Chapter 1

Introduction

1.1 WorldClim Data

WorldClim is a widely-used database that provides high-resolution global climate data, covering a range of variables such as temperature, precipitation, solar radiation, wind speed, and water vapor pressure. This data has been invaluable for researchers in fields like ecology, conservation, and climate change studies, enabling detailed spatial modeling and mapping applications [1][2][3].

The WorldClim dataset is generated by interpolating monthly climate observations from weather stations around the world onto a high-resolution grid, typically at 30 arc-second (1 km^2) or coarser resolutions [4][5]. This interpolation process allows for the creation of continuous climate surfaces from the discrete station data points. However, in areas with sparse weather station coverage, the interpolation may introduce uncertainties and potential errors in the final gridded climate layers.

For my final-year project, I aim to investigate the validity of the ”no data” values present in the WorldClim database. These no data cells represent grid locations where the interpolation algorithm was unable to generate a reliable estimate, often due to a lack of nearby weather station observations. Understanding the distribution and characteristics of these no data regions is crucial for assessing the overall quality and limitations of the WorldClim dataset, which is widely used in research and applications.

By analyzing the patterns and potential biases in the no data values, I hope to provide insights that will help researchers better understand the strengths and weaknesses of the WorldClim data, and guide them in selecting appropriate use cases and interpretation of the results derived from this important climate dataset.

1.2 Rasterio Package

Rasterio is a Python package that provides a high-level interface for reading, writing, and manipulating geospatial raster data. It is built on top of the Geospatial Data Abstraction Library (GDAL) and aims to provide a more Pythonic and user-friendly API compared to GDAL's Python bindings. Key features of the Rasterio package include:

- Reading and Writing Raster Data: Rasterio can read and write a variety of raster data formats, including GeoTIFF, Erdas Imagine, and others. It provides a simple and intuitive interface for accessing the raster data as NumPy arrays.

- Metadata Handling: Rasterio automatically extracts and exposes the spatial metadata associated with raster datasets, such as the coordinate reference system, geotransform, and nodata values.

- Raster Processing: The package includes functions for performing common raster operations, such as reprojection, resampling, and applying arithmetic operations on raster data.

- Plotting: Rasterio integrates with Matplotlib to provide built-in functions for visualizing raster data.

- In-Memory Files: Rasterio supports working with raster data stored in memory, without the need for a physical file on disk.

- Migrating from GDAL: Rasterio provides a more Pythonic interface compared to GDAL's Python bindings, making it easier for users to migrate their existing GDAL-based code to Rasterio.

The Rasterio package is designed to be fast, efficient, and easy to use, while still providing access to the full capabilities of the underlying GDAL library. It is widely used in the geospatial Python community for a variety of applications, including remote sensing, GIS, and environmental modeling.

Chapter 2

Exploring the WorldClim data

For the research, we use the 5 minute data. We analyse bioclimatic variable data(bio), precipitation data(prec), annual average temperature(tavg), annual maximum temperature(tmax), and annual minimum temperature(tmin).