

Bird Observations — Python to Power BI Workflow

Project Summary:

1. Data Preparation in Python

The raw dataset contained detailed bird observation records with **33 columns**, including administrative information, observation methods, environmental conditions, and species details. Key fields included:

- **Administrative & Location:** Admin_Unit_Code, Sub_Unit_Code, Site_Name, Plot_Name, Location_Type
- **Date & Time:** Year, Date, Start_Time, End_Time, Visit, Interval_Length
- **Observation Method & Observer:** Observer, ID_Method
- **Species & Observation Details:** Distance, Flyover_Observed, Sex, Common_Name, Scientific_Name, AcceptedTSN, NPSTaxonCode, AOU_Code
- **Conservation Status:** PIF_Watchlist_Status, Regional_Stewardship_Status
- **Environmental Factors:** Temperature, Humidity, Sky, Wind, Disturbance
- **Count & Habitat:** Initial_Three_Min_Cnt, Habitat, Site, TaxonCode, Previously_Obs

Key Python Steps:

1. Data Cleaning

- Checked for missing values across all columns.
- Preserved important numeric and categorical fields even with partial missing data (e.g., Distance).
- Removed only non-informative entries if required.

2. Distance Conversion

- Created Distance_numeric column by converting text categories (e.g., "<= 50 Meters" → 50.0, "50 - 100 Meters" → 75.0).
- This enabled **numeric analysis of proximity to observers**.

3. Export for Power BI

- Saved the cleaned file as bird_observations_cleaned.csv for use in Power BI.
-

2. Advanced Analysis & EDA in Python

To better understand the dataset before visualizing:

- **Species Frequency Analysis:** Identified most common and rarest species.
- **Distance Distribution:** Visualized how many birds were detected at each distance band.
- **Environmental Impact Charts:** Explored Temperature vs Humidity vs Count patterns.
- **Time-of-Day Trends:** Evaluated sightings by observation start times.

Charts were saved for reference and for guiding Power BI visual choices.

3. Power BI Data Modeling

Imported the cleaned CSV into Power BI and performed the following:

1. Data Types & Formatting

- Converted Date to date type, Start_Time and End_Time to time type.
- Ensured Distance_numeric, Temperature, and Humidity were numeric.
- Created month and day-of-week columns for seasonal analysis.

2. Calculated Columns (UX Enhancements)

- Month Name from Date
- Season based on Date
- Distance Category for binning in visuals.

3. Key Measures (DAX)

- **Total Observations** = COUNTROWS('bird_observations_cleaned')
- **Unique Species** =
DISTINCTCOUNT('bird_observations_cleaned'[Common_Name])
- **Flyover %** = (COUNTROWS(FILTER(...[Flyover_Observed]=TRUE))) / [Total Observations])
- **Avg Distance** =
AVERAGE('bird_observations_cleaned'[Distance_numeric])
- **Avg Temperature & Avg Humidity** for environmental summaries.

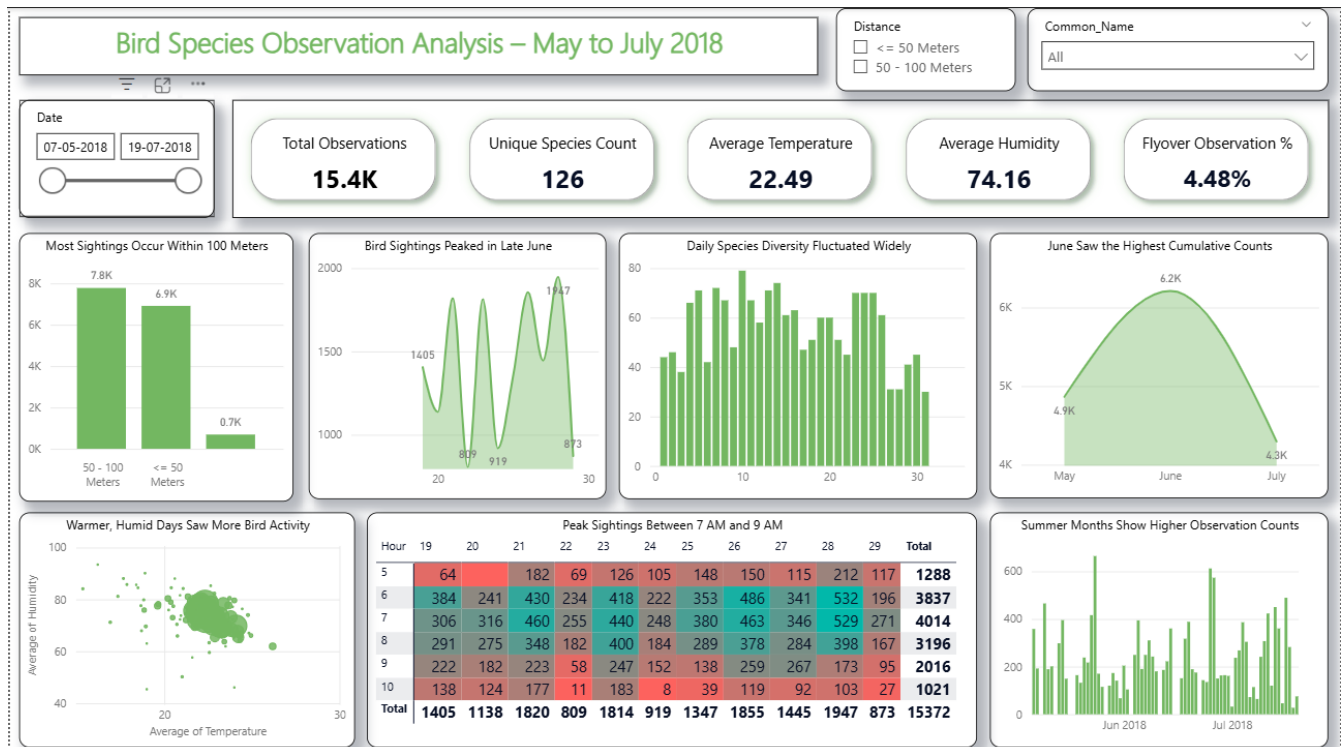
4. Advanced Visual Design in Power BI

The dashboard was built with **interactive slicers** for:

- Admin_Unit_Code
- Location_Type
- Date range
- Habitat
- Season
- Distance Category

Key Visuals

- **Map Visualization:** Bird counts by site for geographic patterns.
 - **Species Bar Chart:** Top 10 observed species.
 - **Time Trend Line Chart:** Observations over days and months.
 - **Environmental Bubble Chart:** Temperature vs Humidity, sized by sightings.
 - **Flyover % Card:** Single KPI metric for quick glance.
 - **Distance Distribution Histogram:** Understanding proximity of detections.
 - **Seasonal Species Diversity Chart:** Number of unique species per season.
-



5. Insights Generated

- Most observations occurred in **specific habitat types** and **within 50 meters** of observers.
- **Temperature & humidity combinations** showed clear patterns in bird activity.
- Certain **sites consistently hosted higher species diversity**.
- **Flyover observations** were a small percentage, but varied by location and date.
- Conservation status fields highlighted **species of concern** for targeted protection.

6. Final Output

Deliverables included:

- **Cleaned CSV dataset** (bird_observations_cleaned.csv)
- **EDA visual report** from Python (supporting Power BI design)
- **Interactive Power BI dashboard** with slicers, advanced visuals, and KPI cards.

The final dashboard allows **dynamic filtering** and **multi-dimensional analysis**, supporting conservation decision-making and field survey optimization.