

3D And Time Geoviz Assignment Workflow

Starting with data from Movebank, I began by downloading the dataset in CSV format with UTM coordinates. Using Python's pandas library, I loaded the data into a DataFrame for manipulation. With plotly, I prepared to visualize the data in an interactive 3D space-time cube format.

The initial step involved parsing the timestamps from a string to a datetime object using pandas and then converting these to Unix time. This format is necessary for temporal axis representation in the 3D scatter plot. After verifying the conversions, I identified unique identifiers within the *'individual-local-identifier'* column to distinguish between different bird classes – specifically, *'anosmic'*, *'magnetic'*, and *'control'*.

For each bird class, I filtered the data and generated a 3D scatter plot, color-coding individual bird trajectories using *plotly.express's qualitative color palette for clarity*. Each class was plotted separately to maintain distinct visual separation, and I added an interactive legend for user-friendly navigation.

Ultimately, the visualizations were saved as HTML files, which allows for easy sharing and interaction with the data. Throughout this process, scripting automated data preparation and visualization, while manual checks ensured accuracy and coherence in the presentation.

REFERENCES

pandas—*Python Data Analysis Library*. (n.d.). Retrieved 20 March 2024, from

<https://pandas.pydata.org/>

Unixtime.org. (n.d.). *Unix TimeStamp—Epoch Converter—TimeStamp Converter*. Retrieved 20

March 2024, from <https://unixtime.org>

Plotly. (n.d.). Retrieved 20 March 2024, from <https://plotly.com/python/plotly-express/>