Bird Monitoring Data Analysis

**Project Overview**

This project focuses on the exploratory data analysis (EDA) of bird monitoring datasets. The objective is to clean, preprocess, and visualize the data to gain insights into bird populations, their behaviors, and environmental factors that influence them. The analysis leverages Python libraries such as pandas, numpy, and visualization tools.

**Objectives**

* Clean and preprocess bird monitoring datasets to ensure consistency and accuracy.
* Perform exploratory data analysis to uncover trends and patterns.
* Visualize data for better understanding and communication of insights.
* Provide actionable insights for conservation efforts and biodiversity studies.

**Dataset Description**

**Source**

The data used in this project comes from Excel files containing bird monitoring information.

**Key Columns**

The datasets include the following key columns:

* **Admin\_Unit\_Code**: Administrative unit for the observation location.
* **Plot\_Name**: Specific plots where observations were made.
* **Common\_Name**: Common names of bird species observed.
* **Scientific\_Name**: Scientific names of bird species.
* **AcceptedTSN**: Taxonomic Serial Number for validation.
* **Temperature**: Weather condition during observations.
* **Flyover\_Observed**: Indicator of flyover sightings.
* **Sex**: Gender of observed birds (if identified).

**Workflow**

**1. Data Cleaning**

The data cleaning process addressed missing values, inconsistencies, and formatting issues:

* Missing values in the AcceptedTSN column were replaced with a default value (997805).
* Null or empty fields were identified using the .isnull() function and systematically handled.
* Unnecessary columns were removed to focus on relevant features for analysis.
* The Sub\_Unit\_Code column was dropped due to an excess of null values, making it uninformative for the analysis.
* Boolean columns (Flyover\_Observed, PIF\_Watchlist\_Status, Regional\_Stewardship\_Status, Previously\_Obs, and Initial\_Three\_Min\_Cnt) were converted into binary format (0 and 1) for consistency and easier analysis.
* **Interval\_Length Column Conversion**: The Interval\_Length column contained ranges in text format (e.g., "5-10 min"). These were converted into numerical midpoints using the following method:
* def convert\_to\_number(interval):
* try:
* interval = interval.replace(' min', '').strip()
* start, end = interval.split('-')
* return (float(start.strip()) + float(end.strip())) / 2
* except ValueError:
* return None

Example: "5-10 min" was converted to 7.5.

* **Distance Column Conversion**: The Distance column was cleaned and converted into numeric values. It handled ranges, inequalities, and exact values using the following method:
* def convert\_to\_numeric(distance):
* distance = distance.replace(' Meters', '').strip()
* if '<=' in distance:
* return 50 # Example default value for "<= 50 Meters"
* elif '-' in distance:
* start, end = distance.split(' - ')
* return (float(start) + float(end)) / 2
* else:
* return float(distance)

Example: "50 - 100 Meters" was converted to 75.0.

**2. Outlier Detection and Handling**

Outliers were identified and addressed to ensure accurate analysis:

* **Detection**: Used boxplots and the interquartile range (IQR) method to identify outliers in numerical columns such as Temperature and Humidity.
* **Fixing**: Outliers were either capped and floored to the nearest valid values within the IQR range or removed entirely if deemed erroneous.

**3. Dropping Unnecessary Columns**

Certain columns were dropped to streamline the dataset and focus on analysis-relevant information:

* Columns with minimal variance or irrelevance to the analysis objectives were removed.
* Examples include redundant identifiers or metadata columns that did not contribute to the insights.
* The Sub\_Unit\_Code column was specifically removed due to having a high proportion of null values, rendering it unhelpful for meaningful analysis.

**4. Exploratory Data Analysis (EDA)**

EDA steps included:

* Inspecting unique values in categorical columns such as Admin\_Unit\_Code.
* Generating descriptive statistics for numerical columns (e.g., Temperature).
* Counting and visualizing the distribution of bird observations based on various attributes, such as Common\_Name and Flyover\_Observed.

**5. Data Visualization**

Visualization techniques included:

* **Bar Charts**: To display the frequency of bird species.
* **Pie Charts**: To represent the proportion of flyover vs. non-flyover observations.
* **Line Graphs**: To explore seasonal trends in observations.

**Tools and Libraries**

The following tools and libraries were used:

* **Python**:
  + pandas for data manipulation and analysis.
  + numpy for numerical computations.
  + matplotlib and seaborn for data visualization.
* **Excel**: Initial data storage and structure.

**Types of Analysis**

1. **Temporal Analysis**

* **Seasonal Trends: Analyze the Date and Year columns to detect patterns in bird sightings across different seasons or years.**
* **Observation Time: Study the Start\_Time and End\_Time to determine if specific time windows correlate with higher bird activity.**

1. **Spatial Analysis**

* **Location Insights: Group data by Location\_Type (e.g., Grassland) to identify biodiversity hotspots.**
* **Plot-Level Analysis: Compare observations across different Plot\_Name to see which plots attract more species or specific kinds of birds.**

**3. Species Analysis**

* **Diversity Metrics: Count unique species (Scientific\_Name) observed and their distribution across Location\_Type.**
* **Activity Patterns: Check the Interval\_Length and ID\_Method columns to identify the most common activity types (e.g., Singing).**
* **Sex Ratio: Analyze the Sex column to understand the male-to-female ratio for different species.**

**4. Environmental Conditions**

* **Weather Correlation: Explore how Temperature, Humidity, Sky, and Wind impact observations, such as the number of birds or their distances.**
* **Disturbance Effect: Assess the impact of Disturbance (e.g., slight effect) on bird sightings.**

**5. Distance and Behavior**

* **Distance Analysis: Evaluate the Distance column to identify species typically observed closer or farther from the observer.**
* **Flyover Frequency: Examine the Flyover\_Observed column to detect trends in bird behavior during observation.**

**6. Observer Trends**

* **Observer Bias: Analyze data by Observer to check if specific individuals report more observations or certain species.**
* **Visit Patterns: Evaluate the Visit column to see how repeated visits affect species count or diversity.**

**7. Conservation Insights**

* **Watchlist Trends: Use the PIF\_Watchlist\_Status and Regional\_Stewardship\_Status to identify trends in species that are at risk or require conservation focus.**
* **AOU Code Patterns: Study the distribution of species based on their AOU\_Code to correlate with regional or national conservation priorities.**
* **Flyover Observations**: Flyover sightings account for a small percentage (~4.5%) of total observations, while the majority are non-flyover.
* **Species Diversity**: Common bird species include Eastern Bluebirds, Red-tailed Hawks, and Northern Cardinals.
* **Environmental Factors**: Weather conditions like temperature and humidity significantly influence bird behavior and visibility.

**Challenges**

* Handling missing or inconsistent data entries.
* Optimizing large datasets for analysis in memory-constrained environments.
* Identifying and addressing outliers without compromising data integrity.

**Conclusion**

This project successfully cleaned and analyzed bird monitoring data, uncovering valuable insights into bird populations and behaviors. The findings can inform conservation strategies and enhance biodiversity studies.

**Future Work**

* Extend the analysis to include more datasets from different geographical regions.
* Incorporate machine learning models to predict bird activity based on environmental conditions.