**Port & Sail Calculation**

**Comprehensive Maritime Voyage Event Analysis**

**Objective:** This report aims to analyze maritime voyage data using both SQL and Python, focusing on extracting, transforming, and analyzing temporal and spatial aspects of voyage events ('SOSP' and 'EOSP'). The goal is to generate insights into voyage efficiencies and patterns, including duration calculations and distance traveled between ports.

**Expanded Requirements:**

1. **SQL Query Enhancements:**
   * **Data Extraction:** Retrieve data for a specified vessel (imo\_num) and voyage (voyage\_Id), excluding records with non-null allocatedVoyageId.
   * **Date-Time Calculation:** Convert dateStamp and timeStamp to precise UTC datetime for each event.
   * **Voyage Segmentation:** Identify and segment voyage stages using 'SOSP' (Start of Sea Passage) and 'EOSP' (End of Sea Passage) events.
   * **Duration Calculation:** Compute cumulative sailing time and port stay duration for each voyage segment.
   * **Geographic Movement:** Calculate the distance between consecutive ports based on latitude and longitude data.
2. **Python Programming Task Enhancements:**
   * **Data Processing:** Implement Python functions to replicate SQL query logic for data processing.
   * **Distance Calculation:** Develop a function using the Haversine formula to compute distances between geographic coordinates.
   * **Data Visualization:** Utilize matplotlib to plot the timeline of voyage events and durations for visual insights.
   * **Error Handling:** Incorporate robust error handling and data validation to ensure script reliability.

**Sample Data Enhancement:**

The voyages table is extended with latitude (lat) and longitude (lon) data for ports, enabling geographic calculations between consecutive ports.

**Enhanced Expected Output:**

In addition to basic output fields (id, event, event\_datetime, voyage\_From, lat, lon, imo\_num, voyage\_Id), the following fields are included:

* **distance\_travelled**: Distance in nautical miles between the current and previous port.
* **sailing\_time**: Duration spent sailing from the last port to the current port.
* **port\_stay\_duration**: Duration spent at the current port before departure.

**Advanced Deliverables:**

1. **Enhanced SQL Query:** A sophisticated SQL query that retrieves required information and calculates distances and durations.
2. **Advanced Python Script:** A script replicating SQL tasks, including geographic calculations and visualization of voyage timelines.
3. **Extended Documentation:** Detailed explanation of calculations, assumptions, methods, and visualizations.

**Evaluation Criteria:**

* **Complexity and Depth:** Demonstrates a comprehensive understanding of SQL and Python capabilities in handling complex maritime data scenarios.
* **Accuracy and Performance:** Outputs are accurate, efficiently computed, and considerate of large datasets.
* **Clarity, Documentation, and Innovation:** Clear, well-documented code with innovative solutions for distance calculations and data visualization, along with thorough explanations.

This task simulates a practical data analysis scenario, testing proficiency in integrating data handling, geographic calculations, and visualization to provide valuable insights into maritime operations.