
# activate the conda environment
conda activate sesame_env
# install pip
conda install pip
# install sesame from pip
pip install sesame

# Then at the top of your python
import sesame as ssm
```

If you are using Windows and encounter the error: "ERROR: Failed building wheel for cartopy," please follow these steps:

1. Download and install the Microsoft C++ Build Tools from the official website: (https://visualstudio.microsoft.com/visual-cpp-build-tools/)
2. During installation, make sure to select the "Desktop development with C++" workload. This is required to build packages like `cartopy` that depend on C++ extensions.



Modifying — Visual Studio Build Tools 2022 — 17.13.6

Desktop & Mobile (4)

☑ **Desktop development with C++**
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

☐ **.NET Multi-platform App UI development**
Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.

☐ **.NET desktop build tools**
Tools for building WPF, Windows Forms, and console applications using C#, Visual Basic, and F#.

☐ **WinUI application development build tools**
Build applications for the Windows platform using WinUI with C# or optionally C++.

Web & Cloud (4)

☐ **Web development build tools**
MSBuild tasks and targets for building web applications.

☐ **Azure development build tools**
MSBuild tasks and targets for building Azure applications.

☐ **Office/SharePoint build tools**
Build Office and SharePoint add-ins, and VSTO add-ins

☐ **Data storage and processing build tools**
Build SQL Server Database Projects

Installation details

Desktop development with C++

Included
- ✓ C++ Build Tools core features
- ✓ C++ 2022 Redistributable Update
- ✓ C++ core desktop features

Optional
- ☑ MSVC v143 - VS 2022 C++ x64/x86 build t...
- ☑ Windows 11 SDK (10.0.22621.0)
- ☑ C++ CMake tools for Windows
- ☑ Testing tools core features - Build Tools
- ☑ C++ AddressSanitizer
- ☑ vcpkg package manager
- ☐ C++ ATL for latest v143 build tools (x86 &...
- ☐ C++ MFC for latest v143 build tools (x86...
- ☐ C++/CLI support for v143 build tools (Late...
- ☐ C++ Modules for v143 build tools (x64/x8...
- ☐ C++ Clang tools for Windows (19.1.1 - x64...
- ☐ Windows 11 SDK (10.0.26100.0)
- ☐ Windows 11 SDK (10.0.22000.0)

Location
E:\visual studio build tools desktop development with C++

System drive (C) 0 B
Other drives 0 B
Total space required 0 B

By continuing, you agree to the license for the product edition you selected. We also offer the ability to download other software. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

## 1. Major functions

### 1.1 grid_2_grid

*grid_2_grid(raster_path, agg_function, variable_name, long_name, units="value/grid-cell", source=None, time=None, resolution=1, netcdf_variable=None, output_directory=None, output_filename=None, zero_is_value=False, normalize_by_area=False, verbose=False)*

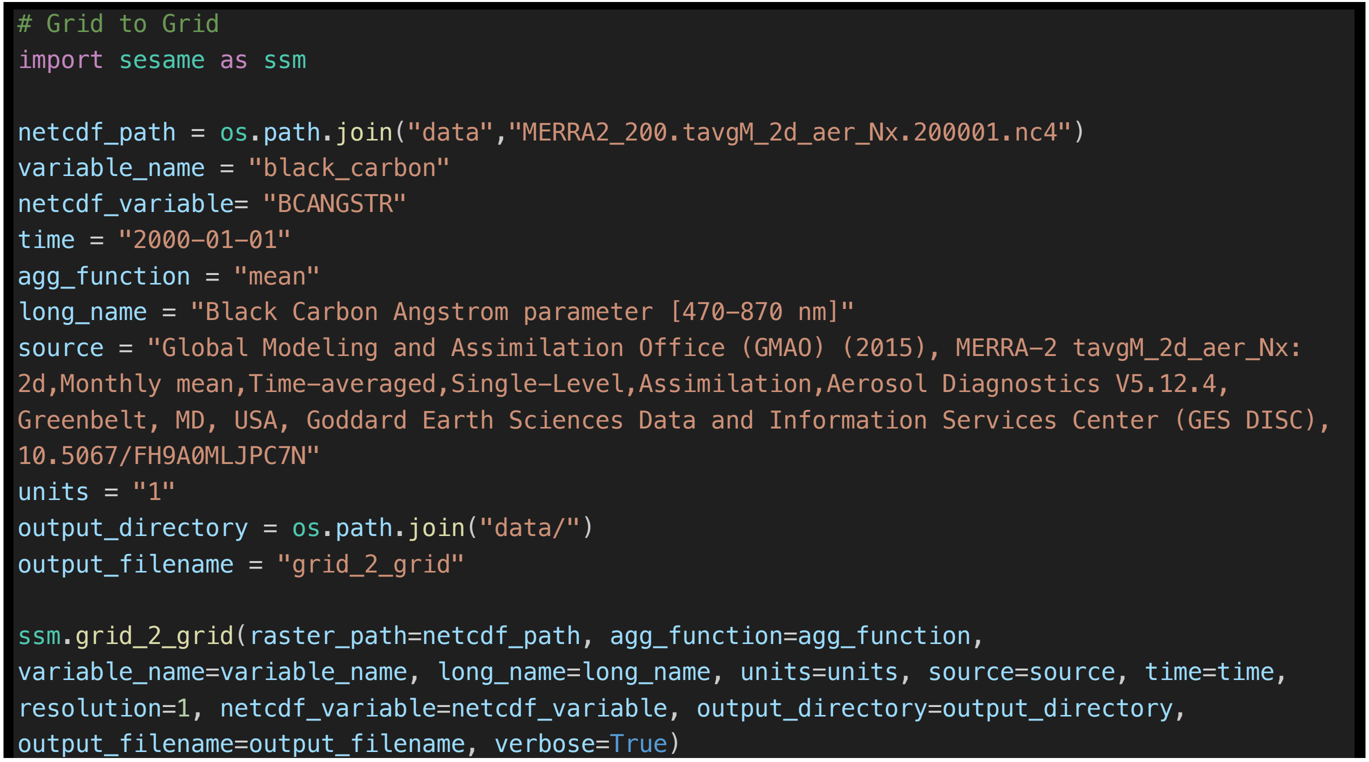
**Input Arguments**

- 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.
- agg_function: str – describing how a user wants to aggregate the grid values to a coarser resolution. A user can request sum, mean, STD or 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), 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 – argument to be added to the attributes metadata. The default is “value/grid-cell”.
- source: str, optional – source related description to be added to the attributes metadata.
- time: str, 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.
- resolution: float, optional – desired cell resolution in degree. The default is 1.
- netcdf_variable: str, optional – netcdf variable name if raw raster data is a netcdf file.
- output_directory: str, optional – Directory where the output NetCDF file will be saved. If None, but output_filename is True, the file will be saved in the current working directory.
- output_filename : str, optional – Name of the output NetCDF file (without the `.nc` extension). If not provided:
  - Uses `variable_name` if it is specified.
  - Defaults to `regridded.nc` if none of the above are provided.
- zero_is_value: bool, optional – if “yes”, then the function will consider zero as an existent value and 0 values will be considered while calculating mean or STD.
- normalize_by_area: bool, optional – if True, the value will be transformed to “value m$^{-2}$”, assuming the input was “value per grid-cell”.
- verbose: bool, optional – default=False, but if true, the global gridded statistics (e.g., sum) 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 uses standardized latitude, longitude and time dimensions.

**Example: grid_2_grid**

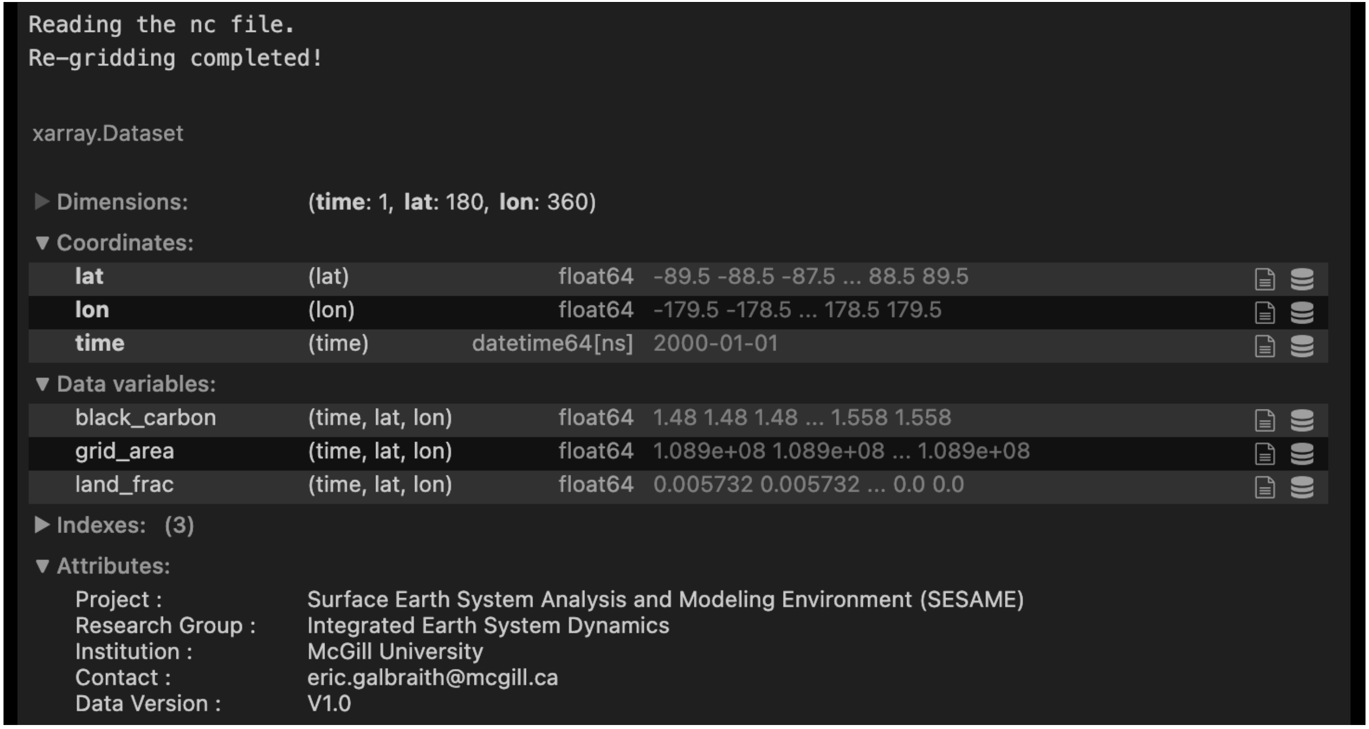


```
# Grid to Grid
import sesame as ssm

netcdf_path = os.path.join("data","MERRA2_200.tavgM_2d_aer_Nx.200001.nc4")
variable_name = "black_carbon"
netcdf_variable= "BCANGSTR"
time = "2000-01-01"
agg_function = "mean"
long_name = "Black Carbon Angstrom parameter [470-870 nm]"
source = "Global Modeling and Assimilation Office (GMAO) (2015), MERRA-2 tavgM_2d_aer_Nx:
2d,Monthly mean,Time-averaged,Single-Level,Assimilation,Aerosol Diagnostics V5.12.4,
Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC),
10.5067/FH9A0MLJPC7N"
units = "1"
output_directory = os.path.join("data/")
output_filename = "grid_2_grid"

ssm.grid_2_grid(raster_path=netcdf_path, agg_function=agg_function,
variable_name=variable_name, long_name=long_name, units=units, source=source, time=time,
resolution=1, netcdf_variable=netcdf_variable, output_directory=output_directory,
output_filename=output_filename, verbose=True)
```

**Output:**



```
Reading the nc file.
Re-gridding completed!


xarray.Dataset

▶ Dimensions:        (time: 1, lat: 180, lon: 360)
▼ Coordinates:
    lat              (lat)                    float64  -89.5 -88.5 -87.5 ... 88.5 89.5
    lon              (lon)                    float64  -179.5 -178.5 ... 178.5 179.5
    time             (time)            datetime64[ns]  2000-01-01
▼ Data variables:
    black_carbon     (time, lat, lon)         float64  1.48 1.48 1.48 ... 1.558 1.558
    grid_area        (time, lat, lon)         float64  1.089e+08 1.089e+08 ... 1.089e+08
    land_frac        (time, lat, lon)         float64  0.005732 0.005732 ... 0.0 0.0
▶ Indexes:  (3)
▼ Attributes:
    Project :          Surface Earth System Analysis and Modeling Environment (SESAME)
    Research Group :   Integrated Earth System Dynamics
    Institution :      McGill University
    Contact :          eric.galbraith@mcgill.ca
    Data Version :     V1.0
```

### 1.2 point_2_grid

*point_2_grid(points, variable_name=None, long_name=None, units="value/grid-cell", source=None, time=None, resolution=1, agg_column=None, agg_function="sum", attr_field=None, output_directory=None, output_filename=None, normalize_by_area=False, zero_is_value=False, verbose=False)*

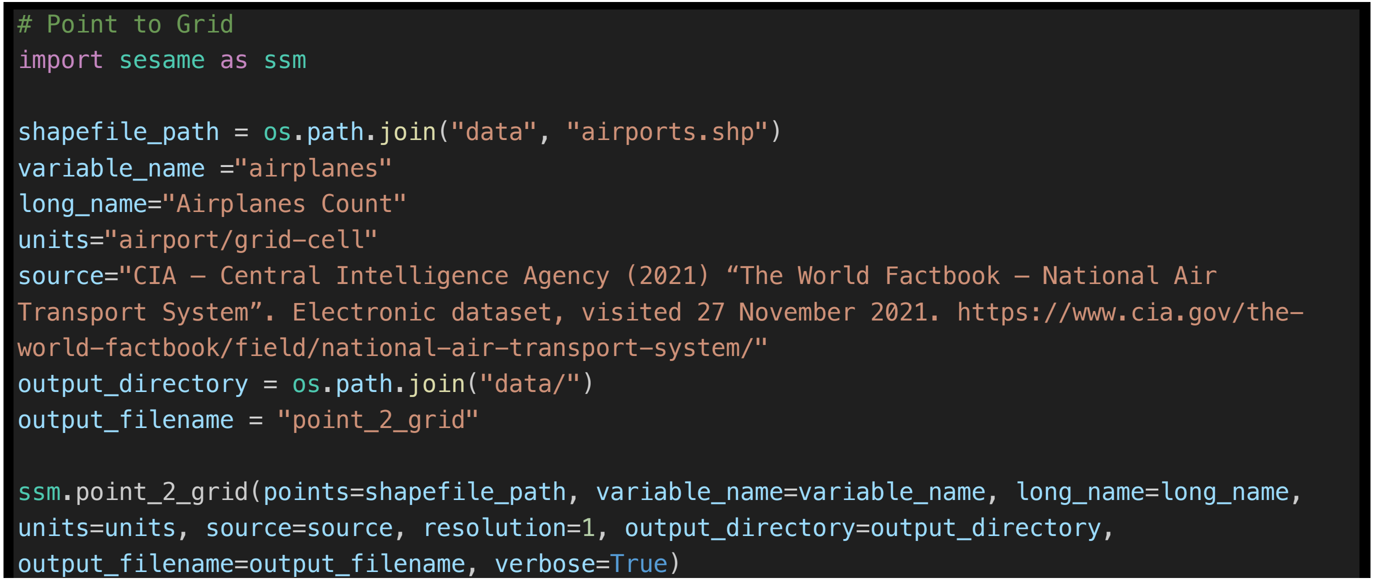
**Input Arguments**

- points: GeoDataFrame or str, input point data to be gridded. Can be either a GeoDataFrame or a path to a point shapefile (.shp).
- variable_name: str, optional – variable_name description to be added to netcdf attributes metadata. If not specified, then "variable" will be used as the variable name. If attr_field is specified, the variable_name will be the same as the corresponding entries in the unique field column in the shapefile/geodataframe.
- long_name: str, optional - long_name description to be added to netcdf attributes metadata. It behaves the same as variable_name.
- units: str, 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 – 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.
- resolution: float, optional – argument of the desired cell resolution in degree. The default is 1.
- agg_column: str, optional – argument, by default the output netcdf values will be the number of points per grid cell. If agg_column is given, the summary values will be derived from the corresponding shapefile/geodataframe column, combined by the agg_function (which defaults to sum).
- agg_function: str, optional - argument describing how to combine data (if agg_column 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 agg_column 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 agg_function='sum'.
- attr_field: str, optional – argument, if the input shapefile has multiple types of data that the user wants to be stored as separate data variables in the final gridded dataset. If the column name is given, each unique value within the column will be considered as a separate variable.
- output_directory: str, optional – Directory where the output NetCDF file will be saved. If None, but output_filename is True, the file will be saved in the current working directory.
- output_filename : str, optional – Name of the output NetCDF file (without the `.nc` extension). If not provided:
  - Uses the input shapefile name if a shapefile path is given.
  - Saves as `"gridded_points.nc"` if a GeoDataFrame is provided as input.
- normalize_by_area : bool, optional – if True, the value will be transformed to "value $m^{-2}$", assuming the input was "value per grid-cell".
- zero_is_value: bool, optional - if True, treats zero values as valid data rather than as no-data. Default is False.
- verbose: bool, optional – default=False, but if it is True, the global gridded statistics (e.g., sum) 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 in order to avoid double counting.

**Example: point_2_grid**

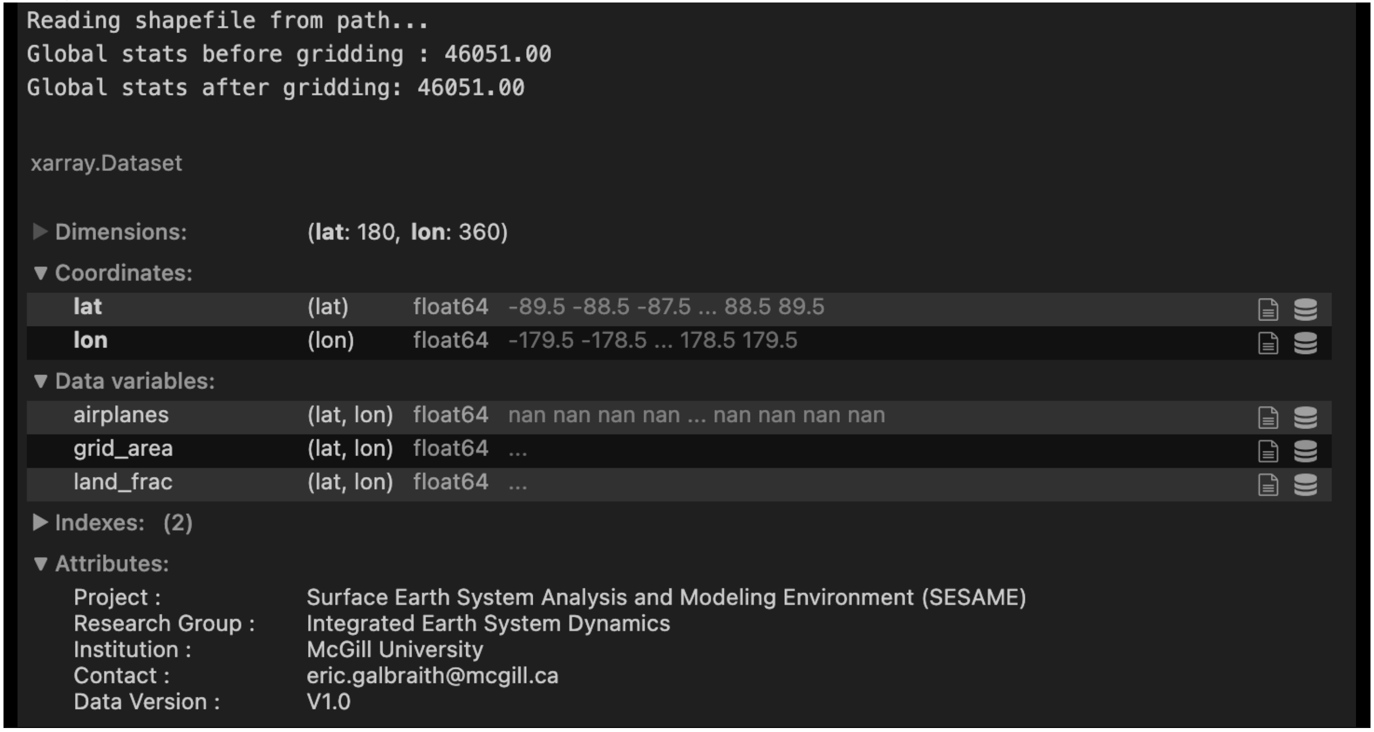


```
# Point to Grid
import sesame as ssm

shapefile_path = os.path.join("data", "airports.shp")
variable_name ="airplanes"
long_name="Airplanes Count"
units="airport/grid-cell"
source="CIA – Central Intelligence Agency (2021) "The World Factbook – National Air
Transport System". Electronic dataset, visited 27 November 2021. https://www.cia.gov/the-
world-factbook/field/national-air-transport-system/"
output_directory = os.path.join("data/")
output_filename = "point_2_grid"

ssm.point_2_grid(points=shapefile_path, variable_name=variable_name, long_name=long_name,
units=units, source=source, resolution=1, output_directory=output_directory,
output_filename=output_filename, verbose=True)
```

**Output:**



```
Reading shapefile from path...
Global stats before gridding : 46051.00
Global stats after gridding: 46051.00


xarray.Dataset

▶ Dimensions:          (lat: 180, lon: 360)
▼ Coordinates:
    lat                (lat)       float64   -89.5 -88.5 -87.5 ... 88.5 89.5
    lon                (lon)       float64   -179.5 -178.5 ... 178.5 179.5
▼ Data variables:
    airplanes          (lat, lon)  float64   nan nan nan nan ... nan nan nan nan
    grid_area          (lat, lon)  float64   ...
    land_frac          (lat, lon)  float64   ...
▶ Indexes:  (2)
▼ Attributes:
    Project :          Surface Earth System Analysis and Modeling Environment (SESAME)
    Research Group :   Integrated Earth System Dynamics
    Institution :      McGill University
    Contact :          eric.galbraith@mcgill.ca
    Data Version :     V1.0
```

### 1.3 line_2_grid

*line_2_grid(lines, variable_name=None, long_name=None, units="meter/grid-cell", source=None, time=None, resolution=1, agg_column=None, agg_function="sum", attr_field=None, output_directory=None, output_filename=None, normalize_by_area=False, zero_is_value=False, verbose=False)*

**Input Arguments**

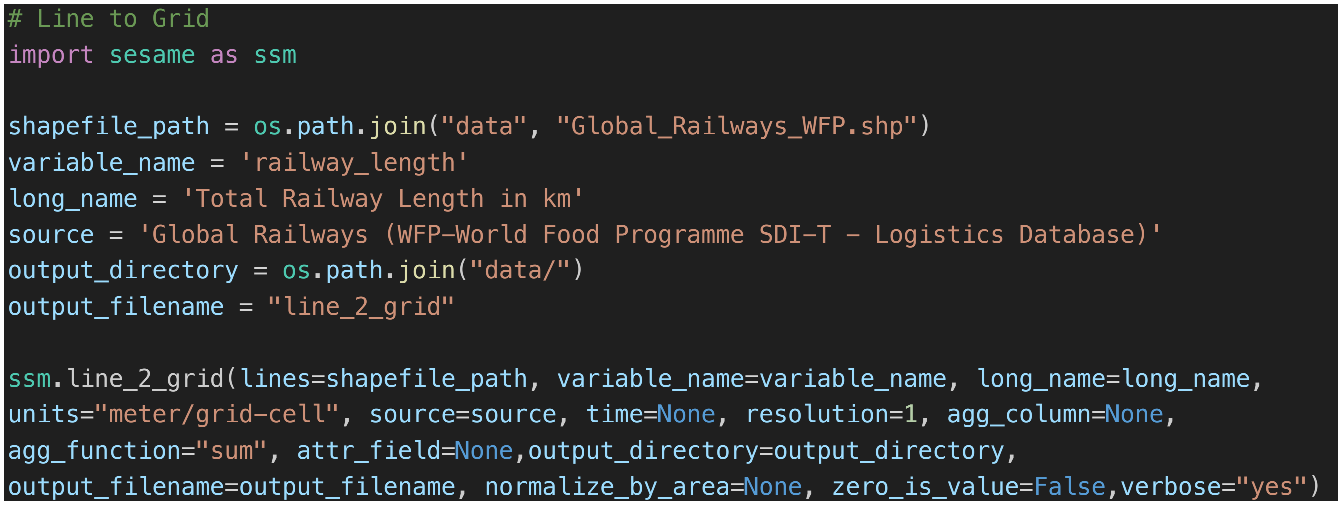
- lines: GeoDataFrame or str – input lines data to be gridded. Can be either a GeoDataFrame or a path to a line/polyline shapefile (.shp).
- variable_name: str, optional – variable_name description to be added to netcdf attributes metadata. If not specified, then "variable" will be used as the variable name. If attr_field is specified, the variable_name will be the same as the corresponding entries in the unique field column in the shapefile/geodataframe.
- long_name: str, optional - long_name description to be added to netcdf attributes metadata. It behaves the same as variable_name.
- units: str, 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: 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.
- resolution: float, optional – argument of the desired cell resolution in degree. The default is 1.
- agg_column: str, optional – by default the output netcdf values will be the summation of length of the intersected lines per grid cell. If agg_column is given, the summary values will be derived from the corresponding shapefile/geodataframe column, combined by the agg_function (which defaults to sum).
- agg_function: str, optional - argument describing how to combine data (if agg_column is given, otherwise length of lines in each grid-cell will simply be calculated). The default is to sum. Options include sum, max, min and std. For example, if each line in the shapefile is associated with a value (and the agg_column argument is given pointing to which column in the shapefile stores that value), then the new grid cell value will be the mean of all the line values (e.g., mean road width in a grid-cell), if agg_function='mean'.
- attr_field: str, optional – argument, if the input shapefile has multiple types of data that the user wants to be stored as separate data variables in the final gridded dataset. If the column name is given, each unique value within the column will be considered as a separate variable.
- output_directory: str, optional – Directory where the output NetCDF file will be saved. If None, but output_filename is True, the file will be saved in the current working directory.
- output_filename : str, optional – Name of the output NetCDF file (without the `.nc` extension). If not provided:
  - Uses the input shapefile name if a shapefile path is given.
  - Saves as `"gridded_lines.nc"` if a GeoDataFrame is provided as input.
- normalize_by_area : bool, optional – if True, the value will be transformed to "value m$^{-2}$", assuming the input was "value per grid-cell".
- zero_is_value: bool, optional - if True, treats zero values as valid data rather than as no-data. Default is False.

- verbose: bool, optional – default=False, but if it is True, the global gridded statistics (e.g., sum) of before and after re-gridding operation will be printed.

**Output and behavior**

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

**Example: line_2_grid**

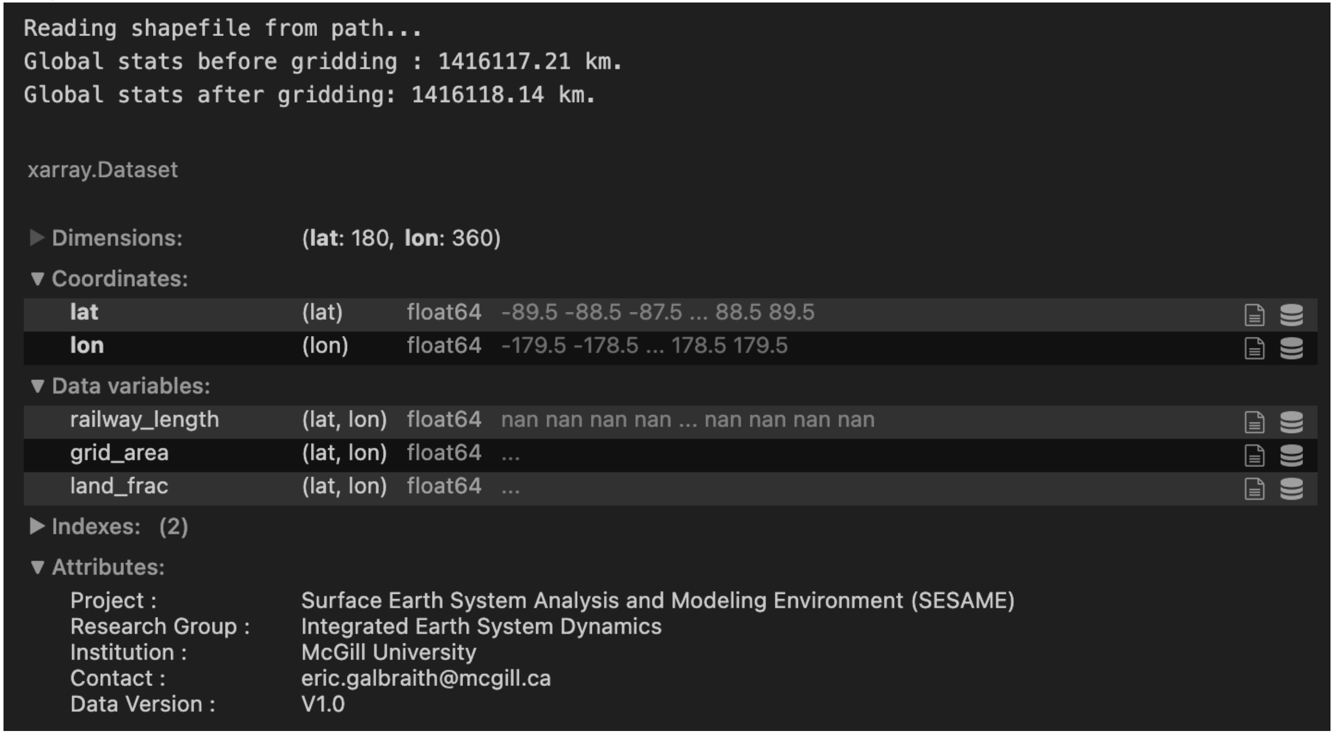


```
# Line to Grid
import sesame as ssm

shapefile_path = os.path.join("data", "Global_Railways_WFP.shp")
variable_name = 'railway_length'
long_name = 'Total Railway Length in km'
source = 'Global Railways (WFP–World Food Programme SDI–T – Logistics Database)'
output_directory = os.path.join("data/")
output_filename = "line_2_grid"

ssm.line_2_grid(lines=shapefile_path, variable_name=variable_name, long_name=long_name,
units="meter/grid-cell", source=source, time=None, resolution=1, agg_column=None,
agg_function="sum", attr_field=None,output_directory=output_directory,
output_filename=output_filename, normalize_by_area=None, zero_is_value=False,verbose="yes")
```

**Output:**



```
Reading shapefile from path...
Global stats before gridding : 1416117.21 km.
Global stats after gridding: 1416118.14 km.


xarray.Dataset

▶ Dimensions:          (lat: 180, lon: 360)
▼ Coordinates:
    lat                (lat)        float64  -89.5 -88.5 -87.5 ... 88.5 89.5
    lon                (lon)        float64  -179.5 -178.5 ... 178.5 179.5
▼ Data variables:
    railway_length     (lat, lon)   float64  nan nan nan nan ... nan nan nan nan
    grid_area          (lat, lon)   float64  ...
    land_frac          (lat, lon)   float64  ...
▶ Indexes:  (2)
▼ Attributes:
    Project :          Surface Earth System Analysis and Modeling Environment (SESAME)
    Research Group :   Integrated Earth System Dynamics
    Institution :      McGill University
    Contact :          eric.galbraith@mcgill.ca
    Data Version :     V1.0
```

## 1.4 poly_2_grid

*poly_2_grid(polygons, variable_name='variable', long_name='variable', units="m2/grid-cell", source=None, time=None, resolution=1, attr_field=None, fraction=False, agg_function="sum", output_directory=None, output_filename=None, normalize_by_area=False, zero_is_value=False, verbose=False)*

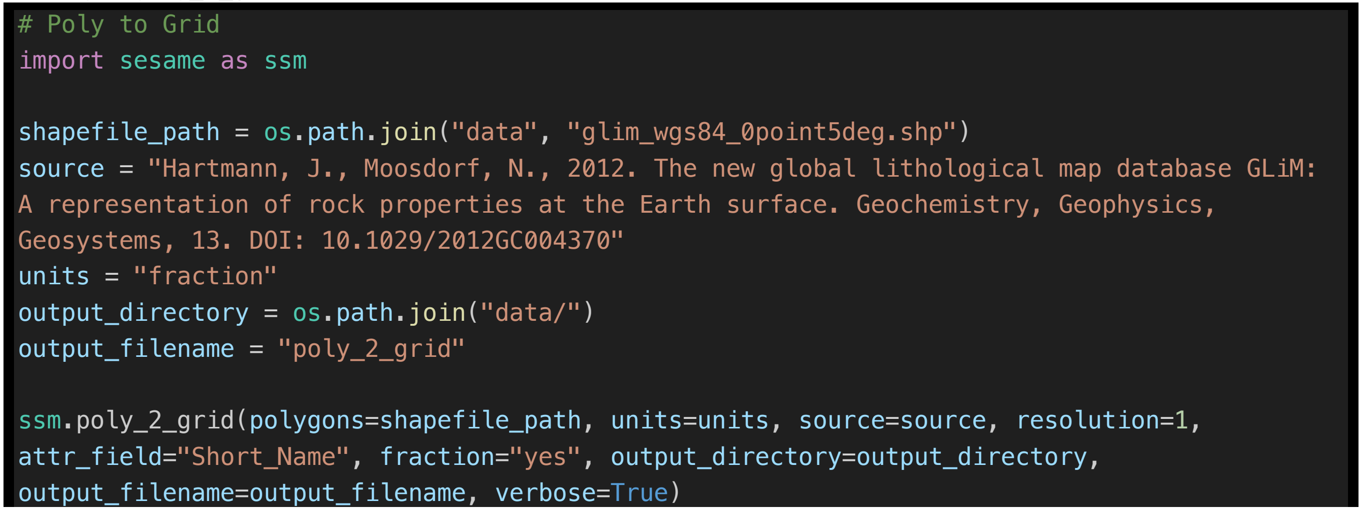
**Input Arguments**

- polygons: GeoDataFrame or str – input polygons data to be gridded. Can be either a GeoDataFrame or a path to a polygons shapefile (.shp).
- variable_name: str, optional – variable_name description to be added to netcdf attributes metadata. If not specified, then "variable" will be used as the variable name. If attr_field is specified, the variable_name will be the same as the corresponding entries in the unique field column in the shapefile/geodataframe.
- long_name: str, optional - long_name description to be added to netcdf attributes metadata. It behaves the same as variable_name.
- units: str, 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.
- resolution: float, optional – argument of the desired cell resolution in degree. The default is 1.
- attr_field: str, optional – argument, if the input shapefile has multiple types of data that the user wants to be stored as separate data variables in the final gridded dataset. If the column name is given, each unique value within the column will be considered as a separate variable.
- 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.
- agg_function : str, optional - aggregation method for combining polygon area values in each grid cell. Default is 'sum'. Options include: 'sum', 'max', 'min', 'mean'.
- output_directory: str, optional – Directory where the output NetCDF file will be saved. If None, but output_filename is True, the file will be saved in the current working directory.
- output_filename : str, optional – Name of the output NetCDF file (without the `.nc` extension). If not provided:
- output_directory: str, optional - Directory where the output NetCDF file will be saved. If None, but output_filename is True, the file will be saved in the current working directory.
- output_filename : str, optional - Name of the output NetCDF file (without the `.nc` extension). If not provided:
  - Uses the input shapefile name if a shapefile path is given.
  - Saves as `"gridded_polygons.nc"` if a GeoDataFrame is provided as input.
- normalize_by_area : bool, optional – if True, the value will be transformed to "value m$^{-2}$", assuming the input was "value per grid-cell".
- zero_is_value: bool, optional - if True, treats zero values as valid data rather than as no-data. Default is False.
- verbose: bool, optional – default=False, but if it is True, the global gridded statistics (e.g., sum) 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.

**Example: poly_2_grid**

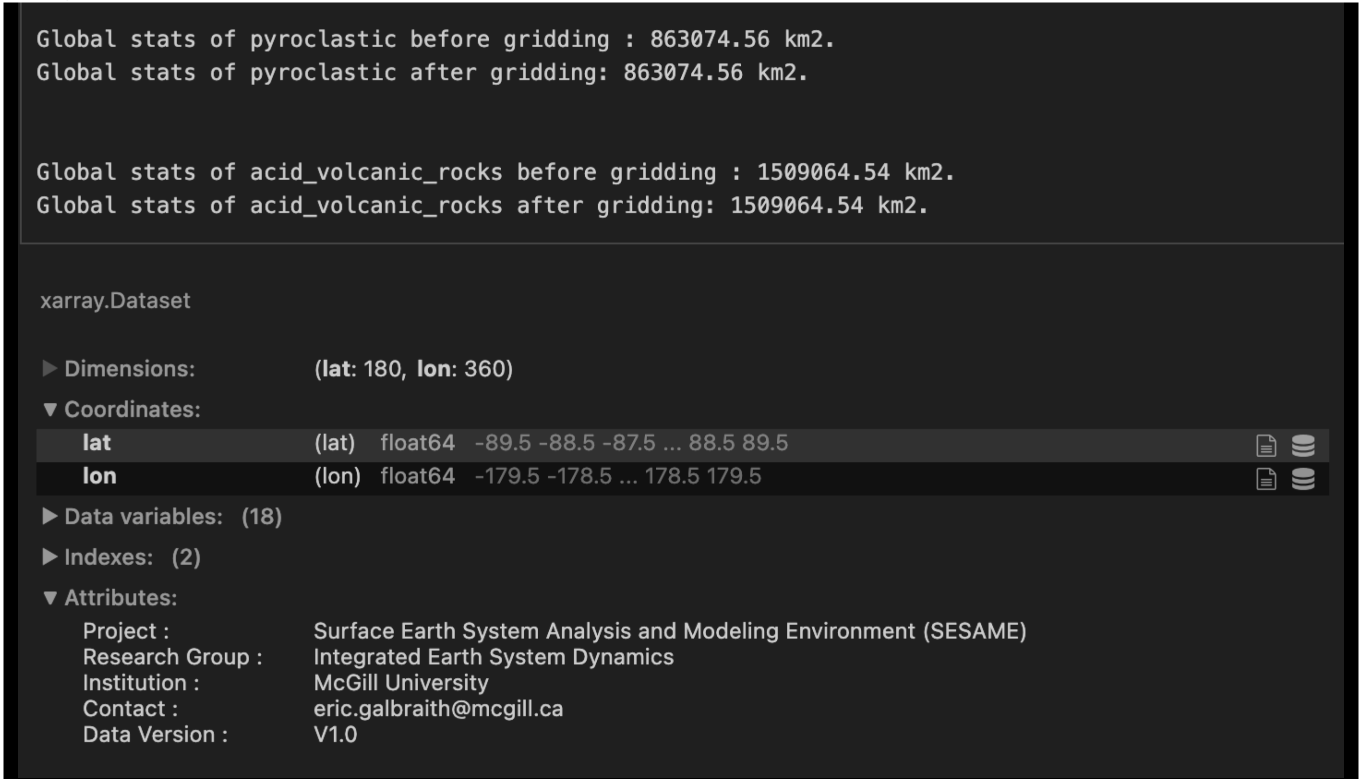


```
# Poly to Grid
import sesame as ssm

shapefile_path = os.path.join("data", "glim_wgs84_0point5deg.shp")
source = "Hartmann, J., Moosdorf, N., 2012. The new global lithological map database GLiM:
A representation of rock properties at the Earth surface. Geochemistry, Geophysics,
Geosystems, 13. DOI: 10.1029/2012GC004370"
units = "fraction"
output_directory = os.path.join("data/")
output_filename = "poly_2_grid"

ssm.poly_2_grid(polygons=shapefile_path, units=units, source=source, resolution=1,
attr_field="Short_Name", fraction="yes", output_directory=output_directory,
output_filename=output_filename, verbose=True)
```

**Output:**



```
Global stats of pyroclastic before gridding : 863074.56 km2.
Global stats of pyroclastic after gridding: 863074.56 km2.


Global stats of acid_volcanic_rocks before gridding : 1509064.54 km2.
Global stats of acid_volcanic_rocks after gridding: 1509064.54 km2.
```

```
xarray.Dataset

▶ Dimensions:          (lat: 180, lon: 360)
▼ Coordinates:
    lat               (lat)  float64  -89.5 -88.5 -87.5 ... 88.5 89.5
    lon               (lon)  float64  -179.5 -178.5 ... 178.5 179.5
▶ Data variables:  (18)
▶ Indexes:  (2)
▼ Attributes:
    Project :          Surface Earth System Analysis and Modeling Environment (SESAME)
    Research Group :   Integrated Earth System Dynamics
    Institution :      McGill University
    Contact :          eric.galbraith@mcgill.ca
    Data Version :     V1.0
```

## 1.5 table_2_grid

*table_2_grid(surrogate_file, surrogate_variable, tabular_file, tabular_column, variable_name=None, long_name=None, units="value/grid-cell", source=None, time=None, output_directory=None, output_filename=None, zero_is_value=False, normalize_by_area=False, eez=False, verbose=False)*

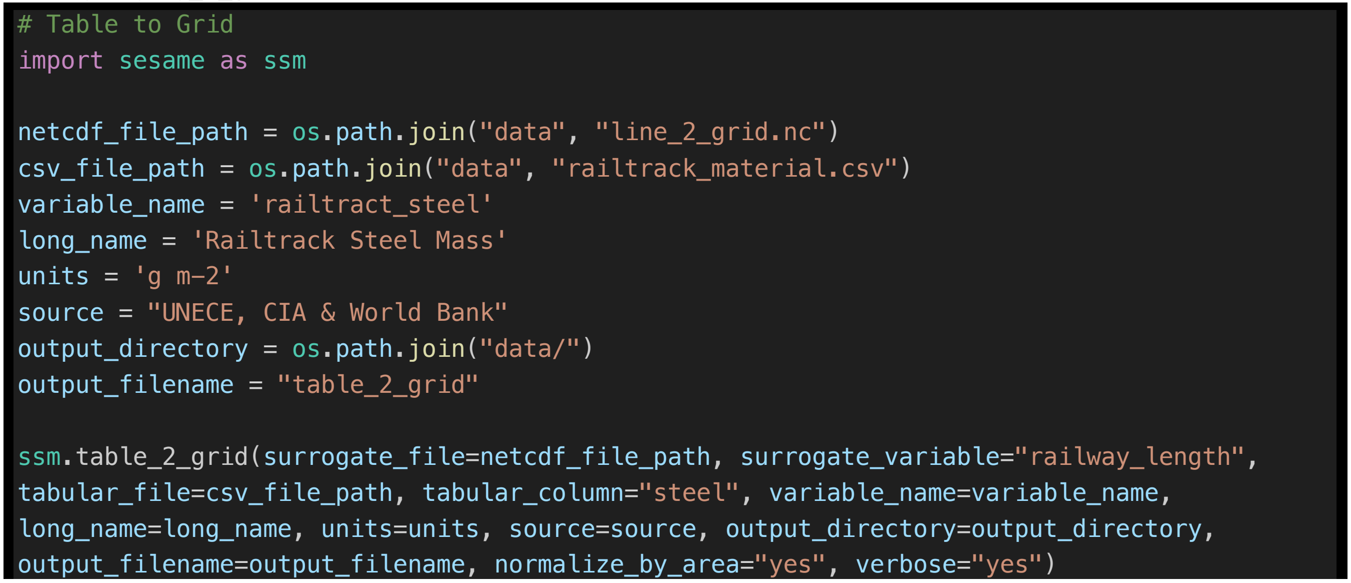
**Input Arguments**

- surrogate_file : xarray.Dataset or str – xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset. The dataset must include the variable specified in `surrogate_variable`.
- surrogate_variable : str – variable name in the NetCDF or xarray dataset used for spatial distribution.
- tabular_file : pandas.DataFrame or str – tabular dataset as a pandas DataFrame or a path to a CSV file. If a file path is provided, it will be automatically loaded into a DataFrame. The data must include a column named "ISO3" representing country codes. If not present, use the `add_iso3_column` utility function to convert country names to ISO3 codes.
- tabular_column : str – column name in the tabular dataset with values to be spatially distributed.
- variable_name: str, optional – 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 – variable description or long name to be added to the netcdf variable's attributes.
- units: str, optional – units of the variable to be added to the variable's attributes. Default is "value/grid-cell".
- source: str, optional – source of variable be added to the netcdf variable's attributes. Default is None.
- time: str, optional – A string time needs to be indicated. Based on the input time, the country fraction and surrogate variable (if available) time will be selected. If no time is mentioned, then latest time available from the surrogate variable and country fraction will be selected.
- output_directory: str, optional - Directory where the output NetCDF file will be saved. If None, but output_filename is True, the file will be saved in the current working directory.
- output_filename : str, optional - Name of the output NetCDF file (without the `.nc` extension). If not provided:
  - Uses `variable_name` if it is specified.
  - Falls back to `long_name` or `tabular_column` if `variable_name` is not given.
  - Defaults to `gridded_table.nc` if none of the above are provided.
- zero_is_value: bool, optional - if True, treats zero values as valid data rather than as no-data. Default is False.
- normalize_by_area : bool, optional – if True, the value will be transformed to "value $m^{-2}$", assuming the input was "value per grid-cell".
- eez: bool, optional - If set to True, the function converts the jurisdictional Exclusive Economic Zone (EEZ) values to a spatial grid.
- verbose: bool, optional – default=False, but if it is True, the global gridded statistics (e.g., 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 with their total percentage of coverage in respect to global total.

**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.

**Example: poly_2_grid**

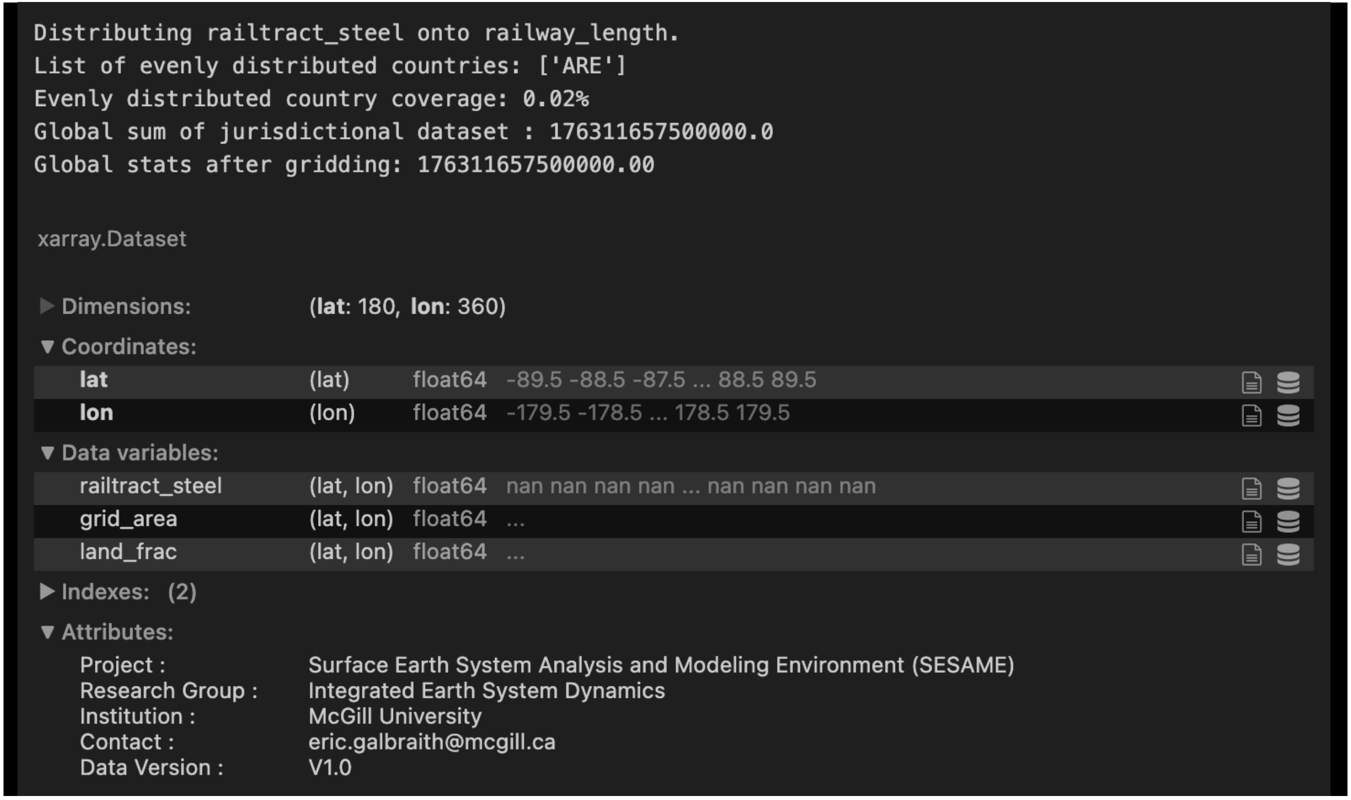


```
# Table to Grid
import sesame as ssm

netcdf_file_path = os.path.join("data", "line_2_grid.nc")
csv_file_path = os.path.join("data", "railtrack_material.csv")
variable_name = 'railtract_steel'
long_name = 'Railtrack Steel Mass'
units = 'g m-2'
source = "UNECE, CIA & World Bank"
output_directory = os.path.join("data/")
output_filename = "table_2_grid"

ssm.table_2_grid(surrogate_file=netcdf_file_path, surrogate_variable="railway_length",
tabular_file=csv_file_path, tabular_column="steel", variable_name=variable_name,
long_name=long_name, units=units, source=source, output_directory=output_directory,
output_filename=output_filename, normalize_by_area="yes", verbose="yes")
```

**Output:**



```
Distributing railtract_steel onto railway_length.
List of evenly distributed countries: ['ARE']
Evenly distributed country coverage: 0.02%
Global sum of jurisdictional dataset : 176311657500000.0
Global stats after gridding: 176311657500000.00

xarray.Dataset

▶ Dimensions:        (lat: 180, lon: 360)
▼ Coordinates:
    lat              (lat)       float64   -89.5 -88.5 -87.5 ... 88.5 89.5
    lon              (lon)       float64   -179.5 -178.5 ... 178.5 179.5
▼ Data variables:
    railtract_steel  (lat, lon)  float64   nan nan nan nan ... nan nan nan nan
    grid_area        (lat, lon)  float64   ...
    land_frac        (lat, lon)  float64   ...
▶ Indexes:  (2)
▼ Attributes:
    Project :          Surface Earth System Analysis and Modeling Environment (SESAME)
    Research Group :   Integrated Earth System Dynamics
    Institution :      McGill University
    Contact :          eric.galbraith@mcgill.ca
    Data Version :     V1.0
```

## 2. Additional functions

### 2.1 grid_2_table

*grid_2_table(grid_file, variables=None, time=None, grid_area=None, resolution=1, aggregation=None, agg_function='sum', verbose=False)*

**Input Arguments**

- grid_file : xarray.Dataset or str - xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset.
- 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.

### 2.2 plot_histogram

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

**Input Arguments**

- dataset : xarray.Dataset or str, xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset.
- 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.
- plot_title: str, optional, the title for the plot.
- x_label: str, optional, the label for the x-axis.
- y_label: str, optional, the label for the y-axis.
- remove_outliers: bool, optional, whether to remove outliers.
- log_transform: str, optional, the type of log transformation ('log10', 'log', 'log2').
- output_dir : str, optional, Directory path to save the output figure. If not provided, the figure is saved in the current working directory.
- filename : str, optional, Filename (with extension) for saving the figure. If not provided, the plot is saved as "output_histogram.png".

**Output and behavior**

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

### 2.3 plot_scatter

*plot_scatter(variable1, variable2, dataset, 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)*

**Input Arguments**

- variable1 : str, name of the variable to be plotted on the x-axis. Must be present in `dataset`.
- variable2 : str, name of the variable to be plotted on the y-axis. If `dataset2` is provided, this variable will be extracted from `dataset2`; otherwise, it must exist in `dataset`.
- dataset : xarray.Dataset or str, the primary dataset or a path to a NetCDF file. This dataset must contain the variable specified by `variable1`, which will be used for the x-axis.
- dataset2 : xarray.Dataset or str, optional, a second dataset or a path to a NetCDF file containing the variable specified by `variable2` (for the y-axis). If not provided, `dataset` will be used for both variables.
- color: str, optional, the color map of the scatter plot.
- 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').
- equation : bool, optional, ff True, fits and displays a linear regression equation.
- output_dir : str, optional, Directory path to save the output figure. If not provided, the figure is saved in the current working directory.
- filename : str, optional, Filename (with extension) for saving the figure. If not provided, the plot is saved as "output_scatter.png".

**Output and behavior**

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

### 2.4 plot_time_series

*plot_time_series(dataset, variable,, agg_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)*

**Input Arguments**

- dataset : xarray.Dataset or str, xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset.
- variable: str, the name of the variable to plot.
- agg_function: str, the operation to apply ('sum', 'mean', 'max', 'std').
- 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'.
- smoothing_window: int, optional, the window size for rolling mean smoothing.
- output_dir : str, optional, Directory path to save the output figure. If not provided, the figure is saved in the current working directory.
- filename : str, optional, Filename (with extension) for saving the figure. If not provided, the plot is saved as "output_time_series.png".

**Output and behavior**

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

### 2.5 plot_hexbin

*plot_hexbin(variable1, variable2, dataset, 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)*

**Input Arguments**

- variable1 : str, name of the variable to be plotted on the x-axis. Must be present in `dataset`.
- variable2 : str, name of the variable to be plotted on the y-axis. If `dataset2` is provided, this variable will be extracted from `dataset2`; otherwise, it must exist in `dataset`.
- dataset : xarray.Dataset or str, the primary dataset or a path to a NetCDF file. This dataset must contain the variable specified by `variable1`, which will be used for the x-axis.
- dataset2 : xarray.Dataset or str, optional, a second dataset or a path to a NetCDF file containing the variable specified by `variable2` (for the y-axis). If not provided, `dataset` will be used for both variables.
- 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').
- output_dir : str, optional, Directory path to save the output figure. If not provided, the figure is saved in the current working directory.
- filename : str, optional, Filename (with extension) for saving the figure. If not provided, the plot is saved as "output_hexbin.png".

**Output and behavior**

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

### 2.6 plot_map

*plot_map(variable, dataset, color='hot_r', title='', label='', vmin=None, vmax=None, extend_min=False, extend_max=False, levels=10, out_bound=True, remove_ata=False, output_dir=None, filename=None)*

**Input Arguments**

- variable : str. Name of the variable in the xarray Dataset to plot.

- dataset : xarray.Dataset or str, xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset.
- color : str, default 'hot_r'. Matplotlib colormap name for the plot (discrete color scale).
- title : str, default '', Title of the map.
- label : str, default ''. Label for the colorbar.
- vmin : float, optional. Minimum data value for the colorbar range. If not provided, the minimum of the variable is used.
- vmax : float, optional. Maximum data value for the colorbar range. If not provided, the maximum of the variable is used.
- extend_min : bool, default False. If True, includes values below `vmin` in the first color class and shows a left arrow on the colorbar.
- extend_max : bool, default False. If True, includes values above `vmax` in the last color class and shows a right arrow on the colorbar.
- levels : int or list of float, default 10. Either the number of color intervals or a list of explicit interval boundaries.
- out_bound : bool, default True. Whether to display the outer boundary (spine) of the map projection.
- remove_ata : bool, default False. If True, removes Antarctica from the map by excluding data below 60°S latitude.
- output_dir : str, optional - directory path to save the output figure. If not provided, the figure is saved in the current working directory.
- filename : str, optional - filename (with extension) for saving the figure. If not provided, the plot is saved as " output_plot.png".
- show: bool, True. Whether or not show the map

**Notes**

- If both `extend_min` and `extend_max` are False, the dataset is clipped strictly within [vmin, vmax].
- The colorbar will use arrows to indicate out-of-bound values only if `extend_min` or `extend_max` is True.
- Tick formatting on the colorbar is: two decimal places if (vmax - vmin) <= 10.
- If `remove_ata` is True, the colorbar is placed slightly higher to avoid overlap with the map.

**Output and behavior**

Plots a 2D map of a variable from an xarray Dataset or NetCDF file with customizable colorbar, projection, and map appearance. Returns axes class of the map.

### 2.7 plot_country

*plot_country(tabular_file, column, title="", label="", color='viridis', levels=10, output_dir=None, filename=None, csv_path=None, remove_ata=False, out_bound=True, vmin=None, vmax=None, extend_min=False, extend_max=False)*

**Input Arguments**

- tabular_file : pandas.DataFrame or str. Input table containing country-level data. Can be either a pandas DataFrame with the required `column` or a string path to a CSV file, which will be automatically read into a DataFrame.
- column : str. Name of the column in the dataframe to visualize.

- title : str, optional. Title of the map. Default is an empty string.
- label : str, optional. Label for the colorbar. Default is an empty string.
- color : str, optional. Name of the matplotlib colormap to use. Default is 'viridis'.
- levels : int or list of float, optional. Number of color levels (if int) or list of bin edges (if list). Default is 10.
- output_dir : str, optional - directory path to save the output figure. If not provided, the figure is saved in the current working directory.
- filename : str, optional - filename (with extension) for saving the figure. If not provided, the plot is saved as " output_country_plot.png".
- csv_path : str or None, optional. Path to CSV file containing the data. Used if `dataframe` is not provided.
- remove_ata : bool, optional. Whether to remove Antarctica ('ATA') from the data. Default is False.
- out_bound : bool, optional. Whether to display map boundaries (spines). Default is True.
- vmin : float or None, optional. Minimum value for the colormap. If None, calculated from the data.
- vmax : float or None, optional. Maximum value for the colormap. If None, calculated from the data.
- extend_min : bool, optional. Whether to extend the colorbar below `vmin`. Default is False.
- extend_max : bool, optional. Whether to extend the colorbar above `vmax`. Default is False.

**Notes**

- If both `extend_min` and `extend_max` are False, the dataset is clipped strictly within [vmin, vmax].
- The colorbar will use arrows to indicate out-of-bound values only if `extend_min` or `extend_max` is True.
- Tick formatting on the colorbar is: two decimal places if (vmax - vmin) <= 10.
- If `remove_ata` is True, the colorbar is placed slightly higher to avoid overlap with the map.

**Output and behavior**

Plots a choropleth map of countries using a specified data column and a world shapefile. Displays the map and optionally saves it to a file.

### 2.8 sum_variables

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

**Input Arguments**

- dataset: xarray.Dataset, or str, xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset.
- variables: list of str, the names of the variables to sum. If None, sum all variables except those starting with 'grid_area' and ‘land_frac’.
- 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.

**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.

### 2.9 subtract_variables

*subtract_variables(variable1, variable2, dataset, new_variable_name=None, time=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, or str, xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset.
- 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.

**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.

### 2.10 divide_variables

*divide_variables(variable1, variable2, dataset, new_variable_name=None, time=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, or str, xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset.
- 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.

**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.

### 2.11 multiply_variables

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

**Input Arguments**

- dataset: xarray.Dataset, or str, xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset.
- variables: list of str, optional, the names of the variables to multiply. If None, multiply all variables except those starting with 'grid_area' and ‘land_frac’.
- 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.

**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.

2.12 average_variables

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

**Input Arguments**

- dataset: xarray.Dataset, or str, xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.Dataset.
- variables: list of str, optional, the names of the variables to average. If None, average all variables except those starting with 'grid_area' and 'land_frac'.
- 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.

**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.

2.13 get_netcdf_info

*get_netcdf_info(netcdf_path, variable_name=None)*

**Input Arguments**

- netcdf_path : xarray.Dataset or str. xarray dataset or a path to a NetCDF file. If a file path is provided, it will be automatically loaded into an xarray.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.

2.14 add_iso3_column

*add_iso3_column(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'.