Bayesian Stat Project 🦜🪩

Tim Zhou 30407324

Github Link

TL;DR

We applied Bayesian inference to estimate and forecast the density of migratory birds in British Columbia. Using hierarchical spatial and temporal models, we investigated long-term population trends from over 50 years of bird survey data.

Dataset

Birds count dataset is accessible through Breeding Bird Survey [USGS BBS](https://www.usgs.gov/centers/eesc) by querying by region for BC. It provides annual bird counts across survey routes in British Columbia (and all of North America), spanning more than five decades. Each year, different number of routes were recorded in the BC region, we therefore analyze observation per route instead of total observation to fight off this bias.

Here's a preview of the raw Dataset birds.csv (full dataset available on GitHub)

Problem Setup & Background

Global animal populations have declined by an average of 68% since 1970 (Grooten et al., 2020), driven by climate change, habitat loss, and human activity. In this project, we focus on migratory bird populations in British Columbia (BC), applying Bayesian modeling to understand how species have responded over time.

We use hierarchical Bayesian time series models to track the trajectories of the most frequently observed species, while modeling less common species as a pooled group. This structure allows us to detect species-specific trends and capture shared dynamics in less abundant populations, and show clearly that population across all major BC bird species have been steadily declining.

By focusing on BC observations, we ground our analysis in a local context with ecological and personal relevance. Our results highlight population shifts and can serve as yet another alarming reminder of biodiversity loss, and hopefully support the need for conservation research, rehabilitation projects and investments, and regulatory policies.

Related Work (Literature Review)

The USGS BBS bird dataset has been used extensively within the North America ecologists community. Many in the past have applied the domain-specific route-based model on the dataset and had reasonable success.

The closest analysis to ours is the "Analysis of the North American Breeding Bird Survey Using Hierarchical Models" by Sauer and Link (2011). The authors applied a Bayesian hierarchical Poisson regression model to estimate long-term population trends from Breeding Bird Survey (BBS) data across North America. This model structure allowed

them to account for variability at multiple levels—such as species, route, year, and region—and to quantify uncertainty in trend estimates more effectively than traditional methods. One key finding was that Bayesian estimates showed substantial differences from classical route-regression results for about 15 species, highlighting the limitations of older estimation techniques.

Our project takes inspiration from theirs by continuing down the Bayesian path, but differs in several important aspects. First, we explicitly address the long-tailed nature of ecological count data: the majority of observations are dominated by a small number of common species, which we model separately from the long tail of rarer species. This two-part modeling strategy simplifies the problem, which allows us to focus on the specific region of BC which only accounts for a very small subset of the original data, and greatly improves computational efficiency. Second, instead of conducting a continent-wide analysis, we restrict our study to the British Columbia (BC) region. This allows us to capture finer-scale spatial trends and better align with local conservation priorities. Additionally, our work explores spatiotemporal modeling extensions and evaluates predictive accuracy using modern Bayesian tools.

Model Definition and Fitting

We define separate slope and intercept for each of the top 10 most frequent species in BC. Intercepts follow an exponential distribution with low rate to make sure domain lines up (positive) and uninformative (since we're not domain experts). Slope follow normal distributions with centers at zero to make sure we're not imposing a declining population bias in our priors. Standard deviations are given exponential priors since they also have positive domains.

Mathematically:

- $s = 1, \ldots, S$ index the top species
- $t = 1, \ldots, T$ index years
- $y_{\text{top},s,t}$ be the observation for top species s at year t
- $y_{\text{other},t}$ be the observation for the "other species" group at year t
- year_t be the year since first recorded observation year

Then our model is

$$
\begin{aligned}
\text{intercept}_s &\sim \text{Exponential}(0.001) && \text{for } s = 1, \ldots, S+1 \\
\text{slope}_s &\sim \mathcal{N}(0, \sigma_{\text{top}}) && \text{for } s = 1, \ldots, S \\
\text{slope}_{S+1} &\sim \mathcal{N}(0, \sigma_{\text{noise}}) \\
\sigma_{\text{top}} &\sim \text{Exponential}(0.5) \\
\sigma_{\text{noise}} &\sim \text{Exponential}(0.5) \\
\text{obs\_noise\_top} &\sim \text{Exponential}(0.1) \\
\text{obs\_noise\_other} &\sim \text{Exponential}(0.1)
\end{aligned}
$$

$$
\begin{aligned}
y_{\text{top},s,t} &\sim \mathcal{N}(\text{intercept}_s + \text{slope}_s \cdot \text{year}_t,\ \text{obs\_noise\_top}) \\
y_{\text{other},t} &\sim \mathcal{N}(\text{intercept}_{S+1} + \text{slope}_{S+1} \cdot \text{year}_t,\ \text{obs\_noise\_other})
\end{aligned}
$$

(fig 1. Mathematical definition of our models. Overleaf.)

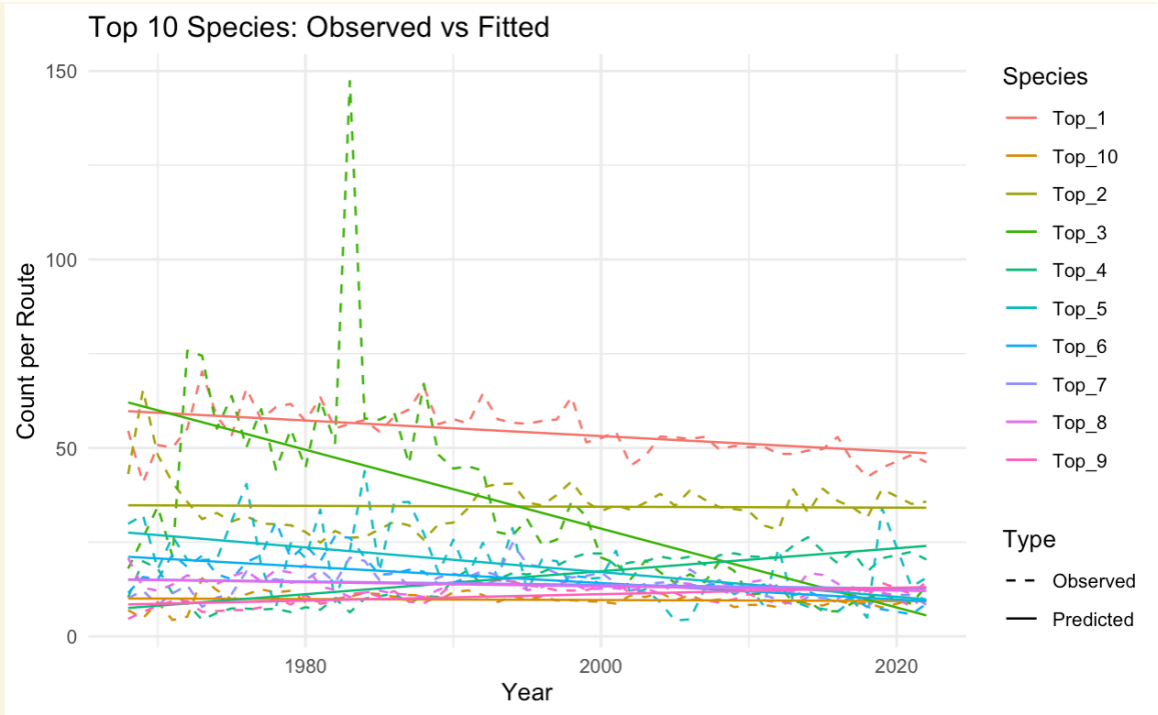
We ran MCMC with 5000 iterations for 4 chains. Here's the results on the slopes variables for each of the ten frequent species and the "others", which are the parameters we care about the most.

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
slope[1]	-0.21	0.00	0.06	-0.33	-0.25	-0.21	-0.16	-0.08	700	1.00
slope[2]	-0.01	0.00	0.06	-0.14	-0.06	-0.01	0.03	0.11	780	1.01
slope[3]	-1.05	0.00	0.07	-1.18	-1.09	-1.05	-1.00	-0.92	612	1.01
slope[4]	0.31	0.00	0.07	0.18	0.26	0.30	0.35	0.43	662	1.01
slope[5]	-0.33	0.00	0.06	-0.45	-0.37	-0.33	-0.28	-0.20	722	1.00
slope[6]	-0.22	0.00	0.06	-0.35	-0.26	-0.22	-0.18	-0.09	699	1.00
slope[7]	-0.05	0.00	0.06	-0.17	-0.09	-0.05	0.00	0.08	678	1.00
slope[8]	-0.06	0.00	0.06	-0.18	-0.10	-0.05	-0.01	0.07	784	1.00
slope[9]	0.09	0.00	0.06	-0.04	0.04	0.09	0.13	0.21	672	1.01
slope[10]	-0.01	0.00	0.06	-0.14	-0.06	-0.01	0.03	0.12	510	1.01
slope[11]	-0.69	0.01	0.28	-1.22	-0.88	-0.69	-0.50	-0.12	565	1.01

(fig 2. slope estimates and confidence intervals)

As we can see, all but one of the species are showing a decline in population when we consider the mean of slopes, with 5 out of 11 even having the 95% confidence interval being entirely below 0, showing a strong evidence of birds population declining with years.

Here we show the fitted results for the top 10 species, we present fitted values for the other species and combined total observations in the appendix.



(fig 3. fitted values of observation per route for the top 10 species, which were American Robin, Swainson Thrush, European Starling, Warbling Vireo, Pine Siskin, American Crow, Dark-eyed junco, Chipping Sparrow, Yellow romped warbler, and song sparrows)

Prediction for Year 2020

One interesting aspect of the birds dataset was that the year 2020 was missing because of covid. We therefore make a prediction to fill in the missing value with our fitted mode. (see code in appendix), we predict that if we were to be collecting data in 2020 we would've seen top 10 =182; other = 343; total = 525.

Project Theme (Time-Series)

In this project, I explored time-series and hierarchical Bayesian models to analyze bird population trends in British Columbia. By carefully designing priors and using linear regression structures, we were able to model the average count per route for both individual species and the overall population.

The time-series component allowed us to estimate posterior distributions over temporal trends (slopes). For many of the most commonly observed species, the entire posterior distribution of the slope lay below zero, strongly indicating declining populations. Additionally, by pooling less frequent species into a single "other species" group, we found similar declines—suggesting that biodiversity loss is not limited to common species alone, but also affects rarer ones. This presents a critical challenge for conservation efforts. The time-series aspect also allowed us to make "predictions" on the 2020 count, which was missing.

Limitation

One key limitation of this approach is the somewhat arbitrary choice of the "top k" species. Larger values of k increase model expressiveness, but also lead to more complex posterior distributions that are harder for MCMC to sample efficiently. In practice, the choice of k should balance model fidelity and computational feasibility, and will depend heavily on the size of the dataset (i.e., number of years available).

Another limitation is the lack of spatial generalizability. The model was tailored specifically for BC routes. If we wanted to extend this analysis across regions (e.g., Canada, the US, and Mexico), we would need to account for region-specific species compositions. The "top k" species in one area may be completely different from those in another, making pooled models or direct comparisons nontrivial.

Summary

Using Bayesian time-series analysis, we have demonstrated clear signs of population decline among migratory bird species in British Columbia—both common and uncommon. Our model also provided predictive insight for missing years, including 2020, when field surveys were paused due to the COVID-19 pandemic.

References:

1. (Grooten et al., 2020) Catastrophic 73% decline in the average size of global wildlife populations in just 50 years reveals a ‘system in peril’, https://www.worldwildlife.org/press-releases/catastrophic-73-decline-in-the-average-size-of-global-wildlife-populations-in-just-50-years-reveals-a-system-in-peril

2. (Sauer and Link 2011) Analysis of the North American Breeding Bird Survey Using Hierarchical Models, https://academic.oup.com/auk/article/128/1/87/5149447

447 Project Appendix (Code)

Code

Github

Repo (https://github.com/TimothyZG/447project)

Data Wrangling:

Hide

```
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)

n = 10

bird <- read_csv("bird.csv", show_col_types = FALSE)
head(bird)
```

Species List <chr>	**1968** <dbl>	**1969** <dbl>	**1970** <dbl>	**1971** <dbl>	**1972** <dbl>	**1973** <dbl>	**1974** <dbl>	**1975** <dbl>	**1976** <dbl>
Route Count	15	13	17	12	9	36	52	50	42
Cackling Goose	0	0	0	0	0	0	0	0	0
Canada Goose	12	0	0	2	17	31	58	53	71
Mute Swan	0	0	0	0	0	0	0	0	0
Trumpeter Swan	0	0	1	0	0	0	0	0	1
Tundra Swan	0	0	0	0	0	0	0	0	0

6 rows | 1-10 of 56 columns

Hide

```
bird$`2020` <- as.numeric(gsub("-", "0", bird$`2020`))
bird <- bird %>% select(-`2020`) # remove 2020 since everything is null this year

route_row <- bird %>%
  filter(`Species List` == "Route Count") %>%
  select(-`Species List`) %>%
  pivot_longer(everything(), names_to = "Year", values_to = "RouteCount") %>%
  mutate(Year = as.integer(Year))

bird_data <- bird %>%
  filter(`Species List` != "Route Count") %>%
  filter(`Species List` != "Total individuals")%>%
  filter(`Species List` != "Total Species")

years <- sort(unique(bird_long$Year))
total_species_count = length(unique(bird_data$`Species List`))
cat("\nTotal recorded number of species in the BC region is:",total_species_count)
```

```

Total recorded number of species in the BC region is: 320
```

Hide

```
T <- length(years)
S <- n
n_other_species <- length(unique(bird_data$`Species List`)) - S

bird_long <- bird_data %>%
  pivot_longer(-`Species List`, names_to = "Year", values_to = "Count") %>%
  mutate(
    Year = as.integer(Year),
    Count = as.numeric(gsub("-", "0", Count))
  ) %>%
  left_join(route_row, by = "Year") %>%
  mutate(CountPerRoute = Count / RouteCount)

top_species <- bird_long %>%
  group_by(`Species List`) %>%
  summarise(total = sum(Count, na.rm = TRUE)) %>%
  slice_max(total, n = n) %>%
  pull(`Species List`)
cat("Top 10 species in the BC region are:",paste(top_species, collapse = ", "))
```

```
Top 10 species in the BC region are: American Robin, Swainson's Thrush, European Starlin
g, Warbling Vireo, Pine Siskin, American Crow, (Oregon Junco) Dark-eyed Junco, Chipping
Sparrow, (Audubon's Warbler) Yellow-rumped Warbler, Song Sparrow
```

Hide

```
y_top <- bird_long %>%
  filter(`Species List` %in% top_species) %>%
  group_by(`Species List`, Year) %>%
  summarise(
    Total = sum(Count, na.rm = TRUE),
    RouteCount = first(RouteCount),
    .groups = "drop"
  ) %>%
  mutate(PerRoute = Total / RouteCount) %>%
  mutate(`Species List` = factor(`Species List`, levels = top_species)) %>%
  pivot_wider(
    id_cols = Year,
    names_from = `Species List`,
    values_from = PerRoute,
    values_fill = 0
  ) %>%
  arrange(Year) %>%
  select(all_of(top_species)) %>%
  t()

# y_other
y_other <- bird_long %>%
  filter(!`Species List` %in% top_species) %>%
  group_by(Year) %>%
  summarise(
    Total = sum(Count, na.rm = TRUE),
    RouteCount = first(RouteCount),
    .groups = "drop"
  ) %>%
  mutate(y_other = Total / RouteCount) %>%
  arrange(Year) %>%
  pull(y_other)

# y_total
y_total <- bird_long %>%
  group_by(Year) %>%
  summarise(
    Total = sum(Count, na.rm = TRUE),
    RouteCount = first(RouteCount),
    .groups = "drop"
  ) %>%
  mutate(y_total = Total / RouteCount) %>%
  arrange(Year) %>%
  pull(y_total)


y_top[is.na(y_top)] <- 0
y_other[is.na(y_other)] <- 0
y_total[is.na(y_total)] <- 0

RouteCount <- route_row %>%
  arrange(Year) %>%
```

```
  pull(RouteCount)

year_zeroed <- years - min(years)

stan_data <- list(
  T = T,
  S = S,
  y_top = y_top,
  y_other = y_other,
  y_total = y_total,
  n_other_species = n_other_species,
  year_zeroed = year_zeroed,
  RouteCount = RouteCount
)
```

Bird Density analysis Model

Hide

```
data {
  int<lower=1> T;
  int<lower=1> S;
  matrix[S, T] y_top;
  vector[T] y_other;
  int<lower=1> n_other_species;
  vector[T] year_zeroed;
}

parameters {
  vector<lower=0>[S + 1] intercept;
  vector[S + 1] slope;
  real<lower=0> sigma_noise;
  real<lower=0> sigma_top;
  real<lower=0> sigma_other;
  real<lower=0> obs_noise_top;
  real<lower=0> obs_noise_other;
}

model {
  intercept ~ exponential(0.005);
  sigma_top ~ exponential(0.5);
  sigma_noise ~ exponential(0.5);
  for (s in 1:S)
    slope[s] ~ normal(0, sigma_top);
  slope[S + 1] ~ normal(0, sigma_noise);

  obs_noise_top ~ exponential(0.1);
  obs_noise_other ~ exponential(0.1);

  for (t in 1:T) {
    for (s in 1:S)
      y_top[s, t] ~ normal(intercept[s] + slope[s] * year_zeroed[t], obs_noise_top);
    y_other[t] ~ normal(intercept[S + 1] + slope[S + 1] * year_zeroed[t], obs_noise_othe
r);
  }
}

generated quantities {
  vector[T] y_total_pred;
  for (t in 1:T) {
    real top_sum = 0;
    for (s in 1:S)
      top_sum += intercept[s] + slope[s] * year_zeroed[t];

    y_total_pred[t] = (top_sum + (intercept[S + 1] + slope[S + 1] * year_zeroed[t]));
  }
}
```

Hide

```
library(rstan)
fit <- sampling(
  object = distrest,
  data = stan_data,
  iter = 5000,
  chains = 4,
  seed = 17,
  refresh=-1
)
```

```
Chain 1:
Chain 1: Gradient evaluation took 0.00012 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.2 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1:
Chain 1:  Elapsed Time: 2.806 seconds (Warm-up)
Chain 1:                2.957 seconds (Sampling)
Chain 1:                5.763 seconds (Total)
Chain 1:
Chain 2:
Chain 2: Gradient evaluation took 5.8e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.58 second
s.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2:
Chain 2:  Elapsed Time: 2.696 seconds (Warm-up)
Chain 2:                2.096 seconds (Sampling)
Chain 2:                4.792 seconds (Total)
Chain 2:
Chain 3:
Chain 3: Gradient evaluation took 5.4e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.54 second
s.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3:
Chain 3:  Elapsed Time: 2.675 seconds (Warm-up)
Chain 3:                2.3 seconds (Sampling)
Chain 3:                4.975 seconds (Total)
Chain 3:
Chain 4:
Chain 4: Gradient evaluation took 5.5e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.55 second
s.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4:
Chain 4:  Elapsed Time: 2.589 seconds (Warm-up)
Chain 4:                1.809 seconds (Sampling)
Chain 4:                4.398 seconds (Total)
Chain 4:
```

```
Warning: There were 9730 divergent transitions after warmup. See
https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
to find out why this is a problem and how to eliminate them.
Warning: Examine the pairs() plot to diagnose sampling problems
```

Hide

```
print(fit, pars = c("slope", "intercept"))
```

```
Inference for Stan model: anon_model.
4 chains, each with iter=5000; warmup=2500; thin=1;
post-warmup draws per chain=2500, total post-warmup draws=10000.

               mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
slope[1]      -0.21    0.00 0.06  -0.33  -0.25  -0.21  -0.16  -0.08   700 1.00
slope[2]      -0.01    0.00 0.06  -0.14  -0.06  -0.01   0.03   0.11   780 1.01
slope[3]      -1.05    0.00 0.07  -1.18  -1.09  -1.05  -1.00  -0.92   612 1.01
slope[4]       0.31    0.00 0.07   0.18   0.26   0.30   0.35   0.43   662 1.01
slope[5]      -0.33    0.00 0.06  -0.45  -0.37  -0.33  -0.28  -0.20   722 1.00
slope[6]      -0.22    0.00 0.06  -0.35  -0.26  -0.22  -0.18  -0.09   699 1.00
slope[7]      -0.05    0.00 0.06  -0.17  -0.09  -0.05   0.00   0.08   678 1.00
slope[8]      -0.06    0.00 0.06  -0.18  -0.10  -0.05  -0.01   0.07   784 1.00
slope[9]       0.09    0.00 0.06  -0.04   0.04   0.09   0.13   0.21   672 1.01
slope[10]     -0.01    0.00 0.06  -0.14  -0.06  -0.01   0.03   0.12   510 1.01
slope[11]     -0.69    0.01 0.28  -1.22  -0.88  -0.69  -0.50  -0.12   565 1.01
intercept[1]  59.74    0.08 1.98  55.90  58.43  59.71  61.05  63.64   655 1.00
intercept[2]  34.78    0.07 1.96  31.01  33.41  34.77  36.13  38.61   788 1.00
intercept[3]  62.06    0.08 2.05  57.99  60.71  62.09  63.39  66.07   596 1.01
intercept[4]   7.53    0.08 2.05   3.39   6.15   7.56   8.92  11.61   620 1.01
intercept[5]  27.53    0.07 1.97  23.46  26.22  27.53  28.88  31.33   717 1.00
intercept[6]  21.11    0.08 1.99  17.18  19.82  21.09  22.41  25.16   661 1.00
intercept[7]  15.11    0.08 1.99  11.19  13.79  15.15  16.46  18.97   671 1.00
intercept[8]  15.05    0.07 1.95  11.08  13.74  15.04  16.40  18.78   784 1.00
intercept[9]   8.47    0.08 1.98   4.59   7.13   8.44   9.82  12.37   648 1.00
intercept[10] 10.05    0.09 2.02   6.08   8.75  10.06  11.37  13.96   474 1.01
intercept[11] 378.85   0.36 8.64 361.47 373.07 378.99 384.86 395.27   575 1.01

Samples were drawn using NUTS(diag_e) at Sat Apr 19 23:47:33 2025.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

Predict for 2020

Hide

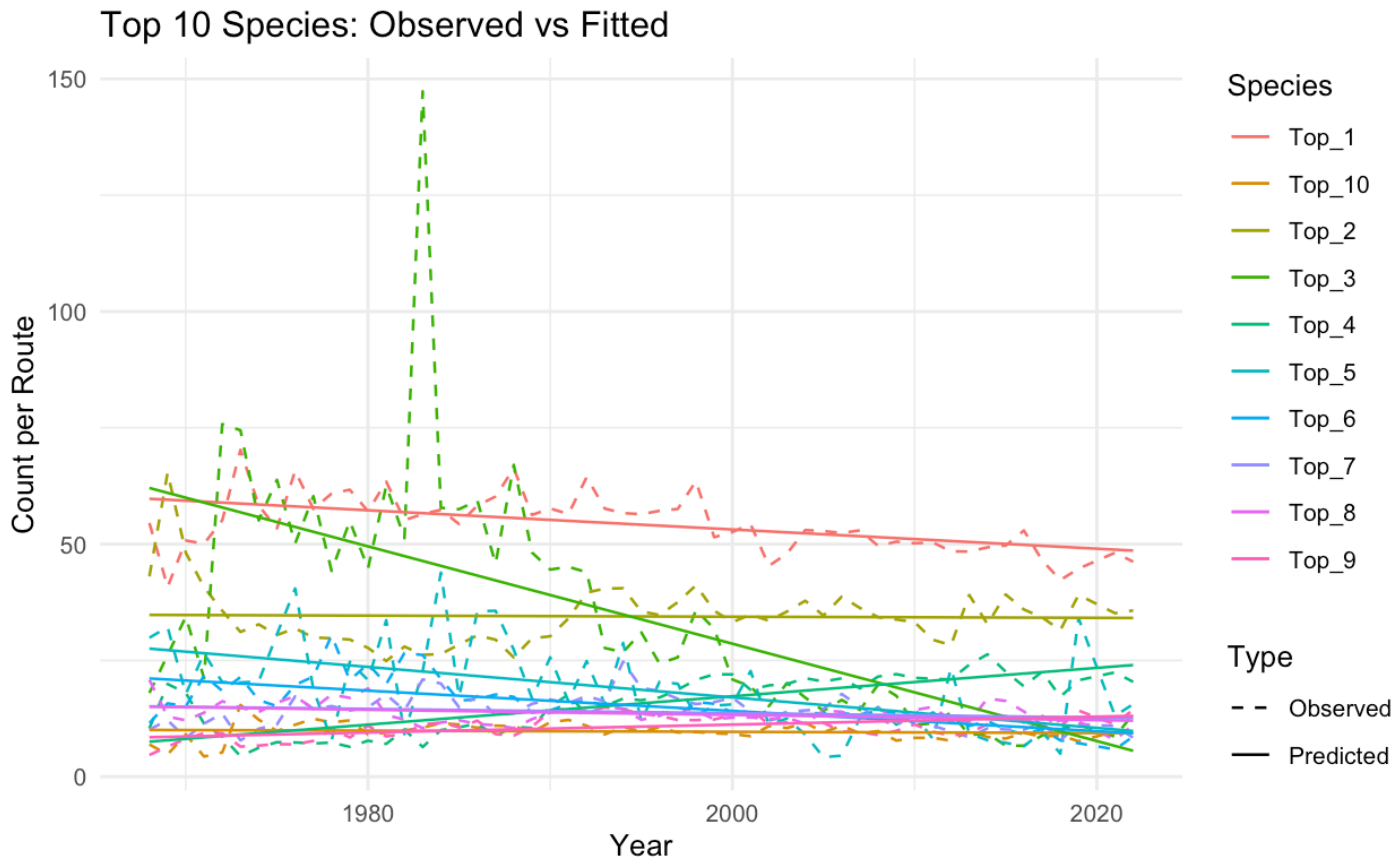
```
posterior <- rstan::extract(fit)
intercepts <- colMeans(posterior$intercept)
slopes     <- colMeans(posterior$slope)
year_base <- min(bird_long$Year)
year_zeroed_2020 <- 2020 - year_base
pred_top_2020 <- intercepts[1:stan_data$S] + slopes[1:stan_data$S] * year_zeroed_2020
pred_other_2020 <- intercepts[stan_data$S + 1] + slopes[stan_data$S + 1] * year_zeroed_2
020
pred_total_2020 <- (sum(pred_top_2020) + pred_other_2020)
cat("pred_top_2020",sum(pred_top_2020),"pred_other_2020",pred_other_2020,"pred_total_202
0",pred_total_2020)
```

```
pred_top_2020 181.6746 pred_other_2020 343.0975 pred_total_2020 524.7721
```

Plotting Predictions

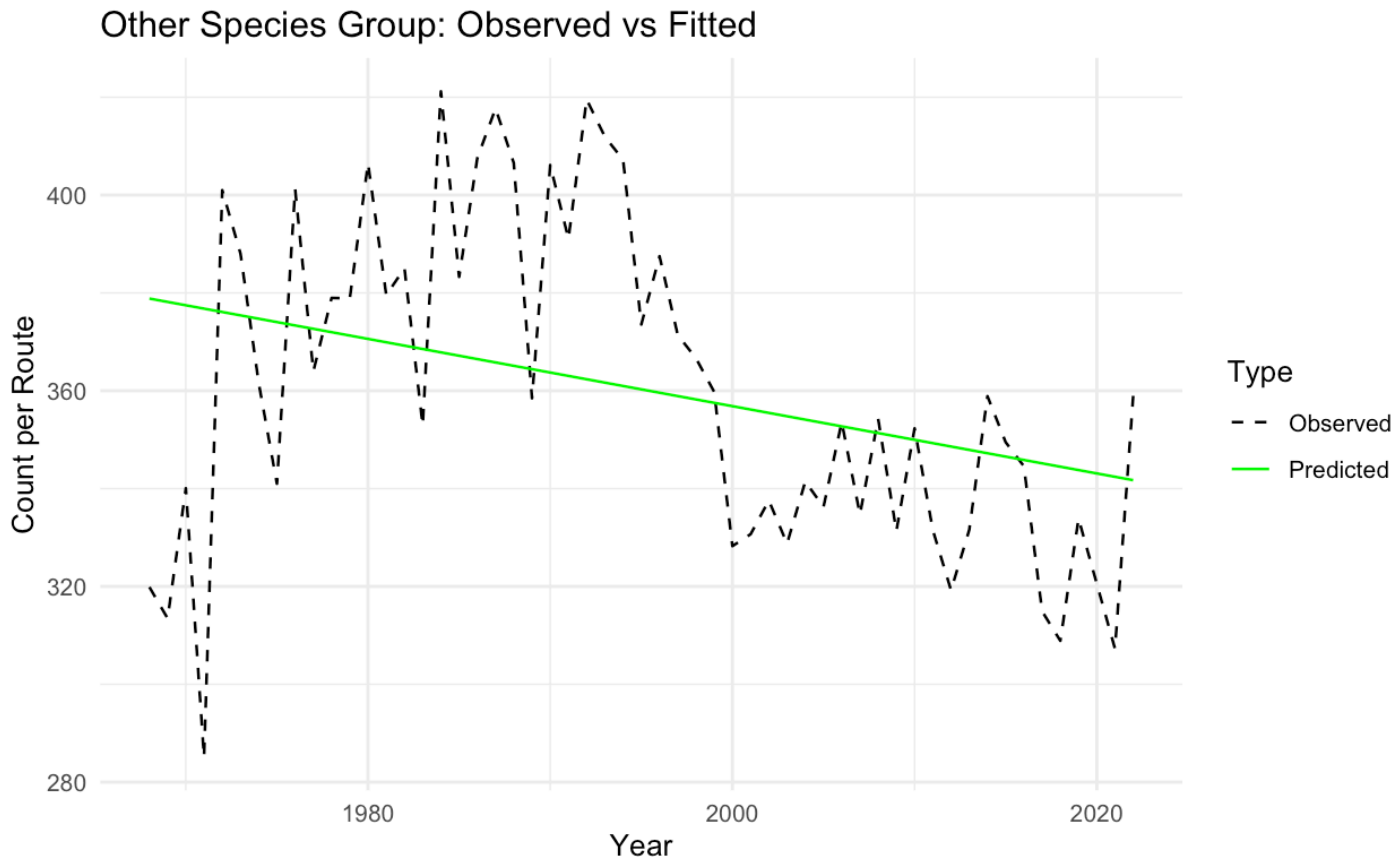
Hide

```
posterior <- rstan::extract(fit)

intercept_mean <- colMeans(posterior$intercept)
slope_mean     <- colMeans(posterior$slope)

y_total_pred <- colMeans(posterior$y_total_pred)

S <- stan_data$S
T <- stan_data$T
years <- stan_data$year_zeroed


fitted_top <- sapply(1:T, function(t) {
  intercept_mean[1:S] + slope_mean[1:S] * years[t]
})
rownames(fitted_top) <- paste0("Top_", 1:S)


fitted_other <- intercept_mean[S + 1] + slope_mean[S + 1] * years

obs_top <- as.data.frame(t(stan_data$y_top))
colnames(obs_top) <- rownames(fitted_top)
obs_top$Year <- years + min(bird_long$Year)

obs_long <- pivot_longer(obs_top, -Year, names_to = "Species", values_to = "Observed")

fit_top <- as.data.frame(t(fitted_top))
colnames(fit_top) <- rownames(fitted_top)
fit_top$Year <- years + min(bird_long$Year)
fit_long <- pivot_longer(fit_top, -Year, names_to = "Species", values_to = "Predicted")

merged_top <- left_join(obs_long, fit_long, by = c("Year", "Species"))

ggplot(merged_top, aes(x = Year, color = Species)) +
  geom_line(aes(y = Observed, linetype = "Observed")) +
  geom_line(aes(y = Predicted, linetype = "Predicted")) +
  scale_linetype_manual(values = c("Observed" = "dashed", "Predicted" = "solid")) +
  labs(
    title = "Top 10 Species: Observed vs Fitted",
    y = "Count per Route",
    linetype = "Type"
  ) +
  theme_minimal()
```



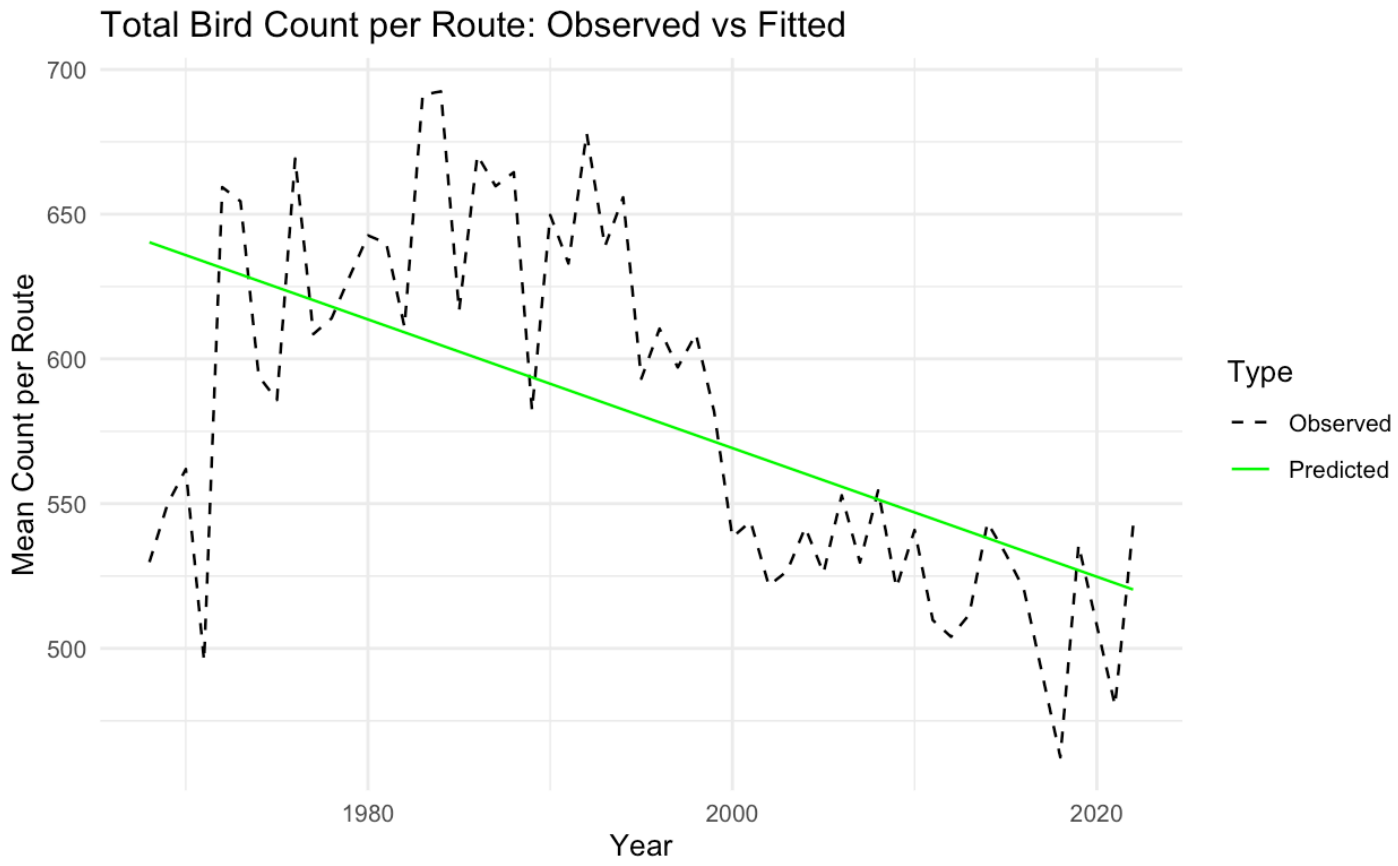
Hide

```
df_other <- data.frame(
  Year = years + min(bird_long$Year),
  Observed = stan_data$y_other,
  Predicted = fitted_other
)

ggplot(df_other, aes(x = Year)) +
  geom_line(aes(y = Observed, linetype = "Observed"), color = "black") +
  geom_line(aes(y = Predicted, linetype = "Predicted"), color = "green") +
  scale_linetype_manual(values = c("Observed" = "dashed", "Predicted" = "solid")) +
  labs(
    title = "Other Species Group: Observed vs Fitted",
    y = "Count per Route",
    linetype = "Type"
  ) +
  theme_minimal()
```



Hide

```
df_total <- data.frame(
  Year = years + min(bird_long$Year),
  Observed = stan_data$y_total,
  Predicted = y_total_pred
)

ggplot(df_total, aes(x = Year)) +
  geom_line(aes(y = Observed, linetype = "Observed"), color = "black") +
  geom_line(aes(y = Predicted, linetype = "Predicted"), color = "green") +
  scale_linetype_manual(values = c("Observed" = "dashed", "Predicted" = "solid")) +
  labs(
    title = "Total Bird Count per Route: Observed vs Fitted",
    y = "Mean Count per Route",
    linetype = "Type"
  ) +
  theme_minimal()
```



Team Contribution Assignment

N/A, single person group.