Data cleaning for Exploratory Data Analysis Eastern Bluebird Capstone

Team: Yichen Le yl4347, Yuheng Shen ys2393, Linyi Xia lx277, Jia Liu jl4769, Rongjian Zhai rz495

Domain problem formulation

We aim to build a robust workflow that predicts Eastern Bluebird occurrences across their range in the eastern United States and southern Canada. The predictors appear to come from gridded environmental products summarizing land cover composition and topography. There is no accompanying metadata, so part of the project involves confirming how these variables were constructed (e.g., spatial resolution, temporal coverage, buffer size). Understanding the provenance of each variable is essential before interpreting model results or making ecological claims.

Step 1: Review background information

Information on data collection

The source file is distributed as `EasternBluebird.csv`, a comma-separated export from an unknown system. No formal documentation accompanied the file. Key open questions:

- Who produced the CSV export and at what stage of processing (raw observations, processed grids, etc.)?
- Do the land cover percentages represent a single year, a multi-year average, or multiple concentric buffers stacked together (totals sometimes exceed 100%)?
- Are repeated latitude/longitude pairs distinct surveys through time or different subsamples of the same remote-sensing grid?

Data dictionary

Column descriptions below are inferred from header names and exploratory analysis. They should be verified against official documentation when it becomes available.

```
tibble::tribble(
  ~column, ~description,
  "LATITUDE", "Latitude of the sampling footprint in decimal degrees (WGS84).",
  "LONGITUDE", "Longitude of the sampling footprint in decimal degrees (WGS84, negative for West).",
  "ELEV", "Elevation of the footprint in meters above sea level (negative values indicate locations bel
  "Shallow_Ocean", "Percent of the footprint classified as shallow ocean water.",
  "CoastShore_lines", "Percent of the footprint flagged as coastal shoreline interface.",
  "Shallow_Inland", "Percent of the footprint covered by shallow inland water bodies.",
  "Deep_Inland", "Percent of the footprint covered by deep inland water bodies.",
  "Moderate_Ocean", "Percent of the footprint in moderate-depth ocean water.",
  "Deep_Ocean", "Percent of the footprint in deep ocean water.",
  "Evergreen_needle", "Percent evergreen needleleaf forest cover.",
  "Grasslands", "Percent grassland cover.",
  "Croplands", "Percent cropland or agricultural cover.",
  "Urban_Built", "Percent urban or built-up land cover.",
  "Barren", "Percent barren land (bare soil/rock).",
```

```
  "Evergreen_broad", "Percent evergreen broadleaf forest cover.",
  "Deciduous_needle", "Percent deciduous needleleaf forest cover.",
  "Deciduous_broad", "Percent deciduous broadleaf forest cover.",
  "Mixed_forest", "Percent mixed forest cover.",
  "Closed_shrubland", "Percent closed shrubland cover.",
  "Open_shrubland", "Percent open shrubland cover.",
  "Woody_savannas", "Percent woody savanna cover.",
  "Savannas", "Percent savanna cover.",
  "y", "Binary indicator of Eastern Bluebird presence (1) or absence (0) for this footprint."
) |>
  knitr::kable()
```

column	description
LATITUDE	Latitude of the sampling footprint in decimal degrees (WGS84).
LONGITUDE	Longitude of the sampling footprint in decimal degrees (WGS84, negative for West).
ELEV	Elevation of the footprint in meters above sea level (negative values indicate locations below sea level).
Shallow_Ocean	Percent of the footprint classified as shallow ocean water.
CoastShore_lines	Percent of the footprint flagged as coastal shoreline interface.
Shallow_Inland	Percent of the footprint covered by shallow inland water bodies.
Deep_Inland	Percent of the footprint covered by deep inland water bodies.
Moderate_Ocean	Percent of the footprint in moderate-depth ocean water.
Deep_Ocean	Percent of the footprint in deep ocean water.
Evergreen_needle	Percent evergreen needleleaf forest cover.
Grasslands	Percent grassland cover.
Croplands	Percent cropland or agricultural cover.
Urban_Built	Percent urban or built-up land cover.
Barren	Percent barren land (bare soil/rock).
Evergreen_broad	Percent evergreen broadleaf forest cover.
Deciduous_needle	Percent deciduous needleleaf forest cover.
Deciduous_broad	Percent deciduous broadleaf forest cover.
Mixed_forest	Percent mixed forest cover.
Closed_shrubland	Percent closed shrubland cover.
Open_shrubland	Percent open shrubland cover.
Woody_savannas	Percent woody savanna cover.
Savannas	Percent savanna cover.
y	Binary indicator of Eastern Bluebird presence (1) or absence (0) for this footprint.

Step 2: Load the data

We ingest the CSV directly with `readr::read_csv` and standardize column names for downstream wrangling.

```
library(tidyverse)
library(janitor)
```

```
data_path <- "EasternBluebird.csv"

file_details <- file.info(data_path)
tibble(
  file_size_mb = round(file_details$size / 1024^2, 2),
  last_modified = file_details$mtime
)
```

```
## # A tibble: 1 x 2
```

```
##   file_size_mb last_modified
##          <dbl> <dttm>
## 1         7.26 2025-10-31 15:42:33
```

```
ingest_eastern_bluebird <- function(path) {
  readr::read_csv(path, show_col_types = FALSE)
}

bluebird_raw <- ingest_eastern_bluebird(data_path)
bluebird_raw |> glimpse()
```

```
## Rows: 64,724
## Columns: 23
## $ LATITUDE         <dbl> 35.27266, 35.95440, 36.72264, 37.02214, 37.29057, 37.~
## $ LONGITUDE        <dbl> -76.61289, -78.94340, -81.48981, -79.46737, -80.45833~
## $ ELEV             <dbl> 2.243650, 100.915229, 939.298682, 212.170286, 773.579~
## $ Shallow_Ocean    <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000,~
## $ CoastShore_lines <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000,~
## $ Shallow_Inland   <dbl> 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000,~
## $ Deep_Inland      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Moderate_Ocean   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Deep_Ocean       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Evergreen_needle <dbl> 40.816327, 0.000000, 0.000000, 2.040816, 0.000000, 0.~
## $ Grasslands       <dbl> 2.040816, 0.000000, 0.000000, 0.000000, 0.000000, 5.5~
## $ Croplands        <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 38.~
## $ Urban_Built      <dbl> 0.000000, 63.888889, 0.000000, 0.000000, 0.000000, 2.~
## $ Barren           <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.0~
## $ Evergreen_broad  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Deciduous_needle <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.0~
## $ Deciduous_broad  <dbl> 0.000000, 0.000000, 100.000000, 10.204082, 100.000000~
## $ Mixed_forest     <dbl> 51.020408, 11.111111, 0.000000, 85.714286, 0.000000, ~
## $ Closed_shrubland <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.0~
## $ Open_shrubland   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Woody_savannas   <dbl> 4.081633, 25.000000, 0.000000, 2.040816, 0.000000, 19~
## $ Savannas         <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0.0~
## $ y                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,~
```

```
landcover_cols <- c(
  "shallow_ocean", "coast_shore_lines", "shallow_inland", "deep_inland",
  "moderate_ocean", "deep_ocean", "evergreen_needle", "grasslands", "croplands",
  "urban_built", "barren", "evergreen_broad", "deciduous_needle",
  "deciduous_broad", "mixed_forest", "closed_shrubland", "open_shrubland",
  "woody_savannas", "savannas"
)

bluebird_ingested <- bluebird_raw |>
  janitor::clean_names() |>
  mutate(
    latitude = as.numeric(latitude),
    longitude = as.numeric(longitude),
    landcover_total = rowSums(across(all_of(landcover_cols))),
    landcover_total_over_100 = landcover_total > 100 + 1e-6
  )

tibble(
```

```
  rows = nrow(bluebird_ingested),
  columns = ncol(bluebird_ingested)
)
```

```
## # A tibble: 1 x 2
##    rows columns
##   <int>   <int>
## 1 64724      25
```

```
bluebird_ingested |>
  slice_head(n = 5)
```

```
## # A tibble: 5 x 25
##   latitude longitude   elev shallow_ocean coast_shore_lines shallow_inland
##      <dbl>     <dbl>  <dbl>         <dbl>              <dbl>          <dbl>
## 1     35.3     -76.6   2.24             0                  0              0
## 2     36.0     -78.9 101.               0                  0              0
## 3     36.7     -81.5 939.               0                  0              0
## 4     37.0     -79.5 212.               0                  0              0
## 5     37.3     -80.5 774.               0                  0              0
## # i 19 more variables: deep_inland <dbl>, moderate_ocean <dbl>,
## #   deep_ocean <dbl>, evergreen_needle <dbl>, grasslands <dbl>,
## #   croplands <dbl>, urban_built <dbl>, barren <dbl>, evergreen_broad <dbl>,
## #   deciduous_needle <dbl>, deciduous_broad <dbl>, mixed_forest <dbl>,
## #   closed_shrubland <dbl>, open_shrubland <dbl>, woody_savannas <dbl>,
## #   savannas <dbl>, y <dbl>, landcover_total <dbl>,
## #   landcover_total_over_100 <lgl>
```

Step 3: Examine the data

We inspect the loaded tibble to confirm that values sit within expected ranges and to flag any issues that may require cleaning.

Invalid values

```
bluebird_ingested |>
  summarise(
    lat_min = min(latitude),
    lat_max = max(latitude),
    lon_min = min(longitude),
    lon_max = max(longitude),
    elev_min = min(elev),
    elev_max = max(elev)
  )
```

```
## # A tibble: 1 x 6
##   lat_min lat_max lon_min lon_max elev_min elev_max
##     <dbl>   <dbl>   <dbl>   <dbl>    <dbl>    <dbl>
## 1    35.0    50.0   -85.0   -70.0    -17.7    1862.
```

Coordinates fall within the eastern United States and southern Canada, and elevations remain plausible for terrestrial sites.

Missing values

```
bluebird_ingested |>
  summarise(across(everything(), ~ sum(is.na(.)))) |>
  tidyr::pivot_longer(everything(), names_to = "column", values_to = "missing_count") |>
  filter(missing_count > 0)
```

```
## # A tibble: 0 x 2
## # i 2 variables: column <chr>, missing_count <int>
```

No missing values are present, so downstream analyses can proceed without imputation.

Data format

```
duplicate_summary <- bluebird_ingested |>
  count(latitude, longitude, name = "n_records") |>
  arrange(desc(n_records))

duplicate_summary |>
  summarise(
    total_sites = n(),
    max_records = max(n_records),
    sites_with_duplicates = sum(n_records > 1),
    pct_sites_with_duplicates = scales::percent(sites_with_duplicates / total_sites)
  )
```

```
## # A tibble: 1 x 4
##   total_sites max_records sites_with_duplicates pct_sites_with_duplicates
##         <int>       <int>                 <int> <chr>
## 1       36434          51                  9296 26%
```

Repeated latitude/longitude footprints suggest the data contain multiple surveys per site. Any train/test split should keep replicated footprints together.

Column names

```
tibble(column = names(bluebird_ingested))
```

```
## # A tibble: 25 x 1
##    column
##    <chr>
##  1 latitude
##  2 longitude
##  3 elev
##  4 shallow_ocean
##  5 coast_shore_lines
##  6 shallow_inland
##  7 deep_inland
##  8 moderate_ocean
##  9 deep_ocean
## 10 evergreen_needle
## # i 15 more rows
```

`janitor::clean_names()` already produces snake_case headers that are consistent across the dataset.

Variable type

```
tibble(
  column = names(bluebird_ingested),
  class = purrr::map_chr(bluebird_ingested, ~ paste(class(.x), collapse = ", "))
)
```

```
## # A tibble: 25 x 2
##    column            class
##    <chr>             <chr>
##  1 latitude          numeric
##  2 longitude         numeric
##  3 elev              numeric
##  4 shallow_ocean     numeric
##  5 coast_shore_lines numeric
##  6 shallow_inland    numeric
##  7 deep_inland       numeric
##  8 moderate_ocean    numeric
##  9 deep_ocean        numeric
## 10 evergreen_needle  numeric
## # i 15 more rows
```

Predictors are numeric, and the response `y` remains coded as 0/1. If factor semantics are required, the conversion can occur closer to modeling.

Data specific explorations

```
bluebird_ingested |>
  summarise(
    min_total = min(landcover_total),
    p10_total = quantile(landcover_total, 0.10),
    median_total = median(landcover_total),
    mean_total = mean(landcover_total),
    p90_total = quantile(landcover_total, 0.90),
    max_total = max(landcover_total),
    pct_over_100 = scales::percent(mean(landcover_total_over_100))
  )
```

```
## # A tibble: 1 x 7
##   min_total p10_total median_total mean_total p90_total max_total pct_over_100
##       <dbl>     <dbl>        <dbl>      <dbl>     <dbl>     <dbl> <chr>
## 1      54.8      100.         100.       106.      131.      200. 19%
```

Land cover totals cluster around 100 but occasionally exceed it, reinforcing the need to confirm whether overlapping buffers or stacked categories generated the export.

Step 4: Clean the data

The cleaning workflow below:

1. Ingests the CSV and standardizes column names.
2. Computes land cover totals and flags rows where totals exceed 100.
3. Adds normalized land cover fractions so that each record sums to one regardless of the original scale.
4. Aggregates to a site-level table (one row per latitude/longitude) with counts of replicate observations and the mean/any presence outcomes. This reduces leakage when splitting data by location.

```r
clean_eastern_bluebird <- function(path) {
  raw <- ingest_eastern_bluebird(path)
  cleaned <- raw |>
    janitor::clean_names() |>
    mutate(
      latitude = as.numeric(latitude),
      longitude = as.numeric(longitude),
      landcover_total = rowSums(across(all_of(landcover_cols))),
      landcover_total_over_100 = landcover_total > 100 + 1e-6
    ) |>
    mutate(
      across(
        all_of(landcover_cols),
        ~ if_else(landcover_total == 0, 0, .x / landcover_total),
        .names = "{.col}_frac"
      )
    )

  site_level <- cleaned |>
    group_by(latitude, longitude) |>
    summarise(
      n_observations = n(),
      presence_any = as.integer(any(y == 1)),
      presence_rate = mean(y),
      elev_mean = mean(elev),
      across(all_of(landcover_cols), mean, .names = "{.col}_mean"),
      across(ends_with("_frac"), mean),
      landcover_total_mean = mean(landcover_total),
      .groups = "drop"
    )

  list(
    raw = raw,
    cleaned = cleaned,
    site_level = site_level
  )
}

bluebird_outputs <- clean_eastern_bluebird(data_path)

bluebird_outputs$site_level |>
  summarise(
    sites = n(),
    mean_records_per_site = mean(n_observations),
    max_records_per_site = max(n_observations)
  )
```

```
## # A tibble: 1 x 3
##   sites mean_records_per_site max_records_per_site
##   <int>                 <dbl>                <int>
## 1 36434                  1.78                   51
```

```r
bluebird_outputs$cleaned |>
  select(
```

```
    latitude, longitude, elev, landcover_total,
    landcover_total_over_100, woody_savannas, woody_savannas_frac, y
  ) |>
  slice_head(n = 5)
```

```
## # A tibble: 5 x 8
##   latitude longitude   elev landcover_total landcover_total_over_100
##      <dbl>     <dbl>  <dbl>           <dbl> <lgl>
## 1     35.3     -76.6   2.24            98.0 FALSE
## 2     36.0     -78.9 101.             100   FALSE
## 3     36.7     -81.5 939.             100   FALSE
## 4     37.0     -79.5 212.             100.  FALSE
## 5     37.3     -80.5 774.             100   FALSE
## # i 3 more variables: woody_savannas <dbl>, woody_savannas_frac <dbl>, y <dbl>
```

```
bluebird_outputs$site_level |>
  slice_head(n = 5)
```

```
## # A tibble: 5 x 45
##   latitude longitude n_observations presence_any presence_rate elev_mean
##      <dbl>     <dbl>          <int>        <int>         <dbl>     <dbl>
## 1     35.0     -80.6              1            0             0     204.
## 2     35.0     -79.1              1            0             0      69.2
## 3     35.0     -83.2              1            0             0     794.
## 4     35.0     -80.6              1            0             0     198.
## 5     35.0     -83.3              1            0             0    1077.
## # i 39 more variables: shallow_ocean_mean <dbl>, coast_shore_lines_mean <dbl>,
## #   shallow_inland_mean <dbl>, deep_inland_mean <dbl>,
## #   moderate_ocean_mean <dbl>, deep_ocean_mean <dbl>,
## #   evergreen_needle_mean <dbl>, grasslands_mean <dbl>, croplands_mean <dbl>,
## #   urban_built_mean <dbl>, barren_mean <dbl>, evergreen_broad_mean <dbl>,
## #   deciduous_needle_mean <dbl>, deciduous_broad_mean <dbl>,
## #   mixed_forest_mean <dbl>, closed_shrubland_mean <dbl>, ...
```

```
bluebird_outputs$site_level |>
  summarise(
    min_rate = min(presence_rate),
    mean_rate = mean(presence_rate),
    median_rate = median(presence_rate),
    max_rate = max(presence_rate)
  )
```

```
## # A tibble: 1 x 4
##   min_rate mean_rate median_rate max_rate
##      <dbl>     <dbl>       <dbl>    <dbl>
## 1        0    0.0562           0        1
```

Next steps:

- Confirm the interpretation of land cover totals that exceed 100 and determine whether they require re-normalization or stratified handling.
- Decide on a final export format (CSV, parquet, or RDS) for both the observation-level and site-level tables once metadata questions are resolved.
- Incorporate temporal information if it is available elsewhere so we can respect survey-years during modeling splits.