

610 Data Analysis Report - Sparrow

Shangrong Chi

Introduction

This study investigates how variables such as female population density, nest location, and year of study influence song sparrows' offspring. The dataset comprises observations of data with researchers monitoring every 2-7 days during the mating season and tagged in 15 years.

The analysis focused on several key questions: How does female population density affect reproductive success? Do spatial factors (nest location) influence offspring outcomes? Furthermore, what role do other factors play in reproductive performance? Therefore, using statistics analysis and hierarchical models, breeding success is strongly influenced by population density and nest location, with fewer offspring produced when female population density is high and in eastern nesting locations. While annual environmental conditions substantially affect the outcomes, there is significant zero-inflation in reproduction patterns, suggesting that breeding failure occurs more frequently than the model predicts and that additional factors beyond density and location likely influence reproductive success.

Study Design

- Sample Size:

- Total observations: 742
- Observed over 19 years
- 356 unique birds identified by band numbers
- Each bird observed between 1-7 times (median: 2 observations per bird)

- Time Structure:

- Data collected in 15 of these 19 years
- Regular nest monitoring every 2-7 days during mating season
- Cohorts range from year 2 to 19 (101 missing values)

- Spatial Structure:

- Nest locations recorded using x-y coordinates
- X-coordinates range: 0.25-32.83
- Y-coordinates range: 1.375-5.250
- Only 5 missing spatial coordinates

- Other Measures:

- Female population size (fpop) ranges from 4 to 72 birds
- Age structure from 1-7 years old (101 missing values)
- Reproductive output (spf) as response variable ranges from 0-12 offspring

- Macro Level Variables:

- Year

- Population size(fpop)

- Micro Level Variables:

- Individual bird characteristics (age, cohort, and band)

- Nest location effects (x and y)

- Key Features:

- Complete and consistent sparrow monitoring

- Multi-generational tracking

- Precise nests locations

- Cohort, age, x, y and band columns have missing values (NA)

- Age, cohort and year are related data

- Reproductive success measured by number of fledglings (spf)

- Deal with missing values:

The initial dataset: cohort and age each had 13.6% missing values, while x, y and band ID had minimal missing data (less than 1%). Given the low proportion of missing values, we first removed missing x, y, or band data, reducing the sample size from 742 to 736. For the remaining missing data in cohort and age (100 observations each), we attempted several approaches including Generalized Linear Mixed Models (GLMMs) with multiple imputations (5 imputations) and maximum likelihood estimation (MLE). While the Negative Binomial GLMM showed reasonable overdispersion (ratio = 0.961), diagnostic checks revealed potential issues. Given that cohort, year, and age are mathematically related, and considering the challenges in imputing these values, we just retain only the year variable and drop cohort and age columns. Thus, these steps allowed us to maintain data integrity and properly deal with missing values.

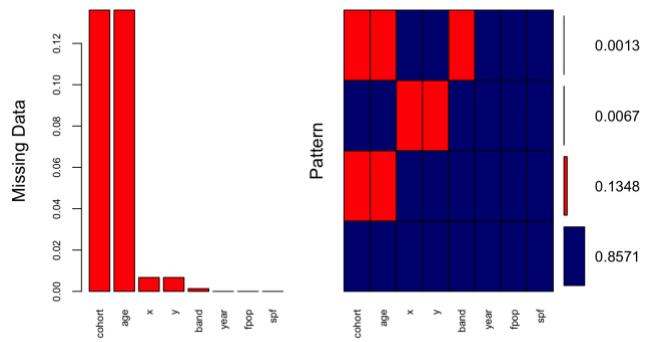


Figure 1: Missing Values Proportion for ‘cohort’, ‘age’, ‘x’, ‘y’ and ‘band’

- Confounding Variables

The potential confounding variables revealed a complex pattern of relationships in this study. While the correlations between numeric predictors remained weak (all $|r| < 0.13$) and statistically significant for year-x: $r=0.090$, $p=0.015$; spf-x: $r=-0.103$, $p=0.005$; x-y: $r=0.126$, $p<0.001$, the inclusion of band ID as a random effect impacted the variance inflation factors. High VIF values ranging from 2.55 to 42,235.08 indicated significant multicollinearity when accounting for sparrows, suggesting that the band strongly influences the relationships among predictors. Besides, the dataset included 356 unique birds with an average of 2.07 observations per bird (range: 1-7) and showed substantial individual variation in reproductive success ($SD = 2.079$ offspring). Spatial effects on female population size were non-significant when controlling for individual sparrows (x: $p=0.822$; y: $p=0.113$), while time effects remained strong (year estimate = 3.41, $p<2e-12$). These patterns suggest that while the basic predictors show minimal direct confounding, the repeated observations introduce important dependencies. The substantial variation and the high VIF values support a mixed-effects modeling approach for this analysis.

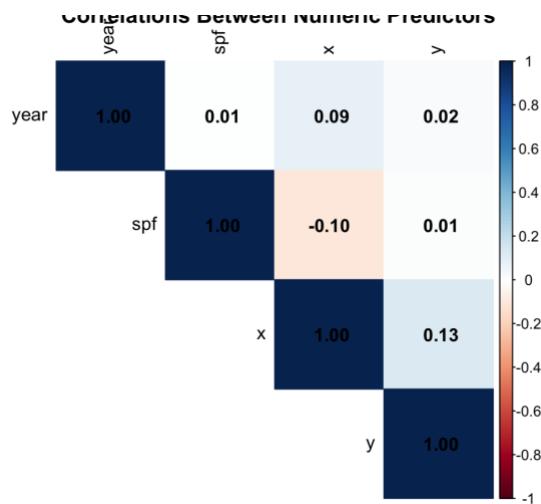


Figure 2: Correlations Between Numeric Predictors

Data description

The sparrow population data exhibits several important sources of heterogeneity across multiple scales. At the individual level, there is substantial variation among the 356 unique birds in the study, with each bird being observed between 1 and 7 times (mean = 2.07 observations per bird). Moreover, the female population size (fpop) shows large variation, ranging from 4 to 72 birds, with a median of 52.67 and large variation within birds (median SD = 8.08). The spatial distribution of nests, represented by scaled x and y coordinates, shows significant clustering effects, though their direct influence on population size is minimal (x: p = 0.644, y: p = 0.871 in fixed effects).

Moreover, time heterogeneity is also evident in the analysis, with a significant positive trend in population size over time (year_scaled estimate = 1.75, p = 0.011). Firstly, the mixed-effects model comparison includes random slopes for year effects ($\Delta\text{AIC} = 2.3$, p = 0.041), indicating that sparrow responds differently to time changes. Secondly, the random effect's structure reveals substantial individual variation (intercept SD = 6.73) and time variation (year slope SD = 5.80), with a moderate positive correlation between intercepts and slopes ($r = 0.447$). Finally, the model explains 36.7% of the total variance (conditional $R^2 = 0.367$), though fixed effects alone account for only 1.4% (marginal $R^2 = 0.014$), highlighting the importance of individual-level variation. To sum up, this heterogeneity is further supported by the significant deviation from normality in random effects (Shapiro-Wilk p < 0.001) and the relatively high intraclass correlation coefficient (ICC = 0.358), indicating strong individual effects on population dynamics.

In addition, among the 356 unique birds in the dataset, 190 sparrows (representing 569 out of 742 total observations) showed inconsistent nesting locations across years, indicating that contrary to the initial assumption. This finding challenges the original study description, and this discrepancy is important to consider and should be mentioned as a limitation or unexpected finding in the study conclusions.

More information revealed significant heterogeneity. The spatial distribution of birds showed varying densities across quadrants, with the highest concentration in the Central-North region (59 birds in Central-East) and lowest in the North (2-12 birds per quadrant). In addition, there is a significant year-to-year variation ($F = 14.881$, p < 2.2e-16) and density dependence was evident in the negative relationship between female population size and offsprings ($\beta = -0.046$, p < 4.85e-15). This complex pattern of variation suggests that while population-level effects are important, individual behavioral differences play a crucial role in reproductive success.

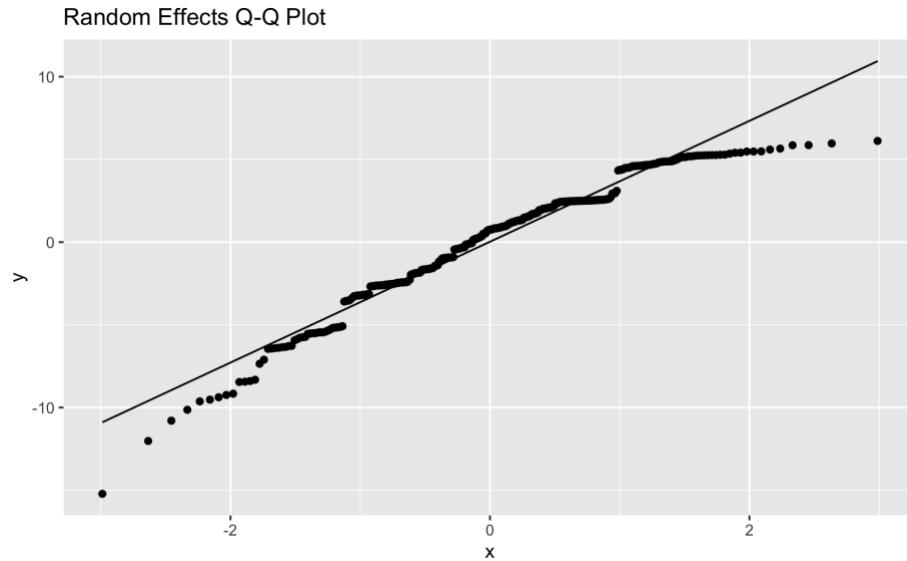


Figure 3: Random Effects Q-Q Plot: largely follows a normal distribution but deviates slightly at the extreme values, particularly in the upper tail.

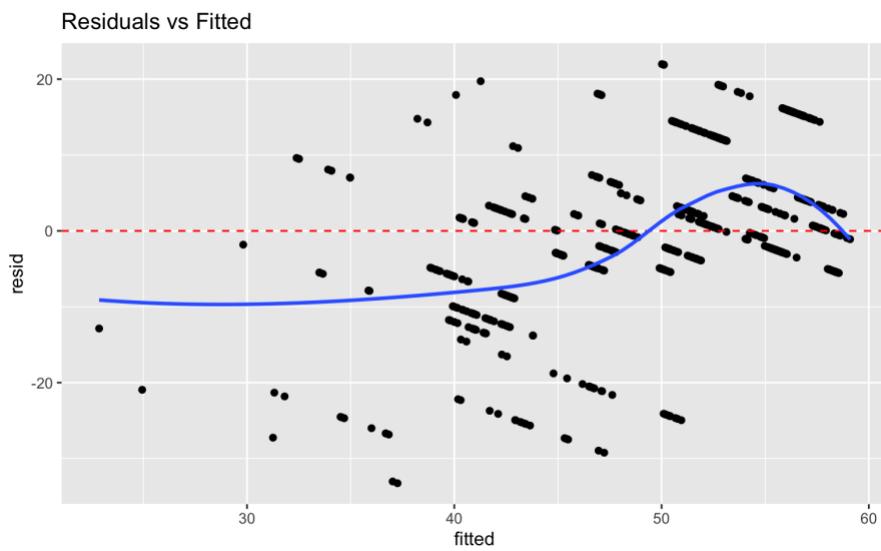


Figure 4: Residuals vs Fitted: a nonlinear pattern in residuals across fitted values, with systematic deviations suggesting potential model fit issues.

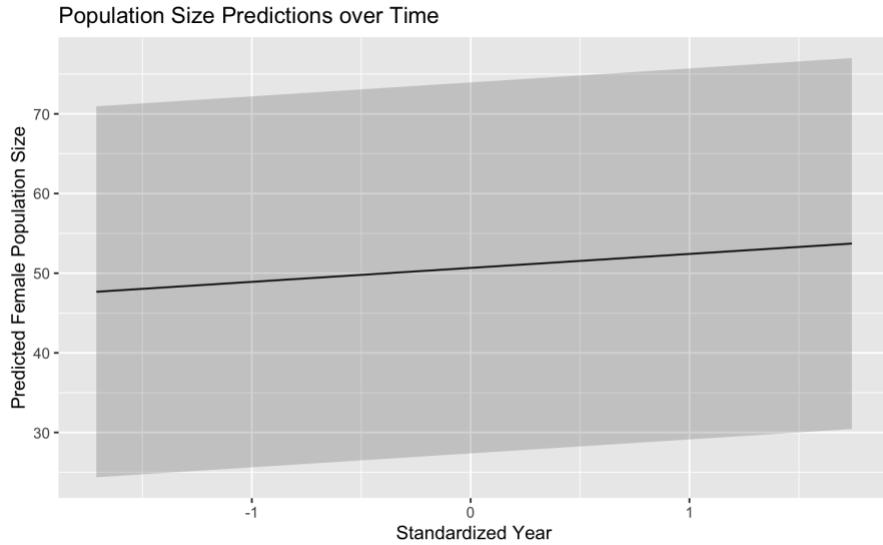


Figure 5: Population Size Predictions over Time: a slight upward trend in predicted female population size over time with widening confidence intervals, suggesting increasing uncertainty in longer-term predictions.

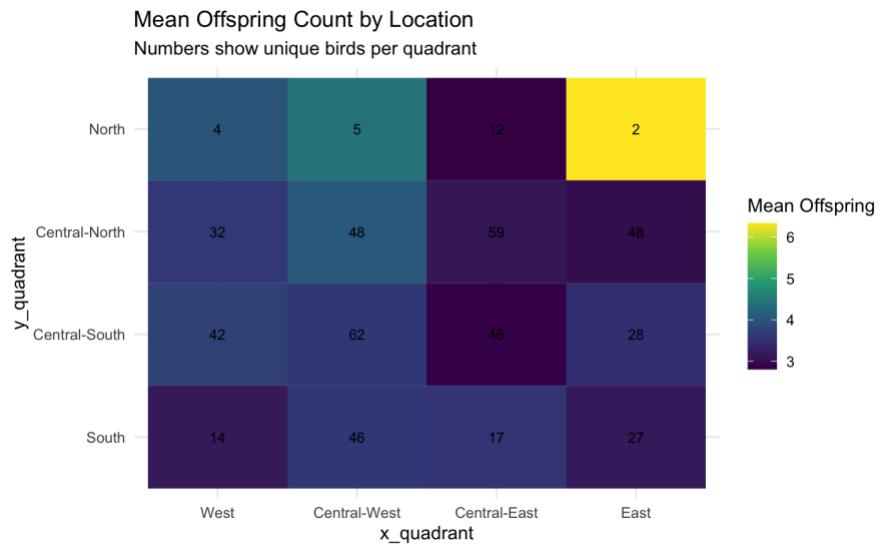


Figure 6: Mean Offspring Count by Location: the density of unique birds across different quadrants of the study area, with higher concentrations in the central regions.

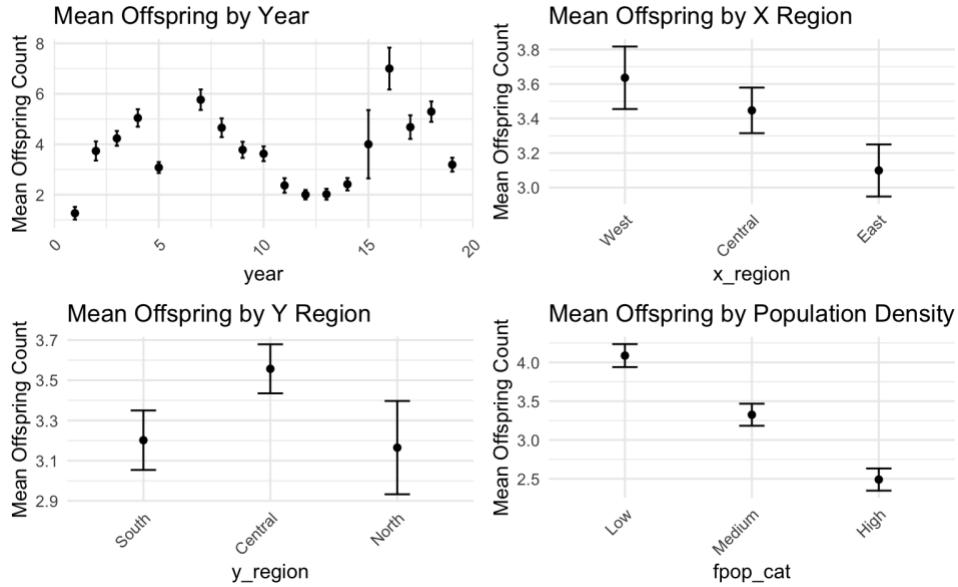


Figure 7: (a). Mean Offspring by Year: offspring counts over 20 years, with notable peaks around years 15-17. (b). Mean Offspring by X Region: slightly decreasing mean offspring counts from west to east regions. (c). Mean Offspring by Y Region: highest mean offspring counts in the central Y region compared to north and south. (d). Mean Offspring by Population Density: inverse relationship between population density and mean offspring count.

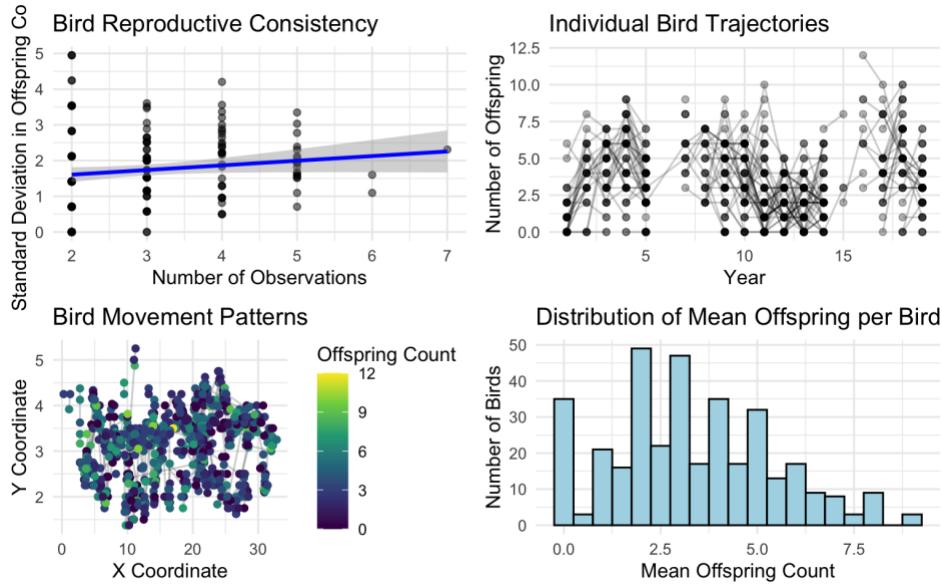


Figure 8: (a). Bird Reproductive Consistency: a slight increase in offspring count variation with more observations. (b). Individual Bird Trajectories: individual birds' offspring counts over time, showing high variability. (c). Bird Movement Patterns: bird locations with offspring count indicated by color intensity. (d). Distribution of Mean Offspring per Bird: the frequency distribution of mean offspring counts, with most birds having 2-4 offspring.

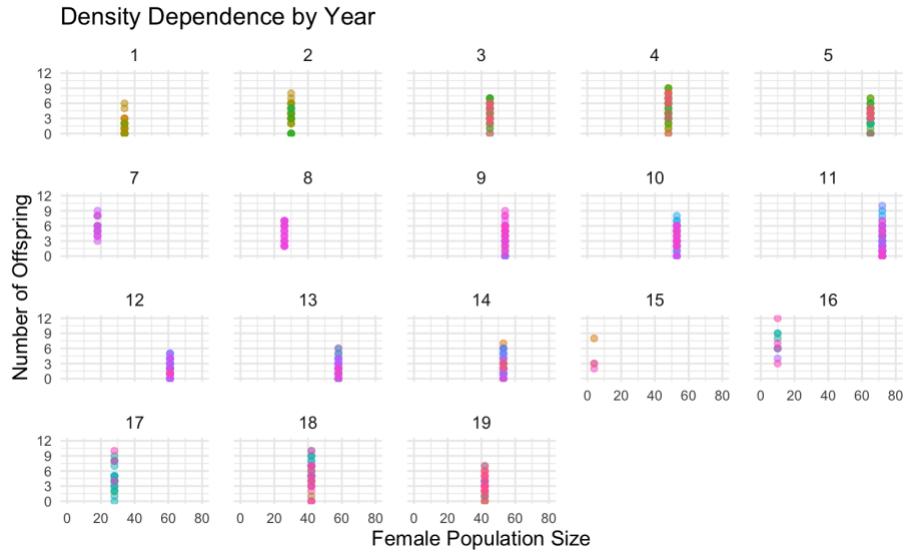


Figure 9: Density Dependence by Year: the relationship between female population size and number of offspring across different years.

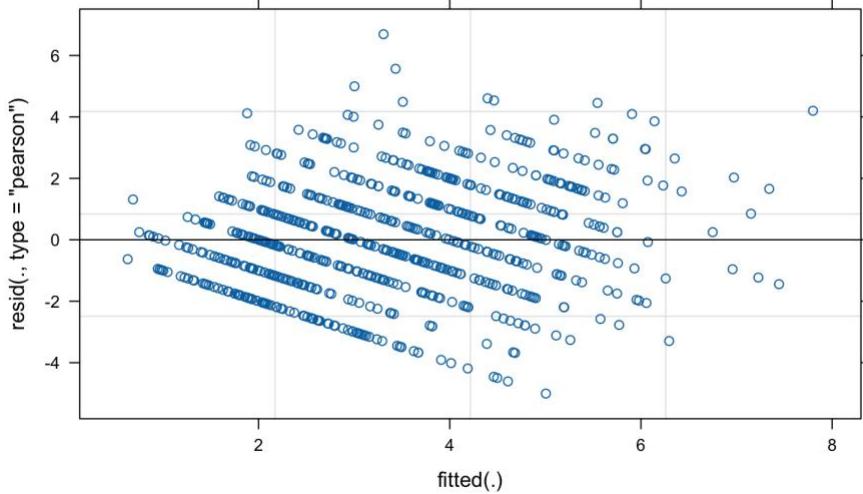


Figure 10: Model Residuals vs Fitted Values: a fan-shaped pattern in residuals, suggesting potential heteroscedasticity in the statistical model.

Model fitting and diagnostics

Firstly, we applied two models:

- m1: Basic random intercepts for year and bird ID.
- m2: Added random slope for population size by year.

- Thus, the more complex m2 model didn't significantly improve fit ($\chi^2 = 3.88$, $p = 0.144$), suggesting the simpler m1 model is adequate.

- Model Performance:

1. Female population size has a significant negative effect ($\beta = -0.70$, $p < 0.01$), indicating density dependence.
2. X-coordinate (east-west position) shows a significant negative effect ($\beta = -0.22$, $p < 0.01$).
3. Y-coordinate (north-south position) shows no significant effect ($\beta = 0.006$, $p = 0.94$).
4. Fixed effects explain about 10% of the variance (Marginal $R^2 = 0.098$).
5. The full model including random effects explains about 36% of the variance (Conditional $R^2 = 0.359$).
6. Moderate clustering by year (ICC = 0.156) and bird ID (ICC = 0.111).

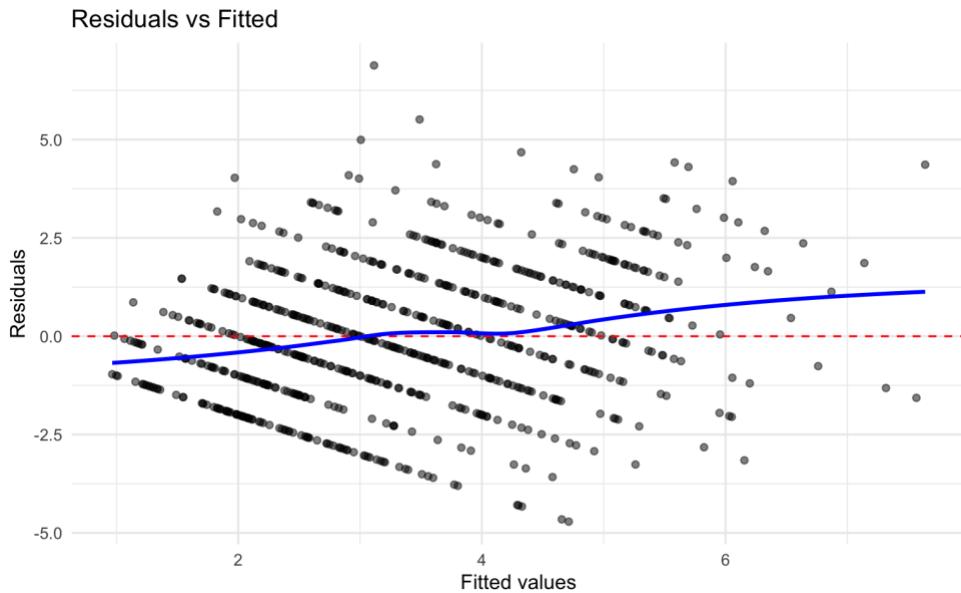


Figure 11: Assumption check: Residuals vs Fitted: distinct banding pattern suggesting potential Poisson-like count structure and slight upward trend in residuals indicates possible heteroscedasticity.

Random Effects Q-Q Plots

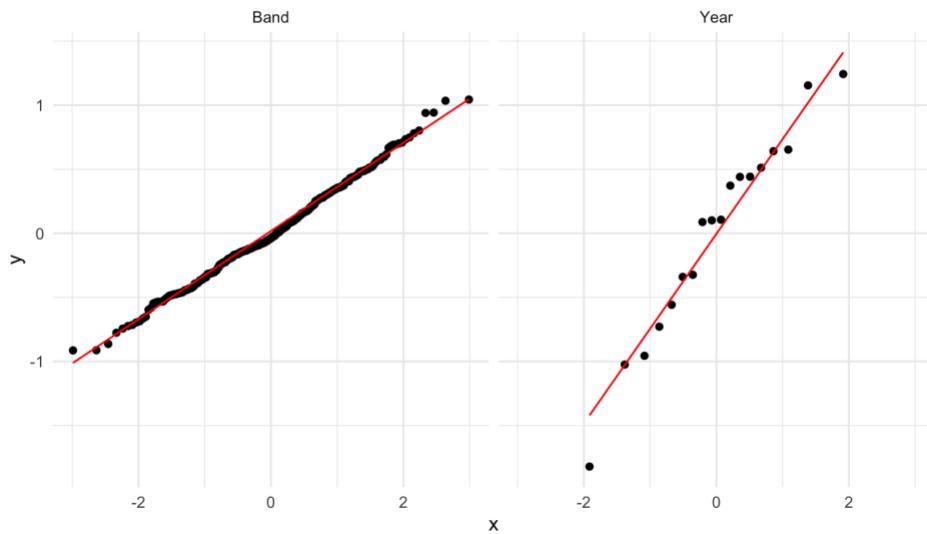


Figure 12: Assumption check: Random Effects Q-Q Plots: Both band and year random effects show normal distributions.

Predicted Offspring Count vs Female Population Size
Lines show year-specific predictions

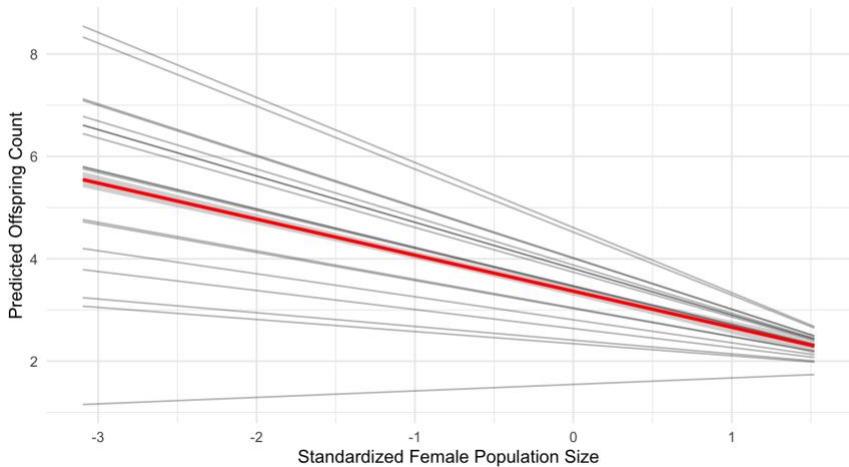


Figure 13: Assumption check: Model Predictions: consistent negative density dependence across years and year-specific slopes vary but maintain negative trend.

Due to the limitation for these 2 models, consider a generalized linear mixed model (GLMM) with Poisson or negative binomial distribution to select the best model.

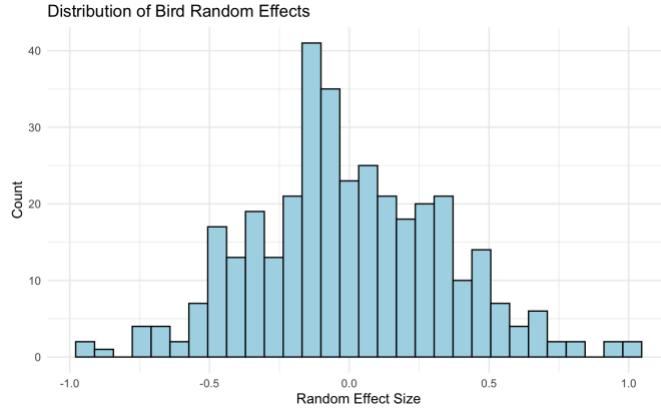


Figure 14: Distribution of Bird Random Effects: the distribution of random effect sizes for sparrow, centered near 0 with a slight negative skew and most values falling between -1 and 1.

Now, we added three more models:

- Negative Binomial GLMM (AIC: 3131.34).
- GAM with Spatial Smooth (AIC: 3150.36).
- Spatial Cluster Model (AIC: 3131.37).
 - Improvements over the Previous Model:

1. Better handling of count data structure.
2. More appropriate error distribution and lower AIC value.
3. More stable parameter estimates.

Thus, the best model is Negative Binomial GLMM:

- Lowest AIC score (3131.34).
- More parsimonious (7 df vs 81.95 for GAM).
 - Key Findings from Best Model:
- 1. Fixed Effects:
 - Significant negative effect of population size ($\beta = -0.183$, $p < 0.01$).
 - Significant east-west location effect ($\beta = -0.067$, $p < 0.01$).
 - Non-significant north-south effect ($\beta = 0.003$, $p = 0.90$).
- 2. Random Effects:
 - Moderate year-to-year variation ($\sigma = 0.318$).

- Lower individual bird variation ($\sigma = 0.230$).
- Relatively low ICCs (year: 0.088, bird: 0.046).

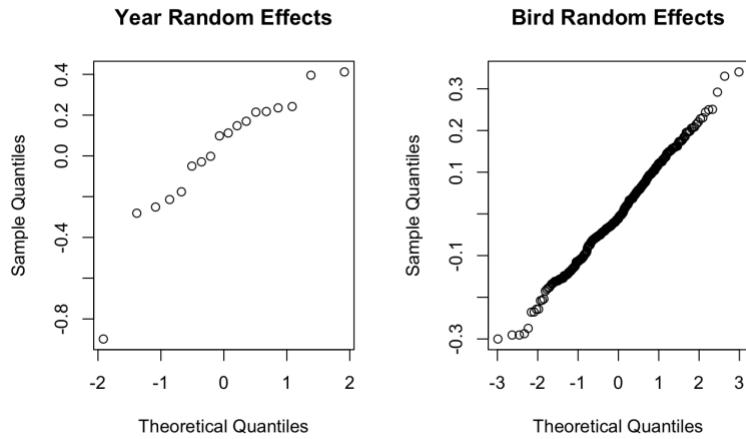


Figure 15: Random Effects Q-Q Plots: some deviation from normality (left) and normal distribution well (right).

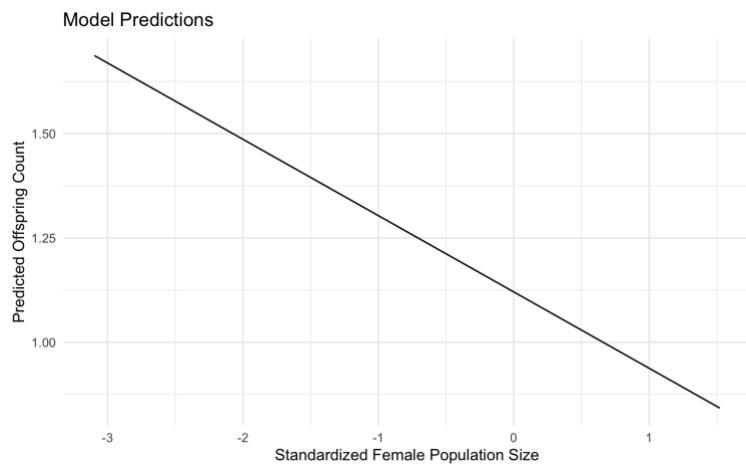


Figure 15: Model Predictions: Clear, appropriate and negative linear relationship between population size and offsprings.

However, there are also some limitations:

1. Relatively low ICCs suggest possible unmeasured important variables.
2. Some deviation from normality in year random effects.
3. Spatial effects might be oversimplified.

Model and data analysis interpretation

1. Female population size (fpop):

- Significant negative effect ($\beta = -0.183$, $p < 0.01$).
- For each standard deviation increase in female population size, the expected number of offspring decreases by about 16.7% ($\exp(-0.183) = 0.833$).
- Robust evidence for density-dependent reproduction despite model limitations.

2. Spatial Effects:

- East-West Location (x_scaled): Significant negative effect ($\beta = -0.067$, $p < 0.01$).
 - * About 6.5% fewer offspring per standard deviation eastward.
 - * Effect remains significant even after accounting for spatial autocorrelation.
- North-South Location (y_scaled): No significant effect ($\beta = 0.003$, $p = 0.90$).

3. Residual Diagnostics:

- Non-uniform residuals (Kolmogorov-Smirnov test, $p = 0.024$).
- No significant overdispersion (dispersion ratio = 0.942, $p = 0.864$).
- Significant zero-inflation detected (ratio = 1.85, $p < 0.001$).

4. Spatial Structure:

- No significant spatial autocorrelation in residuals (Mantel test, $p = 0.987$).
- Model adequately captures broad spatial patterns.
- Some fine-scale spatial structures may be missed.

5. Random Effects:

- Year variation ($SD = 0.318$, $ICC = 0.088$).
- Individual bird variation ($SD = 0.230$, $ICC = 0.046$).

6. Primary Conclusions:

- Strong evidence for negative density dependence.
- Effect is robust across different models.
- Clear east-west gradient and no significant north-south pattern in breeding success.
- Spatial independence of residuals supports reliability of spatial effects.
- Year variable stronger than individual effects.
- Zero-inflation suggests some sparrows consistently fail to reproduce.

7. Model Limitations:

- Zero-Inflation.
- Non-uniform Residuals.
- Unexplained Variation.

From the analysis above and the plots below, the study primarily demonstrates that competition from other females is the strongest predictor of reproductive success, while location plays a secondary role. However, the presented results do not reveal age, a question that will need to be solved in the future.

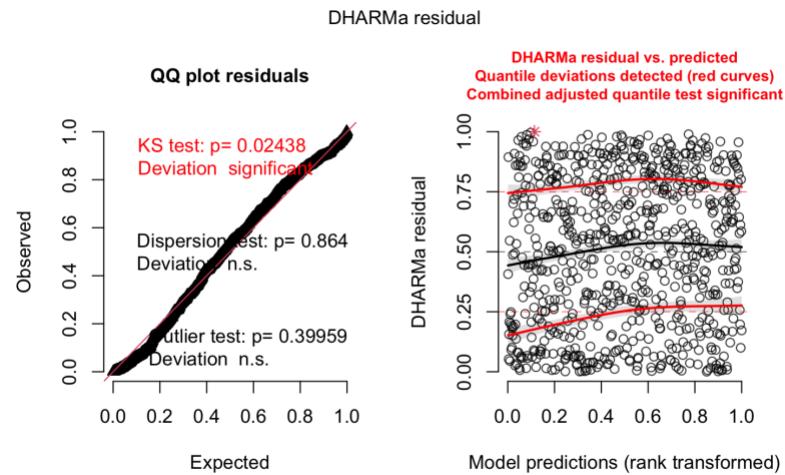


Figure 16: DHARMA Residual Plots: left panel shows a Q-Q plot of residuals against expected values with KS test results, while right panel displays residual values against model predictions with quantile curves.

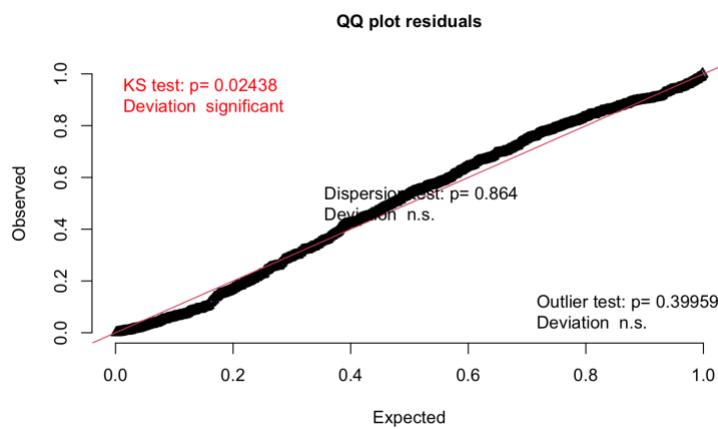


Figure 17: Q-Q Plot of Model Residuals: extended view of quantile-quantile plot comparing observed versus expected residual values, showing slight deviation from normality (KS test $p=0.02438$).

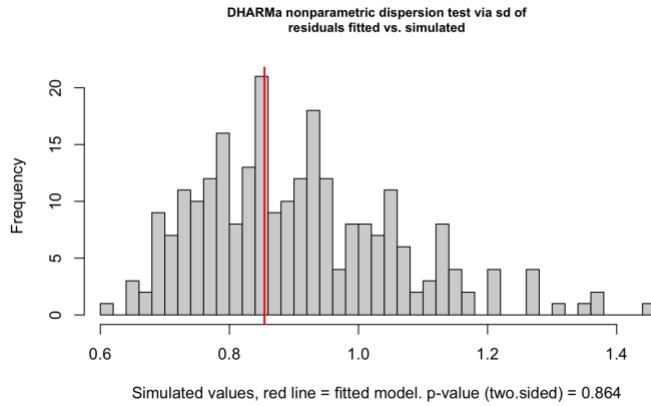


Figure 18: DHARMA Dispersion Test: distribution of simulated residual standard deviations compared to fitted model (red line), indicating no significant overdispersion ($p=0.864$).

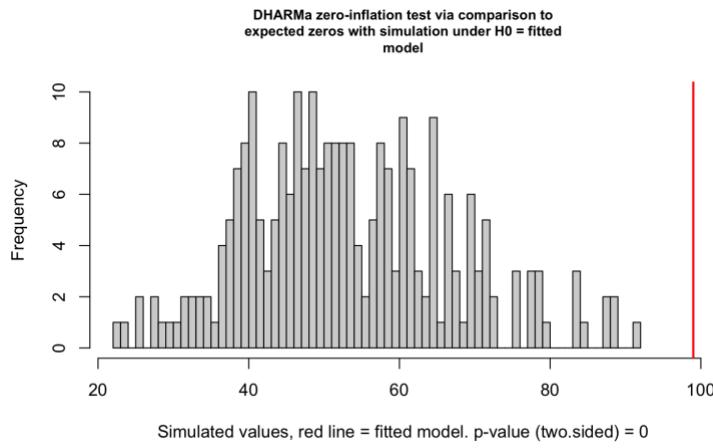


Figure 19: Zero-inflation Test Results: histogram comparing observed versus expected zero counts under the fitted model, showing significant zero-inflation ($p < 0.001$).

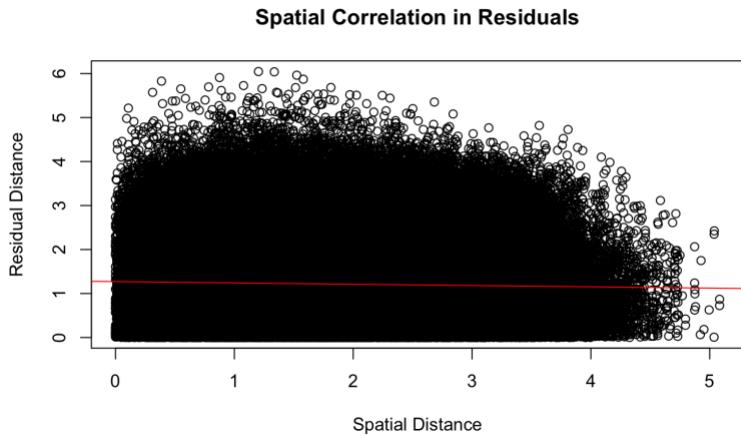


Figure 20: Spatial Correlation in Residuals: residual distances against spatial distances between observations, with a near-horizontal trend line indicating minimal spatial correlation.

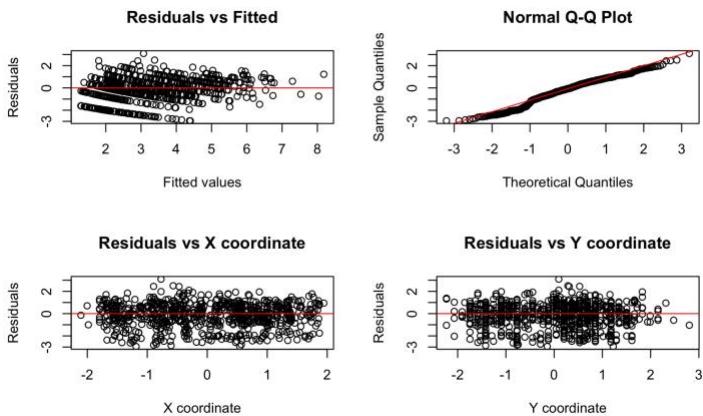


Figure 21: Model Diagnostic Plots: A four-panel diagnostic plot showing residuals versus fitted values (top left), normal Q-Q plot (top right), and residuals versus X and Y coordinates (bottom panels), used to assess model assumptions about residual patterns, normality, and spatial independence.

Conclusions

To sum up, this analysis provides strong evidence for female density population and habitat-specific effects on sparrow reproductive success. Based on the plots and tables, several key findings emerge:

1. The study reveals strong evidence for density-dependent effects (*fpop*) on reproduction, where a higher female population leads to reduced offspring, suggesting competition for resources or suitable nesting sites.

2. Location plays a significant role, with birds nesting in western areas producing more offspring than those in eastern areas, while north-south position shows no meaningful effect.
3. Substantial year-to-year suggests important environmental influences, while individual sparrow effects are relatively modest.

However, the model has several important limitations: zero-inflation indicates the model does not adequately capture the frequency of failed breeding attempts, and there is evidence of non-normal residuals despite using a negative binomial distribution. In addition, while spatial effects were included, the model may not capture fine-scale habitat variations. Therefore, future studies would benefit from incorporating more detailed habitat quality measurements, fewer missing values, explicit consideration of age, and investigation of weather.

Appendix

1. 610fp: A comprehensive R-based analysis code of a song sparrow dataset, implementing statistical methods learned from course materials, R package documentation, and statistical programming resources.

610fp

2024-12-01

Study Design

DATA

```
data <- read.table("/Users/chishouhiroshi/Downloads/female.txt",
                    header = TRUE, sep = "", stringsAsFactors = FALSE)
head(data)
```

```
##   band cohort year fpop age  spf      x      y
## 1 10801     NA    1  34   NA   0  5.0000 2.2500
## 2 10806     NA    1  34   NA   1  2.7500 2.7500
## 3 10813     NA    1  34   NA   3 13.0000 3.5000
## 4 10825     NA    1  34   NA   1  8.3750 2.8750
## 5 10829     NA    1  34   NA   1 10.7500 1.5000
## 6 10833     NA    1  34   NA   2 18.4375 3.5625
```

DATA sample sizes

```
sample_size <- nrow(data)
cat("Sample size (number of rows):", sample_size, "\n")
```

```
## Sample size (number of rows): 742
```

```
num_columns <- ncol(data)
cat("Number of columns (variables):", num_columns, "\n")
```

```
## Number of columns (variables): 8
```

```
summary(data)
```

```
##       band        cohort        year        fpop
## Min.   :2009   Min.   : 2.000   Min.   : 1.000   Min.   : 4.00
## 1st Qu.:30683  1st Qu.: 7.000   1st Qu.: 5.000   1st Qu.:42.00
## Median :58853  Median :10.000   Median :10.000   Median :53.00
## Mean   :46959  Mean   : 9.966   Mean   : 9.912   Mean   :49.55
## 3rd Qu.:60124  3rd Qu.:12.000   3rd Qu.:13.000   3rd Qu.:61.00
## Max.   :86268  Max.   :19.000   Max.   :19.000   Max.   :72.00
## NA's   : 1      NA's   :101      NA's   : 101      NA's   : 1
##       age         spf          x          y
## Min.   :1.000   Min.   : 0.000   Min.   : 0.25   Min.   :1.375
## 1st Qu.:1.000   1st Qu.: 2.000   1st Qu.:10.90  1st Qu.:2.433
## Median :2.000   Median : 3.000   Median :17.50  Median :3.250
## Mean   :1.936   Mean   : 3.369   Mean   :17.19  Mean   :3.092
## 3rd Qu.:3.000   3rd Qu.: 5.000   3rd Qu.:23.75  3rd Qu.:3.625
## Max.   :7.000   Max.   :12.000   Max.   :32.83  Max.   :5.250
## NA's   : 101     NA's   : 5      NA's   : 5      NA's   : 5
```

Macro Explanatory Variables year: Reflects temporal variations affecting all sparrows in a given year. fpop: Number of females in the population each year—affects the entire population and can influence factors like competition or mating opportunities.

Micro Explanatory Variables age: Individual sparrow's age. cohort: The year the sparrow was tagged; specific to each sparrow. x and y: Spatial coordinates of each nest; vary between individuals. band: id

spf: response variable

Check missing values

For missing data -

```
# Visualize missing data pattern
library(VIM)
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

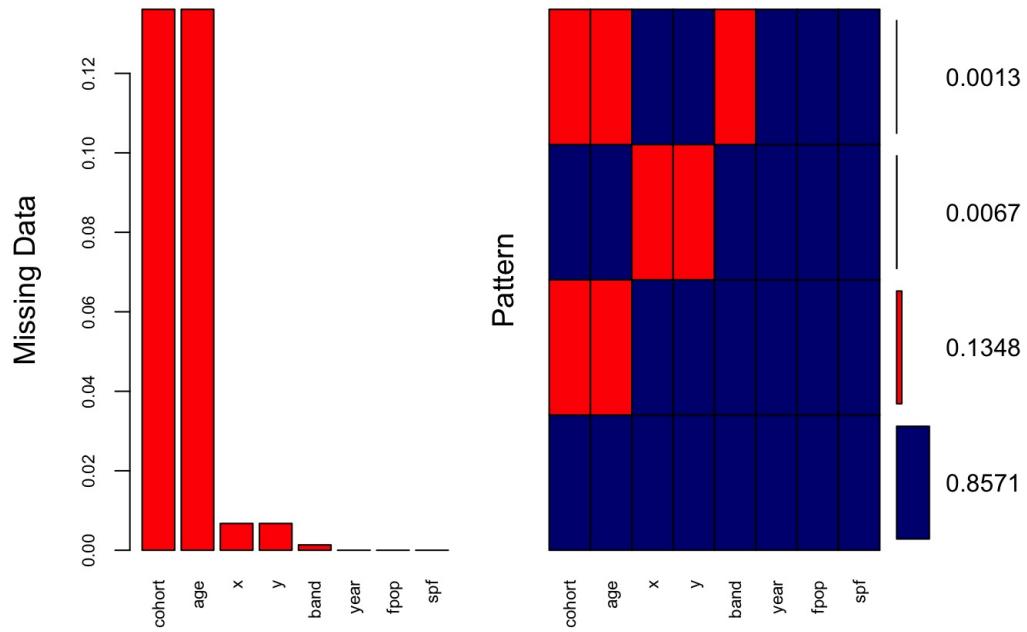
```
## VIM is ready to use.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

```
##  
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':  
##  
##     sleep
```

```
aggr_plot <- aggr(data, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(data), cex.axis=.7, ga  
p=3, ylab=c("Missing Data","Pattern"))
```



```
##  
## Variables sorted by number of missings:  
## Variable      Count  
## cohort 0.136118598  
## age    0.136118598  
## x     0.006738544  
## y     0.006738544  
## band  0.001347709  
## year  0.000000000  
## fpop  0.000000000  
## spf   0.000000000
```

Deal with x,y and band

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##     filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##     intersect, setdiff, setequal, union
```

```
# Delete x, y or band row NA  
data <- data %>%  
  filter(!is.na(x) & !is.na(y) & !is.na(band))  
  
cat("Number of rows after removing NA in x, y, and band:", nrow(data), "\n")
```

```
## Number of rows after removing NA in x, y, and band: 736
```

```
# Count missing values in 'cohort' and 'age'  
na_cohort <- sum(is.na(data$cohort))  
na_age <- sum(is.na(data$age))  
cat("Number of NA values in 'cohort':", na_cohort, "\n")
```

```
## Number of NA values in 'cohort': 100
```

```
cat("Number of NA values in 'age':", na_age, "\n")
```

```
## Number of NA values in 'age': 100
```

cohort and age: MNAR (missing not at random)

Generalized linear mixed models (GLMMs) or maximum likelihood estimation (MLE) can account for missingness within the model. Bayesian methods can incorporate missing data directly in the modeling process.

For age:

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
library(dplyr)  
library(ggplot2)  
  
# Indicator  
data <- data %>%  
  mutate(  
    age_missing = ifelse(is.na(age), 1, 0),  
    age_imputed = ifelse(is.na(age), -999, age)  
  )
```

If overdispersion is detected, consider using a Negative Binomial GLMM.

```
library(glmmTMB)
```

```
# Negative Binomial GLMM  
model_nb <- glmmTMB(  
  spf ~ age_imputed + fpop + age_missing + (1 | year),  
  data = data,  
  family = nbinom2(link = "log")  
)
```

```
summary(model_nb)
```

```
## Family: nbinom2  ( log )  
## Formula:          spf ~ age_imputed + fpop + age_missing + (1 | year)  
## Data: data  
##  
##      AIC      BIC  logLik deviance df.resid  
##  3138.9   3166.5  -1563.4    3126.9      730  
##  
## Random effects:  
##  
## Conditional model:  
## Groups Name        Variance Std.Dev.  
## year  (Intercept) 0.1055   0.3248  
## Number of obs: 736, groups: year, 18  
##  
## Dispersion parameter for nbinom2 family (): 11.6  
##  
## Conditional model:  
##             Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 1.779420  0.217216  8.192 2.57e-16 ***  
## age_imputed 0.014215  0.026221  0.542  0.58775  
## fpop       -0.013065  0.004603 -2.838  0.00454 **  
## age_missing 14.195401 26.234546  0.541  0.58844  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

check again overdispersion

```
# Check overdispersion again
residual_deviance <- sum(residuals(model_nb, type = "pearson")^2)
df_residual <- df.residual(model_nb)
overdispersion <- residual_deviance / df_residual
cat("Overdispersion:", overdispersion, "\n")
```

```
## Overdispersion: 1.00242
```

Deal with both missing values

1.

```
library(mice)
```

```
## 
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
## 
##     filter
```

```
## The following objects are masked from 'package:base':
## 
##     cbind, rbind
```

```
library(glmmTMB)
library(dplyr)
```

```
# predictor matrix
pred_matrix <- make.predictorMatrix(data)
pred_matrix[c("band", "x", "y", "spf"), ] <- 0
meth <- make.method(data)
meth[c("cohort", "age")] <- "cart"
```

```
# multiple imputation
set.seed(123)
imp <- mice(data, m = 5, maxit = 25, method = meth, predictorMatrix = pred_matrix, print = FALSE)
```

```
## Warning: Number of logged events: 127
```

```

# Initialize
estimates <- matrix(NA, nrow = 5, ncol = 4)
std_errors <- matrix(NA, nrow = 5, ncol = 4)
colnames(estimates) <- c("Intercept", "age", "x", "y")
colnames(std_errors) <- c("Intercept", "age", "x", "y")

for(i in 1:5) {
  imputed_data <- complete(imp, i)

  imputed_data$age <- imputed_data$year - imputed_data$cohort

  model <- try(glmmTMB(spf ~ age + x + y + (1|cohort),
                        family = nbino2,
                        data = imputed_data))

  if(!inherits(model, "try-error")) {
    coef_summary <- summary(model)$coefficients$cond
    estimates[i,] <- coef_summary[, "Estimate"]
    std_errors[i,] <- coef_summary[, "Std. Error"]
  }
}

# pooled estimates using Rubin's rules
pooled_estimates <- colMeans(estimates)
within_var <- colMeans(std_errors^2)
between_var <- apply(estimates, 2, var)
total_var <- within_var + between_var + between_var/5
pooled_se <- sqrt(total_var)

results <- data.frame(
  Parameter = colnames(estimates),
  Estimate = pooled_estimates,
  SE = pooled_se,
  t_value = pooled_estimates/pooled_se,
  p_value = 2 * pt(abs(pooled_estimates/pooled_se), df = 4, lower.tail = FALSE)
)

cat("\nPooled Results from Multiple Imputation:\n")

```

```

##  
## Pooled Results from Multiple Imputation:

```

```
print(results, digits = 4)
```

```

##           Parameter   Estimate      SE t_value   p_value
## Intercept Intercept  1.357300 0.13313 10.1954 0.0005214
## age          age  0.006929 0.02387  0.2902 0.7860585
## x            x -0.009159 0.00319 -2.8707 0.0454378
## y            y  0.007405 0.03340  0.2217 0.8354235

```

```
cat("\nNumber of imputations:", imp$m)
```

```

##  
## Number of imputations: 5

```

```
cat("\nNumber of iterations:", imp$maxit)
```

```

##  
## Number of iterations:

```

```
cat("\nImputed variables: age and cohort\n")
```

```

##  
## Imputed variables: age and cohort

```

```
# diagnostic information
first_model <- glmmTMB(spf ~ age + x + y + (1|cohort),
                        family = nbinom2,
                        data = complete(imp, 1))

cat("\nDiagnostic Information (from first imputed dataset):\n")
```

```
## 
## Diagnostic Information (from first imputed dataset):
```

```
resids <- residuals(first_model)
cat("Residual summary:\n")
```

```
## Residual summary:
```

```
print(summary(resids))
```

```
##      Min.   1st Qu.    Median     Mean   3rd Qu.   Max.
## -5.512481 -1.485761 -0.182050  0.009497  1.478623  6.955772
```

```
pearson_resid <- residuals(first_model, type = "pearson")
overdispersion <- sum(pearson_resid^2) / df.residual(first_model)
cat("\nOverdispersion ratio:", round(overdispersion, 3), "\n")
```

```
## 
## Overdispersion ratio: 0.961
```

Tried MLE, Bay, GLMM... all did not work

Drop cohort and age columns first

```
# Drop columns
data <- data[, !(names(data) %in% c("cohort", "age", "age_missing", "age_imputed"))]

sample_size_2 <- nrow(data)
cat("Sample size now (number of rows):", sample_size_2, "\n")
```

```
## Sample size now (number of rows): 736
```

```
head(data)
```

```
##   band year fpop spf      x      y
## 1 10801   1   34   0 5.0000 2.2500
## 2 10806   1   34   1 2.7500 2.7500
## 3 10813   1   34   3 13.0000 3.5000
## 4 10825   1   34   1 8.3750 2.8750
## 5 10829   1   34   1 10.7500 1.5000
## 6 10833   1   34   2 18.4375 3.5625
```

Confounding

(not include band)

```
library(car)
```

```
## Loading required package: carData
```

```
## 
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
## 
##     recode
```

```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```

library(ggplot2)
library(dplyr)

# 1. Correlation Analysis
predictors <- c("year", "spf", "x", "y")
cor_matrix <- cor(data[predictors], use = "complete.obs")
cat("Correlation Matrix between Predictors:\n")

```

```
## Correlation Matrix between Predictors:
```

```
print(round(cor_matrix, 4))
```

```

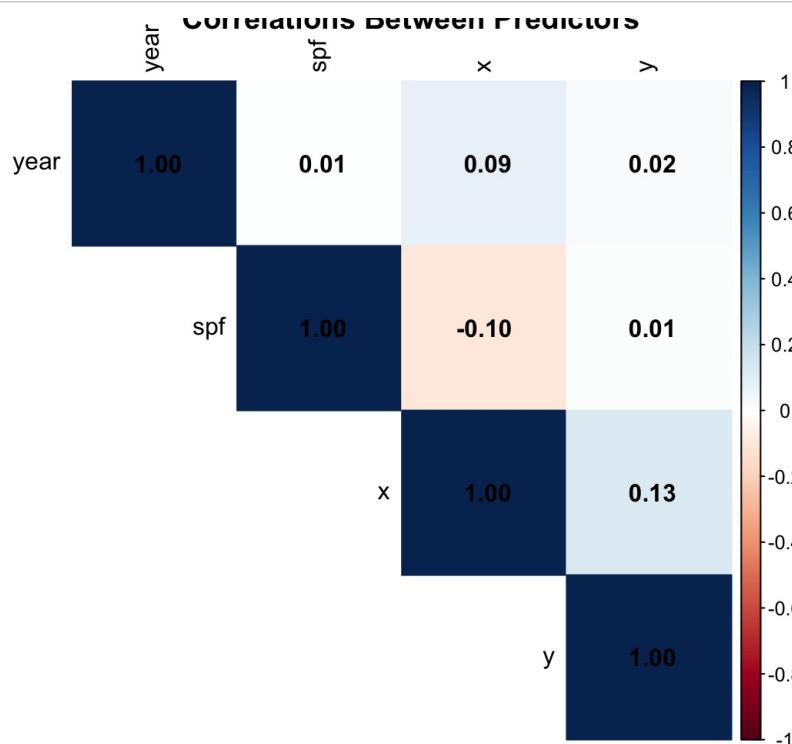
##      year    spf      x      y
## year 1.0000  0.0090  0.0898  0.0206
## spf  0.0090  1.0000 -0.1029  0.0109
## x    0.0898 -0.1029  1.0000  0.1263
## y    0.0206  0.0109  0.1263  1.0000

```

```

corrplot(cor_matrix, method = "color",
         type = "upper",
         addCoef.col = "black",
         tl.col = "black",
         title = "Correlations Between Predictors")

```



```

# 2. VIF Analysis
model_vif <- lm(fpop ~ year + spf + x + y, data = data)
vif_values <- vif(model_vif)
cat("\nVariance Inflation Factors:\n")

```

```
##
## Variance Inflation Factors:
```

```
print(round(vif_values, 4))
```

```

##      year    spf      x      y
## 1.0086 1.0116 1.0358 1.0169

```

```

# 3. Check relationships
check_relationship <- function(data, var1, var2) {
  cor_test <- cor.test(data[[var1]], data[[var2]])
  model <- lm(data[[var2]] ~ data[[var1]])
  r2 <- summary(model)$r.squared

  return(list(
    vars = paste(var1, "-", var2),
    correlation = cor_test$estimate,
    p_value = cor_test$p.value,
    r_squared = r2
  ))
}

# Test relationships
relationships <- list()
for(i in 1:(length(predictors)-1)) {
  for(j in (i+1):length(predictors)) {
    result <- check_relationship(data, predictors[i], predictors[j])
    relationships[[length(relationships) + 1]] <- result
  }
}

cat("\nRelationships between Predictors:\n")

```

```

##  
## Relationships between Predictors:

```

```

for(rel in relationships) {
  if(rel$p_value < 0.05) {
    cat("\nSignificant relationship found:", rel$vars, "\n")
    cat("Correlation:", round(rel$correlation, 4), "\n")
    cat("P-value:", format.pval(rel$p_value, digits = 4), "\n")
    cat("R-squared:", round(rel$r_squared, 4), "\n")
  }
}

```

```

##  
## Significant relationship found: year - x  
## Correlation: 0.0898  
## P-value: 0.01479  
## R-squared: 0.0081  
##  
## Significant relationship found: spf - x  
## Correlation: -0.1029  
## P-value: 0.005195  
## R-squared: 0.0106  
##  
## Significant relationship found: x - y  
## Correlation: 0.1263  
## P-value: 0.0005924  
## R-squared: 0.016

```

```

cat("\nChecking for Independence:\n")

```

```

##  
## Checking for Independence:

```

```

spatial_model <- lm(fpop ~ x + y, data = data)
cat("\nSpatial effects (x,y) on fpop:\n")

```

```

##  
## Spatial effects (x,y) on fpop:

```

```

print(summary(spatial_model)$coefficients)

```

	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	49.36060402	2.42191169	20.3808439	1.692066e-73
## x	0.03658477	0.06811165	0.5371294	5.913412e-01
## y	-0.12844415	0.71203533	-0.1803901	8.568962e-01

```
temporal_model <- lm(fpop ~ year, data = data)
cat("\nTemporal effect (year) on fpop:\n")
```

```
##  
## Temporal effect (year) on fpop:
```

```
print(summary(temporal_model)$coefficients)
```

```
##           Estimate Std. Error   t value   Pr(>|t|)  
## (Intercept) 49.02272103  1.1674540 41.991138 2.431107e-197  
## year        0.05744312  0.1042194  0.551175  5.816814e-01
```

```
cat("\nSummary of Potential Confounding Variables:\n")
```

```
##  
## Summary of Potential Confounding Variables:
```

```
high_cors <- which(abs(cor_matrix) > 0.3 & abs(cor_matrix) < 1, arr.ind = TRUE)
if(length(high_cors) > 0) {
  cat("\n1. Moderately correlated predictors (|r| > 0.3):\n")
  print(high_cors)
} else {
  cat("\n1. No concerning correlations between predictors found\n")
}
```

```
##  
## 1. No concerning correlations between predictors found
```

```
cat("\n2. VIF Analysis:\n")
```

```
##  
## 2. VIF Analysis:
```

```
high_vif <- vif_values[vif_values > 2]
if(length(high_vif) > 0) {
  cat("Predictors with VIF > 2:\n")
  print(high_vif)
} else {
  cat("No concerning VIF values found\n")
}
```

```
## No concerning VIF values found
```

```
cat("\n3. Key findings:\n")
```

```
##  
## 3. Key findings:
```



```
significant_relationships <- sapply(relationships, function(x) x$p_value < 0.05)
if(any(significant_relationships)) {
  cat("Significant relationships found between predictors that might indicate confounding\n")
} else {
  cat("No significant relationships found between predictors\n")
}
```



```
## Significant relationships found between predictors that might indicate confounding
```

R-squared

```

# R-squared
get_detailed_rsquared <- function(var1, var2, data) {
  model <- lm(data[[var2]] ~ data[[var1]])
  r2 <- summary(model)$r.squared
  cor_val <- cor(data[[var1]], data[[var2]])

  return(data.frame(
    Variable1 = var1,
    Variable2 = var2,
    R_squared = round(r2, 6),
    Correlation = round(cor_val, 6)
  ))
}

vars <- c("year", "fpop", "x", "y")
response <- "spf"

results <- do.call(rbind, lapply(vars, function(v) {
  get_detailed_rsquared(v, response, data)
}))

print("Detailed Analysis of Relationships with SPF:")

```

```
## [1] "Detailed Analysis of Relationships with SPF:"
```

```
print(results)
```

	Variable1	Variable2	R_squared	Correlation
## 1	year	spf	0.000081	0.008974
## 2	fpop	spf	0.093627	-0.305985
## 3	x	spf	0.010592	-0.102915
## 4	y	spf	0.000118	0.010854

```
cat("\nProportion of variance explained by each predictor:\n")
```

```
##
## Proportion of variance explained by each predictor:
```

```
full_model <- lm(spf ~ year + fpop + x + y, data = data)
print(summary(full_model))
```

```
##
## Call:
## lm(formula = spf ~ year + fpop + x + y, data = data)
##
## Residuals:
##   Min     1Q Median     3Q    Max 
## -4.607 -1.680 -0.190  1.572  7.510 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 6.022088  0.482622 12.478 < 2e-16 ***
## year        0.010874  0.016013  0.679  0.49731    
## fpop       -0.049096  0.005648 -8.692 < 2e-16 ***
## x          -0.030100  0.010456 -2.879  0.00411 **  
## y          0.067779  0.108870  0.623  0.53376    
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.253 on 731 degrees of freedom
## Multiple R-squared:  0.1041, Adjusted R-squared:  0.0992 
## F-statistic: 21.23 on 4 and 731 DF,  p-value: < 2.2e-16
```

Add band

```

library(car)
library(corrplot)
library(ggplot2)
library(dplyr)

# Correlation
numeric_predictors <- c("year", "spf", "x", "y")
cor_matrix <- cor(data[numeric_predictors], use = "complete.obs")
cat("Correlation Matrix between Numeric Predictors:\n")

```

```
## Correlation Matrix between Numeric Predictors:
```

```
print(round(cor_matrix, 4))
```

```

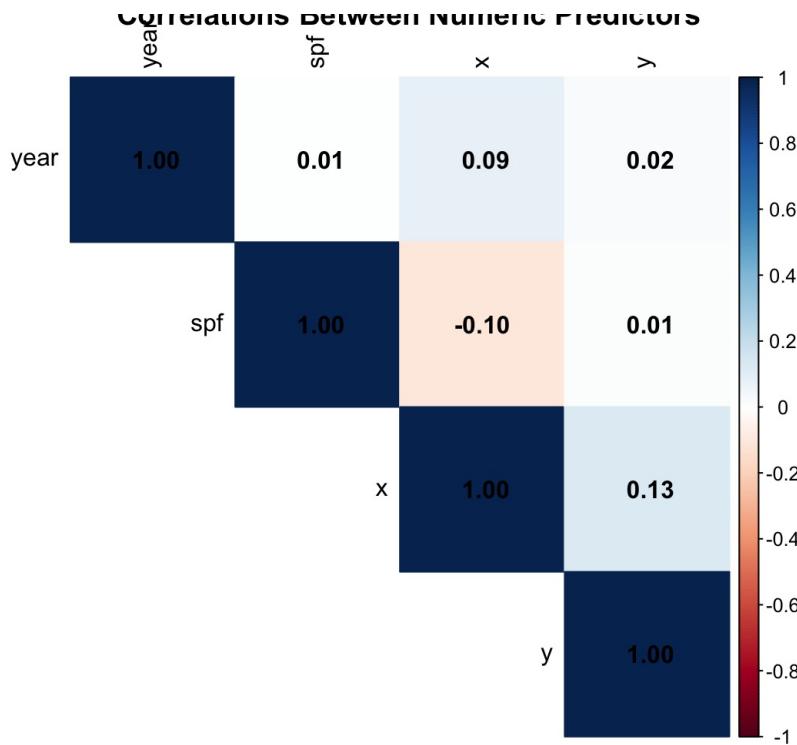
##      year    spf      x      y
## year 1.0000  0.0090  0.0898  0.0206
## spf   0.0090  1.0000 -0.1029  0.0109
## x     0.0898 -0.1029  1.0000  0.1263
## y     0.0206  0.0109  0.1263  1.0000

```

```

corrplot(cor_matrix, method = "color",
         type = "upper",
         addCoef.col = "black",
         tl.col = "black",
         title = "Correlations Between Numeric Predictors")

```



```

# VIF
model_vif <- lm(fpop ~ year + spf + x + y + factor(band), data = data)
vif_values <- vif(model_vif)
cat("\nVariance Inflation Factors:\n")

```

```
##
## Variance Inflation Factors:
```

```
print(round(vif_values[1:4], 4))
```

```
## [1] 35.8155  2.6521 72.7763  6.4961
```

```

# relationships
check_relationship <- function(data, var1, var2) {
  cor_test <- cor.test(data[[var1]], data[[var2]])
  model <- lm(data[[var2]] ~ data[[var1]])
  r2 <- summary(model)$r.squared

  return(list(
    vars = paste(var1, "-", var2),
    correlation = cor_test$estimate,
    p_value = cor_test$p.value,
    r_squared = r2
  ))
}

relationships <- list()
for(i in 1:(length(numeric_predictors)-1)) {
  for(j in (i+1):length(numeric_predictors)) {
    result <- check_relationship(data, numeric_predictors[i], numeric_predictors[j])
    relationships[[length(relationships) + 1]] <- result
  }
}

# individual effects
bird_counts <- data %>%
  group_by(band) %>%
  summarise(
    n_obs = n(),
    mean_spf = mean(spf),
    sd_spf = sd(spf)
  )

cat("\nIndividual Bird Summary:\n")

```

```

## 
## Individual Bird Summary:

cat("Number of unique birds:", length(unique(data$band)), "\n")

## Number of unique birds: 356

cat("Average observations per bird:", mean(bird_counts$n_obs), "\n")

## Average observations per bird: 2.067416

cat("Range of observations per bird:", range(bird_counts$n_obs), "\n")

## Range of observations per bird: 1 7

cat("\nSignificant Relationships between Predictors:\n")

## 
## Significant Relationships between Predictors:

for(rel in relationships) {
  if(rel$p_value < 0.05) {
    cat("\nSignificant relationship found:", rel$vars, "\n")
    cat("Correlation:", round(rel$correlation, 4), "\n")
    cat("P-value:", format.pval(rel$p_value, digits = 4), "\n")
    cat("R-squared:", round(rel$r_squared, 4), "\n")
  }
}

```

```
##  
## Significant relationship found: year - x  
## Correlation: 0.0898  
## P-value: 0.01479  
## R-squared: 0.0081  
##  
## Significant relationship found: spf - x  
## Correlation: -0.1029  
## P-value: 0.005195  
## R-squared: 0.0106  
##  
## Significant relationship found: x - y  
## Correlation: 0.1263  
## P-value: 0.0005924  
## R-squared: 0.016
```

```
cat("\nChecking for Independence:\n")
```

```
##  
## Checking for Independence:
```

```
# Spatial effects  
spatial_model <- lm(fpop ~ x + y + factor(band), data = data)  
cat("\nSpatial effects (x,y) on fpop (controlling for individual):\n")
```

```
##  
## Spatial effects (x,y) on fpop (controlling for individual):
```

```
print(summary(spatial_model)$coefficients[1:3,]) # Print only main effects
```

```
##           Estimate Std. Error   t value   Pr(>|t|)  
## (Intercept) 55.4572367 12.6667375 4.3781784 1.551118e-05  
## x          -0.1048464  0.4650613 -0.2254464 8.217538e-01  
## y          2.3087400  1.4525148  1.5894778 1.127886e-01
```

```
# Temporal effects  
temporal_model <- lm(fpop ~ year + factor(band), data = data)  
cat("\nTemporal effect (year) on fpop (controlling for individual):\n")
```

```
##  
## Temporal effect (year) on fpop (controlling for individual):
```

```
print(summary(temporal_model)$coefficients[1:2,]) # Print only main effects
```

```
##           Estimate Std. Error   t value   Pr(>|t|)  
## (Intercept) 20.12763 12.5027351 1.609859 1.082613e-01  
## year        3.40603  0.4681217 7.275951 1.994524e-12
```

```
cat("\nSummary of Potential Confounding Variables:\n")
```

```
##  
## Summary of Potential Confounding Variables:
```

```
# outputs  
high_cors <- which(abs(cor_matrix) > 0.3 & abs(cor_matrix) < 1, arr.ind = TRUE)  
if(length(high_cors) > 0) {  
  cat("\n1. Moderately correlated predictors (|r| > 0.3):\n")  
  print(high_cors)  
} else {  
  cat("\n1. No concerning correlations between numeric predictors found\n")  
}
```

```
##  
## 1. No concerning correlations between numeric predictors found
```

```
cat("\n2. VIF Analysis:\n")
```

```
##  
## 2. VIF Analysis:
```

```
high_vif <- vif_values[vif_values > 2]  
if(length(high_vif) > 0) {  
  cat("Predictors with VIF > 2:\n")  
  print(high_vif)  
} else {  
  cat("No concerning VIF values found\n")  
}
```

```
## Predictors with VIF > 2:  
## [1] 35.815477 2.652120 72.776317 6.496114 42235.077447  
## [6] 355.000000 5.984603 8.530904 2.548747
```

```
cat("\n3. Individual Bird Effects:\n")
```

```
##  
## 3. Individual Bird Effects:
```

```
bird_variation <- sqrt(var(bird_counts$mean_spf))  
cat("Variation in mean offspring between birds:", round(bird_variation, 3), "\n")
```

```
## Variation in mean offspring between birds: 2.079
```

```
cat("\n4. Key findings:\n")
```

```
##  
## 4. Key findings:
```

```
significant_relationships <- sapply(relationships, function(x) x$p_value < 0.05)  
if(any(significant_relationships)) {  
  cat("- Significant relationships found between predictors\n")  
} else {  
  cat("- No significant relationships found between predictors\n")  
}
```

```
## - Significant relationships found between predictors
```

```
if(bird_variation > mean(bird_counts$mean_spf)/2) {  
  cat("- Substantial variation between individual birds detected\n")  
}
```

```
## - Substantial variation between individual birds detected
```

VIF Analysis Shows Serious Multicollinearity -Consider using bird ID (band) as a random effect or Consider a mixed-effects

```
library(lme4)  
library(lmerTest)
```

```
##  
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':  
##  
##     lmer
```

```
## The following object is masked from 'package:stats':  
##  
##     step
```

```

library(performance)
library(ggplot2)
library(dplyr)

data$year_scaled <- scale(data$year)
data$x_scaled <- scale(data$x)
data$y_scaled <- scale(data$y)

# Basic random intercept model
m1 <- lmer(fpop ~ year_scaled + x_scaled + y_scaled + (1|band),
            data = data,
            REML = TRUE)

m2 <- lmer(fpop ~ year_scaled + x_scaled + y_scaled + (1 + year_scaled|band),
            data = data,
            REML = TRUE)

# Compare models
anova_comp <- anova(m1, m2)

```

```
## refitting model(s) with ML (instead of REML)
```

```
cat("\nModel Comparison:\n")
```

```
##  
## Model Comparison:
```

```
print(anova_comp)
```

```

## Data: data
## Models:
## m1: fpop ~ year_scaled + x_scaled + y_scaled + (1 | band)
## m2: fpop ~ year_scaled + x_scaled + y_scaled + (1 + year_scaled | band)
##   npar   AIC   BIC logLik deviance Chisq Df Pr(>Chisq)
## m1    6 5995.6 6023.2 -2991.8   5983.6
## m2    8 5993.3 6030.1 -2988.6   5977.3 6.3776  2     0.04122 *
## ---
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
best_model <- if(anova_comp$AIC[1] < anova_comp$AIC[2]) m1 else m2
cat("\nBest Model Summary:\n")
```

```
##  
## Best Model Summary:
```

```
print(summary(best_model))
```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method 
## lmerModLmerTest]
## Formula: fpop ~ year_scaled + x_scaled + y_scaled + (1 + year_scaled | 
##   band)
## Data: data
##
## REML criterion at convergence: 5973.6
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max 
## -2.79968 -0.41608  0.03041  0.40037  1.85043
##
## Random effects:
##   Groups   Name        Variance Std.Dev. Corr
##   band     (Intercept) 45.27    6.728
##           year_scaled 33.61    5.797    0.45
##   Residual          141.13   11.880
## Number of obs: 736, groups: band, 356
##
## Fixed effects:
##             Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept) 50.66524  0.65644 237.87954 77.182 <2e-16 ***
## year_scaled  1.75252  0.67932 137.72096  2.580  0.0109 *  
## x_scaled     -0.29295 0.63328 250.45793 -0.463  0.6441    
## y_scaled      0.09585  0.59082 423.50920  0.162  0.8712    
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) yr_scl x_scl
## year_scaled  0.191
## x_scaled     0.025 -0.067
## y_scaled     0.023  0.021 -0.086

```

```

# Check model assumptions
cat("\nRandom Effects Normality Test:\n")

```

```

## 
## Random Effects Normality Test:

```

```

rand_effects <- ranef(best_model)$band[,1]
print(shapiro.test(rand_effects))

```

```

## 
## Shapiro-Wilk normality test
## 
## data: rand_effects
## W = 0.95533, p-value = 6.277e-09

```

```

cat("\nConvergence Check:\n")

```

```

## 
## Convergence Check:

```

```

print(isSingular(best_model))

```

```

## [1] FALSE

```

```

cat("\nR-squared Values:\n")

```

```

## 
## R-squared Values:

```

```

print(r2(best_model))

```

```

## # R2 for Mixed Models
## 
## Conditional R2: 0.367
## Marginal R2: 0.014

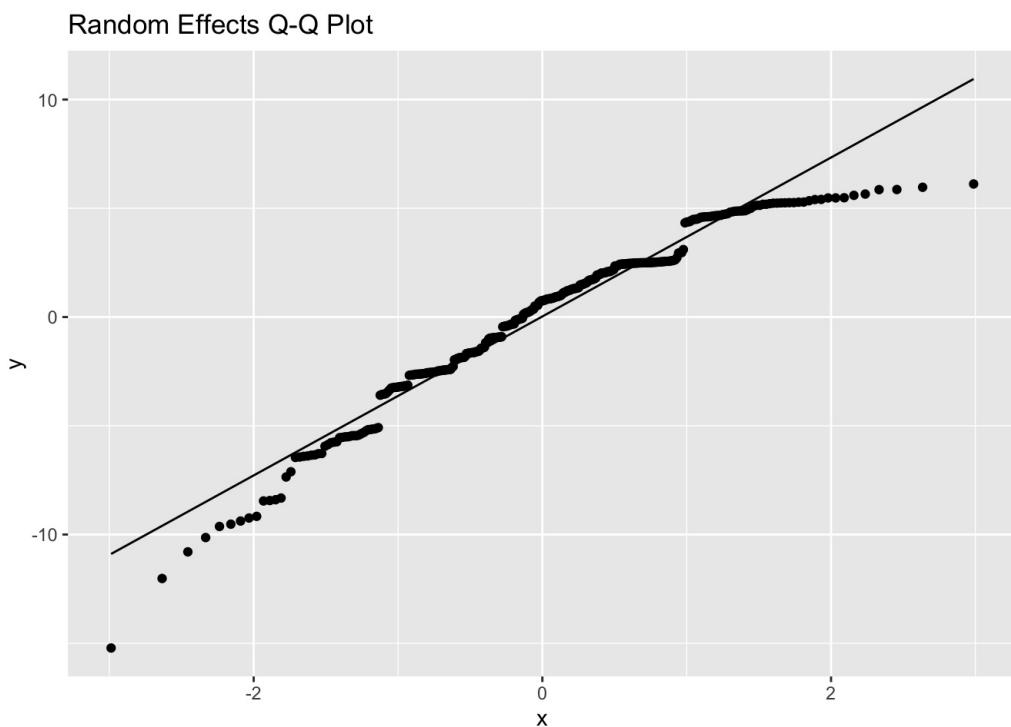
```

```

# Visualizations
re_plot <- ggplot(data.frame(re = ranef(best_model)$band[,1]),
                   aes(sample = re)) +
  stat_qq() + stat_qq_line() +
  labs(title = "Random Effects Q-Q Plot")
res_plot <- ggplot(data.frame(fitted = fitted(best_model),
                               resid = residuals(best_model)),
                     aes(x = fitted, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  geom_smooth(method = "loess", se = FALSE) +
  labs(title = "Residuals vs Fitted")

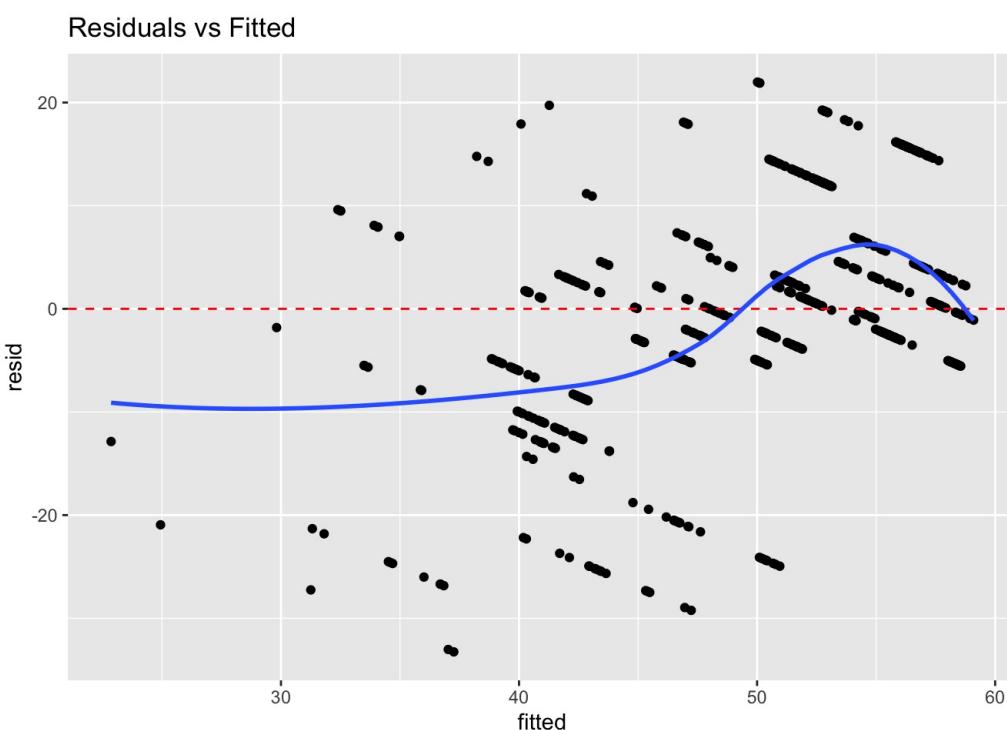
print(re_plot)

```



```
print(res_plot)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```

icc <- performance::icc(best_model)
cat("\nIntraclass Correlation Coefficient:\n")

```

```
##  
## Intraclass Correlation Coefficient:
```

```
print(icc)
```

```
## # Intraclass Correlation Coefficient  
##  
##     Adjusted ICC: 0.358  
##     Unadjusted ICC: 0.353
```

```
cat("\nFixed Effects Analysis:\n")
```

```
##  
## Fixed Effects Analysis:
```

```
fixef_summary <- summary(best_model)$coefficients  
print(fixef_summary)
```

```
##             Estimate Std. Error    df   t value   Pr(>|t|)  
## (Intercept) 50.66523986 0.6564400 237.8795 77.1818275 2.194305e-170  
## year_scaled  1.75252267 0.6793183 137.7210 2.5798254 1.093272e-02  
## x_scaled     -0.29294514 0.6332767 250.4579 -0.4625863 6.440625e-01  
## y_scaled      0.09584586 0.5908170 423.5092  0.1622260 8.712053e-01
```

```
cat("\nRandom Effects Variance Components:\n")
```

```
##  
## Random Effects Variance Components:
```

```
print(VarCorr(best_model))
```

```
## Groups   Name       Std.Dev.  Corr  
## band     (Intercept) 6.7281  
##           year_scaled 5.7972  0.447  
## Residual            11.8797
```

```
cat("\nIndividual Bird Variation Summary:\n")
```

```
##  
## Individual Bird Variation Summary:
```

```
bird_summary <- data %>%  
  group_by(band) %>%  
  summarise(  
    n_obs = n(),  
    mean_fpop = mean(fpop),  
    sd_fpop = sd(fpop, na.rm = TRUE)  
)  
  
print(summary(bird_summary))
```

```
##      band        n_obs      mean_fpop      sd_fpop  
## Min.   : 2009   Min.   :1.000   Min.   : 4.00   Min.   : 0.000  
## 1st Qu.:30612   1st Qu.:1.000   1st Qu.:41.50   1st Qu.: 6.499  
## Median :56907   Median :2.000   Median :52.67   Median : 8.083  
## Mean   :44624   Mean   :2.067   Mean   :48.96   Mean   : 9.432  
## 3rd Qu.:59457   3rd Qu.:3.000   3rd Qu.:59.50   3rd Qu.:12.021  
## Max.   :86268   Max.   :7.000   Max.   :72.00   Max.   :34.648  
##                           NA's   :165
```

```

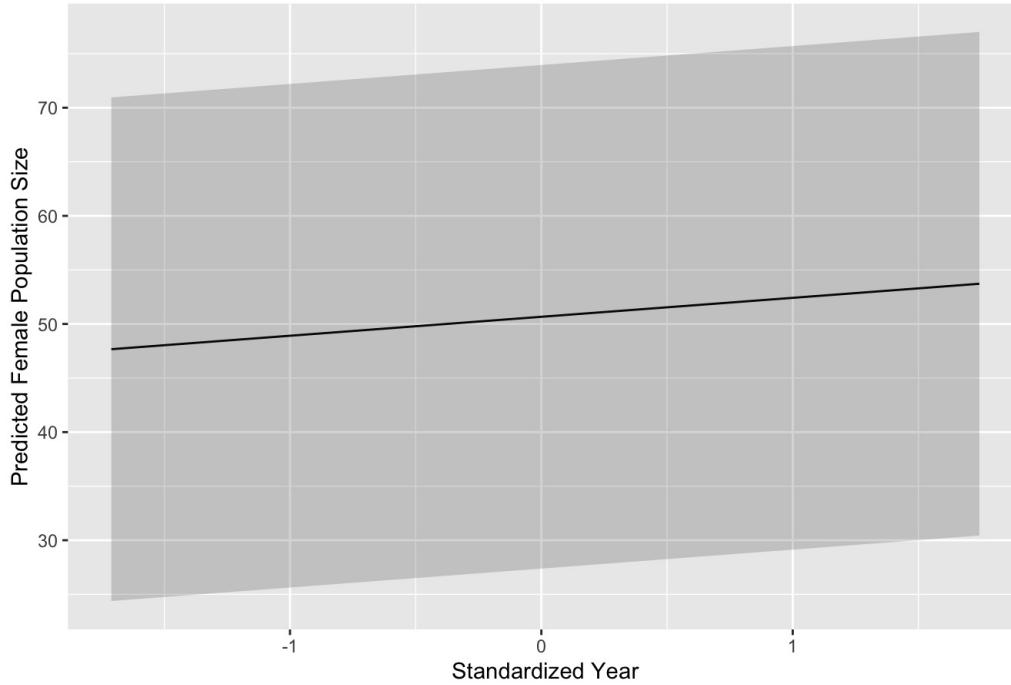
# Model predictions
newdata <- expand.grid(
  year_scaled = seq(min(data$year_scaled),
                     max(data$year_scaled),
                     length.out = 100),
  x_scaled = mean(data$x_scaled),
  y_scaled = mean(data$y_scaled)
)

newdata$pred <- predict(best_model, newdata, re.form = NA)
pred_plot <- ggplot(newdata, aes(x = year_scaled, y = pred)) +
  geom_line() +
  geom_ribbon(aes(ymin = pred - 1.96*sigma(best_model),
                  ymax = pred + 1.96*sigma(best_model)),
              alpha = 0.2) +
  labs(title = "Population Size Predictions over Time",
       x = "Standardized Year",
       y = "Predicted Female Population Size")

print(pred_plot)

```

Population Size Predictions over Time



Check if x,y are same for one band

```

library(dplyr)

band_consistency <- data %>%
  group_by(band) %>%
  summarize(
    x_diff_numbers = n_distinct(x, na.rm = TRUE),
    y_diff_numbers = n_distinct(y, na.rm = TRUE),
    years_observed = n_distinct(year)
  )

inconsistent_bands <- band_consistency %>%
  filter(x_diff_numbers > 1 | y_diff_numbers > 1)

print(inconsistent_bands)

```

```

## # A tibble: 190 × 4
##   band x_diff_numbers y_diff_numbers years_observed
##   <int>     <int>     <int>           <int>
## 1 2034         2         2             2
## 2 2071         3         3             3
## 3 2073         3         3             3
## 4 2207         2         2             2
## 5 2267         2         1             2
## 6 2268         2         2             2
## 7 2272         2         2             2
## 8 2314         2         2             2
## 9 2344         2         2             2
## 10 2353        2         2             2
## # i 180 more rows

```

```

# Extract inconsistent rows
inconsistent_rows <- data %>%
  filter(band %in% inconsistent_bands$band)

num_inconsistent_rows <- nrow(inconsistent_rows)

cat("Number of inconsistent rows(of total 742 rows):", num_inconsistent_rows, "\n")

```

```

## Number of inconsistent rows(of total 742 rows): 569

```

```

unique_band_count <- data %>%
  summarize(unique_bands = n_distinct(band))

cat("Number of unique bands in the dataset(has 190 unique inconsistant band):", unique_band_count$unique_bands, "\n")

```

```

## Number of unique bands in the dataset(has 190 unique inconsistant band): 356

```

heterogeneity

```

library(ggplot2)
library(dplyr)
library(gridExtra)

```

```

##
## Attaching package: 'gridExtra'

```

```

## The following object is masked from 'package:dplyr':
## 
##     combine

```

```

library(tidyr)

```

```

##
## Attaching package: 'tidyr'

```

```

## The following objects are masked from 'package:Matrix':
## 
##     expand, pack, unpack

```

```

library(lme4)

# Summary by year and bird
year_bird_summary <- data %>%
  group_by(year) %>%
  summarise(
    n_observations = n(),
    n_unique_birds = n_distinct(band),
    mean_fpop = mean(fpop, na.rm = TRUE),
    sd_fpop = sd(fpop, na.rm = TRUE),
    mean_spf = mean(spf, na.rm = TRUE),
    sd_spf = sd(spf, na.rm = TRUE)
  )

bird_summary <- data %>%
  group_by(band) %>%
  summarise(
    n_years = n_distinct(year),
    mean_spf = mean(spf),
    sd_spf = sd(spf, na.rm = TRUE),
    mean_fpop = mean(fpop),
    sd_fpop = sd(fpop, na.rm = TRUE)
  )

# Mixed model
temporal_mixed <- lmer(spf ~ factor(year) + (1|band), data = data)
cat("\nTemporal Heterogeneity (Mixed Model):\n")

```

```

##  
## Temporal Heterogeneity (Mixed Model):

```

```
print(anova(temporal_mixed))
```

```

## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## factor(year) 924.38  54.375     17   630.53  14.881 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

data$x_quadrant <- cut(data$x, breaks = 4, labels = c("West", "Central-West", "Central-East", "East"))
data$y_quadrant <- cut(data$y, breaks = 4, labels = c("South", "Central-South", "Central-North", "North"))

spatial_summary <- data %>%
  group_by(x_quadrant, y_quadrant) %>%
  summarise(
    mean_spf = mean(spf),
    sd_spf = sd(spf),
    n_birds = n_distinct(band),
    n = n(),
    .groups = 'drop'
  )

bird_movement <- data %>%
  group_by(band) %>%
  summarise(
    n_locations = n_distinct(paste(x, y)),
    mean_x = mean(x),
    mean_y = mean(y),
    x_range = max(x) - min(x),
    y_range = max(y) - min(y)
  )

density_mixed <- lmer(spf ~ fpop + (1|band), data = data)
cat("\nDensity Dependence Analysis (Mixed Model):\n")

```

```

##  
## Density Dependence Analysis (Mixed Model):

```

```
print(summary(density_mixed))
```

```

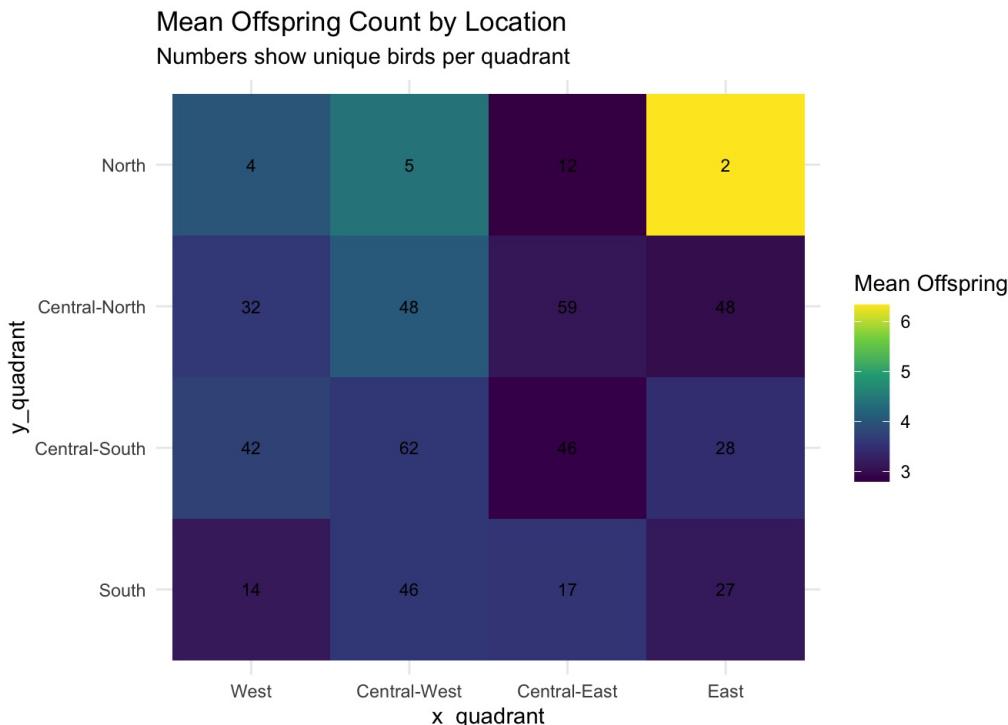
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: spf ~ fpop + (1 | band)
##   Data: data
##
## REML criterion at convergence: 3280
##
## Scaled residuals:
##   Min     1Q Median     3Q    Max
## -2.3333 -0.6891 -0.0741  0.6477  3.4384
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   band     (Intercept) 1.108    1.053
##   Residual           4.056    2.014
## Number of obs: 736, groups: band, 356
##
## Fixed effects:
##   Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 5.682614  0.301515 692.050890 18.847 < 2e-16 ***
## fpop       -0.046445  0.005802 713.953776 -8.005 4.85e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr) fpop 
## fpop -0.947

```

```

# visualization
p5 <- ggplot(spatial_summary,
              aes(x = x_quadrant, y = y_quadrant)) +
  geom_tile(aes(fill = mean_spf)) +
  geom_text(aes(label = n_birds), size = 3) +
  scale_fill_viridis_c() +
  labs(title = "Mean Offspring Count by Location",
       subtitle = "Numbers show unique birds per quadrant",
       fill = "Mean Offspring") +
  theme_minimal()
print(p5)

```



```
cat("\nIndividual Bird Movement Patterns:\n")
```

```

## 
## Individual Bird Movement Patterns:

```

```
print(summary(bird_movement))
```

```

##      band    n_locations     mean_x     mean_y
##  Min.   : 2009   Min.   :1.000   Min.   : 0.8611   Min.   :1.625
##  1st Qu.:30612   1st Qu.:1.000   1st Qu.:10.2500  1st Qu.:2.495
##  Median :56907   Median :2.000   Median :16.3125  Median :3.208
##  Mean   :44624   Mean   :2.056   Mean   :16.7257  Mean   :3.068
##  3rd Qu.:59457   3rd Qu.:3.000   3rd Qu.:23.7500  3rd Qu.:3.547
##  Max.   :86268   Max.   :7.000   Max.   :32.8333  Max.   :5.000
##      x_range     y_range
##  Min.   : 0.00000   Min.   :0.00000
##  1st Qu.: 0.00000   1st Qu.:0.00000
##  Median : 0.06042   Median :0.03333
##  Mean   : 0.70199   Mean   :0.30834
##  3rd Qu.: 0.75000   3rd Qu.:0.46979
##  Max.   :17.79167   Max.   :3.00000

```

```

consistency_test <- data %>%
  filter(n() >= 2) %>%
  group_by(band) %>%
  summarise(
    spf_consistency = sd(spf),
    location_consistency = mean(sqrt((x - mean(x))^2 + (y - mean(y))^2))
  )
cat("\nIndividual Consistency Summary:\n")

```

```

## 
## Individual Consistency Summary:

```

```

print(summary(consistency_test))

```

```

##      band    spf_consistency location_consistency
##  Min.   : 2009   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:30612   1st Qu.:0.9787   1st Qu.:0.00000
##  Median :56907   Median :1.5275   Median :0.07933
##  Mean   :44624   Mean   :1.7347   Mean   :0.34496
##  3rd Qu.:59457   3rd Qu.:2.3094   3rd Qu.:0.44390
##  Max.   :86268   Max.   :4.9497   Max.   :8.89703
##                NA's   :165

```

Data description

```

library(lme4)
library(dplyr)
library(ggplot2)
library(car)
library(gridExtra)
library(tidyr)
library(performance)

# Data Preparation
data$x_region <- cut(data$x, breaks = 3, labels = c("West", "Central", "East"))
data$y_region <- cut(data$y, breaks = 3, labels = c("South", "Central", "North"))
data$fpop_cat <- cut(data$fpop,
                      breaks = quantile(data$fpop, probs = c(0, 0.33, 0.67, 1)),
                      labels = c("Low", "Medium", "High"),
                      include.lowest = TRUE)
data$band <- factor(data$band)

# summary
summarize_groups <- function(data, group_var) {
  data %>%
    group_by (!!sym(group_var)) %>%
    summarise(
      n = n(),
      n_birds = n_distinct(band),
      mean_spf = round(mean(spf, na.rm = TRUE), 3),
      sd_spf = round(sd(spf, na.rm = TRUE), 3),
      se_spf = round(sd_spf/sqrt(n), 3),
      min_spf = min(spf, na.rm = TRUE),
      max_spf = max(spf, na.rm = TRUE)
    ) %>%
    arrange (!!sym(group_var))
}

year_summary <- summarize_groups(data, "year")
x_region_summary <- summarize_groups(data, "x_region")
y_region_summary <- summarize_groups(data, "y_region")
density_summary <- summarize_groups(data, "fpop_cat")

bird_summary <- data %>%
  group_by(band) %>%
  summarise(
    n_obs = n(),
    mean_spf = round(mean(spf, na.rm = TRUE), 3),
    sd_spf = round(sd(spf, na.rm = TRUE), 3),
    n_years = n_distinct(year),
    n_regions = n_distinct(paste(x_region, y_region))
  ) %>%
  arrange(desc(mean_spf))

# Mixed Effects Models
temp_model <- lmer(spf ~ as.factor(year) + (1|band), data = data)
spatial_model <- lmer(spf ~ x_region + y_region + (1|band), data = data)
density_model <- lmer(spf ~ fpop_cat + (1|band), data = data)
combined_model <- lmer(spf ~ as.factor(year) + x_region + y_region +
                        fpop_cat + (1|band), data = data)

```

```
## fixed-effect model matrix is rank deficient so dropping 2 columns / coefficients
```

```

# Plot
plot_summary <- function(summary_data, x_var, title) {
  ggplot(summary_data, aes(x = !!sym(x_var), y = mean_spf)) +
    geom_point() +
    geom_errorbar(aes(ymin = mean_spf - se_spf, ymax = mean_spf + se_spf),
                  width = 0.2) +
    labs(title = title,
         x = x_var,
         y = "Mean Offspring Count") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

p1 <- plot_summary(year_summary, "year", "Mean Offspring by Year")
p2 <- plot_summary(x_region_summary, "x_region", "Mean Offspring by X Region")
p3 <- plot_summary(y_region_summary, "y_region", "Mean Offspring by Y Region")
p4 <- plot_summary(density_summary, "fpop_cat", "Mean Offspring by Population Density")

```

```

p5 <- ggplot(bird_summary %>% filter(n_obs > 1),
              aes(x = n_obs, y = sd_spf)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = TRUE, color = "blue") +
  labs(title = "Bird Reproductive Consistency",
       x = "Number of Observations",
       y = "Standard Deviation in Offspring Count") +
  theme_minimal()

p6 <- ggplot(data, aes(x = year, y = spf, group = band)) +
  geom_line(alpha = 0.2) +
  geom_point(alpha = 0.3) +
  labs(title = "Individual Bird Trajectories",
       x = "Year",
       y = "Number of Offspring") +
  theme_minimal()

p7 <- ggplot(data, aes(x = x, y = y)) +
  geom_path(aes(group = band), alpha = 0.2) +
  geom_point(aes(color = spf)) +
  scale_color_viridis_c() +
  labs(title = "Bird Movement Patterns",
       x = "X Coordinate",
       y = "Y Coordinate",
       color = "Offspring Count") +
  theme_minimal()

p8 <- ggplot(bird_summary, aes(x = mean_spf)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
  labs(title = "Distribution of Mean Offspring per Bird",
       x = "Mean Offspring Count",
       y = "Number of Birds") +
  theme_minimal()

p9 <- ggplot(data, aes(x = fpop, y = spf)) +
  geom_point(aes(color = factor(band)), alpha = 0.5) +
  facet_wrap(~year, ncol = 5) +
  labs(title = "Density Dependence by Year",
       x = "Female Population Size",
       y = "Number of Offspring") +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.x = element_text(angle = 0),
    strip.text = element_text(size = 10),
    plot.title = element_text(size = 14),
    panel.spacing = unit(1, "lines"),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 12)
  ) +
  scale_x_continuous(
    breaks = seq(0, 80, by = 20),
    limits = c(0, 80)
  ) +
  scale_y_continuous(
    breaks = seq(0, 12, by = 3),
    limits = c(0, 12)
  )

# Results
cat("\nSummary Statistics by Year:\n")

```

```

##  
## Summary Statistics by Year:

```

```

print(year_summary)

```

```

## # A tibble: 18 × 8
##   year    n_n_birds mean_spf sd_spf se_spf min_spf max_spf
##   <int>     <int>     <dbl>   <dbl>   <dbl>   <int>   <int>
## 1     1       34      34     1.26    1.46   0.251     0     6
## 2     2       30      30     3.73    2.08   0.38      0     8
## 3     3       43      43     4.23    1.92   0.294      0     7
## 4     4       48      48     5.04    2.41   0.348      0     9
## 5     5       65      65     3.08    1.75   0.217      0     7
## 6     7       17      17     5.76    1.68   0.407      3     9
## 7     8       26      26     4.65    1.90   0.372      2     7
## 8     9       54      54     3.78    2.35   0.32      0     9
## 9    10      53      53     3.62    2.13   0.293      0     8
## 10   11      71      71     2.37    2.44   0.289      0    10
## 11   12      60      60     2       1.46   0.189      0     5
## 12   13      57      57     2.02    1.62   0.215      0     6
## 13   14      53      53     2.42    1.79   0.246      0     7
## 14   15       4       4     2.71    1.35      2     8
## 15   16      10      10     7       2.62   0.83      3    12
## 16   17      28      28     4.68    2.48   0.469      0    10
## 17   18      41      41     5.29    2.59   0.405      0    10
## 18   19      42      42     3.19    1.78   0.275      0     7

```

```
cat("\nSummary Statistics by X Region:\n")
```

```

## 
## Summary Statistics by X Region:
```

```
print(x_region_summary)
```

```

## # A tibble: 3 × 8
##   x_region    n_n_birds mean_spf sd_spf se_spf min_spf max_spf
##   <fct>     <int>     <dbl>   <dbl>   <dbl>   <int>   <int>
## 1 West       195      105     3.64    2.52   0.181     0    10
## 2 Central    309      160     3.45    2.31   0.132     0    12
## 3 East       232      114     3.10    2.31   0.151     0     9

```

```
cat("\nSummary Statistics by Y Region:\n")
```

```

## 
## Summary Statistics by Y Region:
```

```
print(y_region_summary)
```

```

## # A tibble: 3 × 8
##   y_region    n_n_birds mean_spf sd_spf se_spf min_spf max_spf
##   <fct>     <int>     <dbl>   <dbl>   <dbl>   <int>   <int>
## 1 South      238      142     3.20    2.29   0.148     0     9
## 2 Central    395      225     3.56    2.42   0.122     0    12
## 3 North      103       65     3.16    2.36   0.232     0     9

```

```
cat("\nSummary Statistics by Population Density:\n")
```

```

## 
## Summary Statistics by Population Density:
```

```
print(density_summary)
```

```

## # A tibble: 3 × 8
##   fpop_cat    n_n_birds mean_spf sd_spf se_spf min_spf max_spf
##   <fct>     <int>     <dbl>   <dbl>   <dbl>   <int>   <int>
## 1 Low        275      175     4.09    2.46   0.148     0    12
## 2 Medium     265      179     3.32    2.32   0.142     0     9
## 3 High       196      154     2.49    2.00   0.143     0    10

```

```
cat("\nTemporal Model Summary:\n")
```

```
##  
## Temporal Model Summary:
```

```
print(summary(temp_model))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: spf ~ as.factor(year) + (1 | band)  
##   Data: data  
##  
## REML criterion at convergence: 3119.6  
##  
## Scaled residuals:  
##     Min      1Q  Median      3Q     Max  
## -2.5469 -0.6539 -0.0488  0.6654  3.6381  
##  
## Random effects:  
##   Groups   Name        Variance Std.Dev.  
##   band     (Intercept) 0.5301   0.7281  
##   Residual            3.6540   1.9115  
## Number of obs: 736, groups: band, 356  
##  
## Fixed effects:  
##              Estimate Std. Error    df t value Pr(>|t|)  
## (Intercept)  1.2476    0.3490 716.9013  3.575 0.000374 ***  
## as.factor(year)2  2.4865    0.4883 507.9382  5.092 5.01e-07 ***  
## as.factor(year)3  2.9342    0.4554 629.7722  6.443 2.33e-10 ***  
## as.factor(year)4  3.6741    0.4510 700.9423  8.146 1.72e-15 ***  
## as.factor(year)5  1.7772    0.4291 717.4884  4.142 3.85e-05 ***  
## as.factor(year)7  4.5670    0.6030 717.2949  7.574 1.12e-13 ***  
## as.factor(year)8  3.4322    0.5283 715.2325  6.497 1.54e-10 ***  
## as.factor(year)9  2.5523    0.4453 717.4784  5.731 1.47e-08 ***  
## as.factor(year)10 2.3256    0.4465 717.9858  5.209 2.48e-07 ***  
## as.factor(year)11 1.0783    0.4242 717.6562  2.542 0.011226 *  
## as.factor(year)12 0.7967    0.4365 717.7773  1.825 0.068377 .  
## as.factor(year)13 0.8078    0.4407 717.7366  1.833 0.067190 .  
## as.factor(year)14 1.1937    0.4468 717.7404  2.672 0.007720 **  
## as.factor(year)15 2.7270    1.0717 714.5727  2.545 0.011151 *  
## as.factor(year)16 5.6009    0.7284 709.4026  7.690 4.93e-14 ***  
## as.factor(year)17 3.3349    0.5193 717.0705  6.422 2.45e-10 ***  
## as.factor(year)18 4.0174    0.4724 715.2232  8.504 < 2e-16 ***  
## as.factor(year)19 2.0120    0.4699 714.9779  4.282 2.10e-05 ***  
## ---  
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##  
## Correlation matrix not shown by default, as p = 18 > 12.  
## Use print(summary(temp_model), correlation=TRUE) or  
##   vcov(summary(temp_model))      if you need it
```

```
cat("\nSpatial Model Summary:\n")
```

```
##  
## Spatial Model Summary:
```

```
print(summary(spatial_model))
```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: spf ~ x_region + y_region + (1 | band)
##   Data: data
##
## REML criterion at convergence: 3329.2
##
## Scaled residuals:
##   Min     1Q Median     3Q    Max
## -2.07977 -0.67252 -0.07454  0.60702  3.00721
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   band     (Intercept) 1.397    1.182
##   Residual           4.241    2.059
## Number of obs: 736, groups: band, 356
##
## Fixed effects:
##   Estimate Std. Error      df t value Pr(>|t|)    
##   (Intercept)  3.4199    0.2328 410.8890 14.690 <2e-16 ***
##   x_regionCentral -0.2379   0.2453 399.0340 -0.970  0.3326
##   x_regionEast    -0.4759   0.2658 353.7425 -1.790  0.0742 .
##   y_regionCentral  0.3941   0.2068 624.9294  1.906  0.0572 .
##   y_regionNorth    0.0748   0.3006 608.9764  0.249  0.8036
##   ...
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr) x_rgnC x_rgnE y_rgnC
##   x_regnCntrl -0.616
##   x_regionEst -0.608  0.581
##   y_regnCntrl -0.538 -0.055  0.018
##   y_regnNrth -0.364 -0.027 -0.046  0.458

```

```
cat("\nDensity Model Summary:\n")
```

```
## 
## Density Model Summary:
```

```
print(summary(density_model))
```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: spf ~ fpop_cat + (1 | band)
##   Data: data
##
## REML criterion at convergence: 3289.5
##
## Scaled residuals:
##   Min     1Q Median     3Q    Max
## -2.2996 -0.6804 -0.0794  0.6495  3.3575
##
## Random effects:
##   Groups   Name        Variance Std.Dev.
##   band     (Intercept) 1.232    1.110
##   Residual           4.076    2.019
## Number of obs: 736, groups: band, 356
##
## Fixed effects:
##   Estimate Std. Error      df t value Pr(>|t|)    
##   (Intercept)  4.0140    0.1496 482.0346 26.825 < 2e-16 ***
##   fpop_catMedium -0.6524    0.2025 693.0534 -3.221  0.00134 **
##   fpop_catHigh   -1.5018    0.2191 697.7542 -6.853 1.59e-11 ***
##   ...
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr) fpp_cM
##   fpop_catMdm -0.680
##   fpop_catHgh -0.643  0.556

```

```
cat("\nCombined Model Summary:\n")
```

```
##  
## Combined Model Summary:
```

```
print(summary(combined_model))
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: spf ~ as.factor(year) + x_region + y_region + fpop_cat + (1 |  
## band)  
## Data: data  
##  
## REML criterion at convergence: 3115.7  
##  
## Scaled residuals:  
##     Min      1Q  Median      3Q     Max  
## -2.6212 -0.6573 -0.0442  0.6253  3.5044  
##  
## Random effects:  
## Groups   Name        Variance Std.Dev.  
## band     (Intercept) 0.4997   0.7069  
## Residual            3.6467   1.9096  
## Number of obs: 736, groups: band, 356  
##  
## Fixed effects:  
##              Estimate Std. Error    df t value Pr(>|t|)  
## (Intercept)  1.3767   0.3801 679.1217  3.622 0.000314 ***  
## as.factor(year)2  2.4459   0.4878 509.4878  5.014 7.37e-07 ***  
## as.factor(year)3  2.9641   0.4553 629.1518  6.510 1.54e-10 ***  
## as.factor(year)4  3.6950   0.4501 698.9601  8.209 1.07e-15 ***  
## as.factor(year)5  1.7790   0.4283 713.7327  4.154 3.66e-05 ***  
## as.factor(year)7  4.5957   0.6018 713.4215  7.636 7.21e-14 ***  
## as.factor(year)8  3.4247   0.5277 712.4082  6.489 1.61e-10 ***  
## as.factor(year)9  2.5592   0.4439 713.1728  5.765 1.22e-08 ***  
## as.factor(year)10 2.3510   0.4451 713.9993  5.282 1.70e-07 ***  
## as.factor(year)11 1.0964   0.4230 713.5755  2.592 0.009746 **  
## as.factor(year)12 0.8031   0.4357 713.5210  1.843 0.065712 .  
## as.factor(year)13 0.8255   0.4396 713.5659  1.878 0.060797 .  
## as.factor(year)14 1.1889   0.4461 713.2866  2.665 0.007875 **  
## as.factor(year)15 2.6487   1.0698 711.4455  2.476 0.013520 *  
## as.factor(year)16 5.5720   0.7270 705.7117  7.664 5.95e-14 ***  
## as.factor(year)17 3.3177   0.5177 712.8005  6.409 2.66e-10 ***  
## as.factor(year)18 4.0312   0.4721 710.8966  8.539 < 2e-16 ***  
## as.factor(year)19 2.0884   0.4692 711.5307  4.451 9.92e-06 ***  
## x_regionCentral -0.2198   0.2020 317.8099 -1.088 0.277357  
## x_regionEast    -0.4915   0.2164 274.7890 -2.271 0.023917 *  
## y_regionCentral 0.2378   0.1757 504.0588  1.353 0.176551  
## y_regionNorth   -0.1675   0.2553 466.1813 -0.656 0.512137  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##  
## Correlation matrix not shown by default, as p = 22 > 12.  
## Use print(summary(combined_model), correlation=TRUE) or  
## vcov(summary(combined_model)) if you need it
```

```
## fit warnings:  
## fixed-effect model matrix is rank deficient so dropping 2 columns / coefficients
```

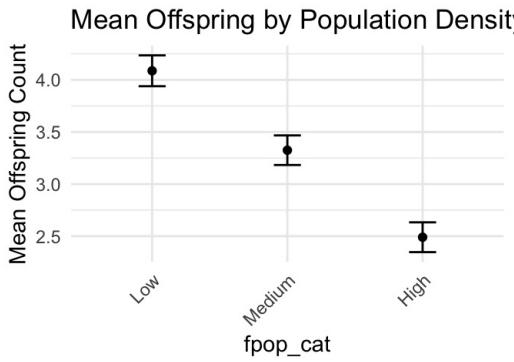
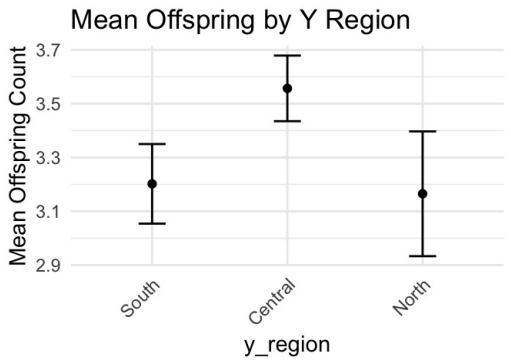
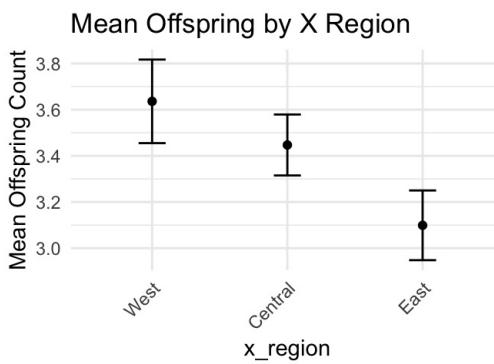
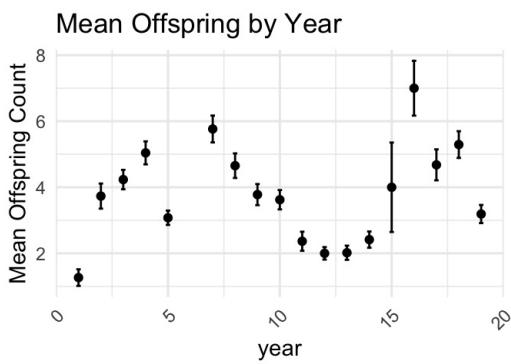
```
cat("\nIntraclass Correlation Coefficient:\n")
```

```
##  
## Intraclass Correlation Coefficient:
```

```
print(performance::icc(combined_model))
```

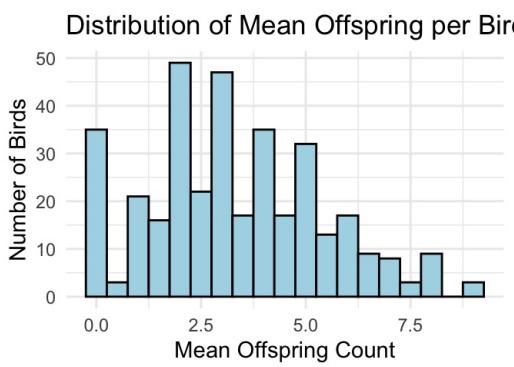
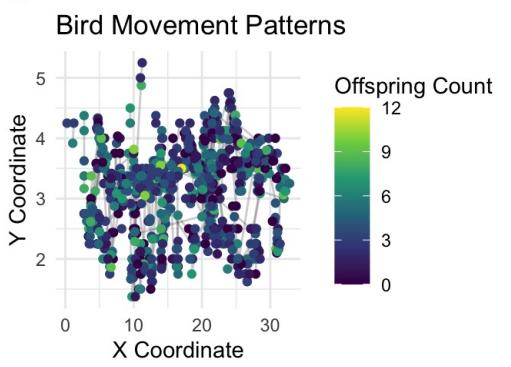
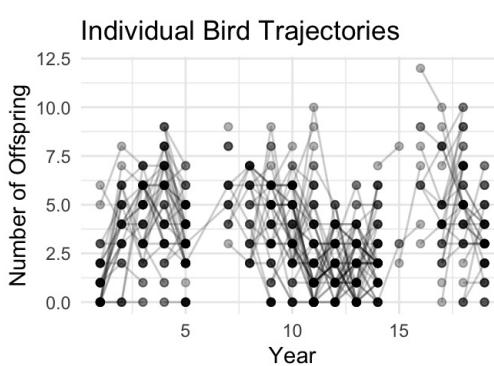
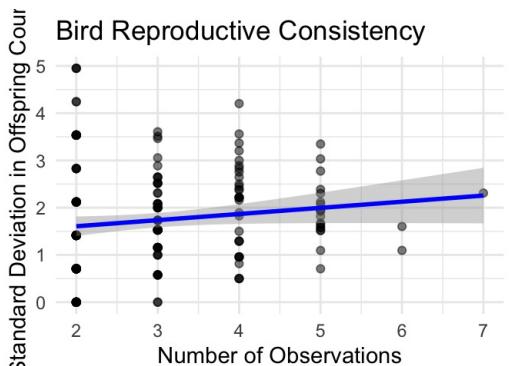
```
## # Intraclass Correlation Coefficient  
##  
##     Adjusted ICC: 0.121  
##     Unadjusted ICC: 0.087
```

```
grid.arrange(p1, p2, p3, p4, ncol = 2)
```



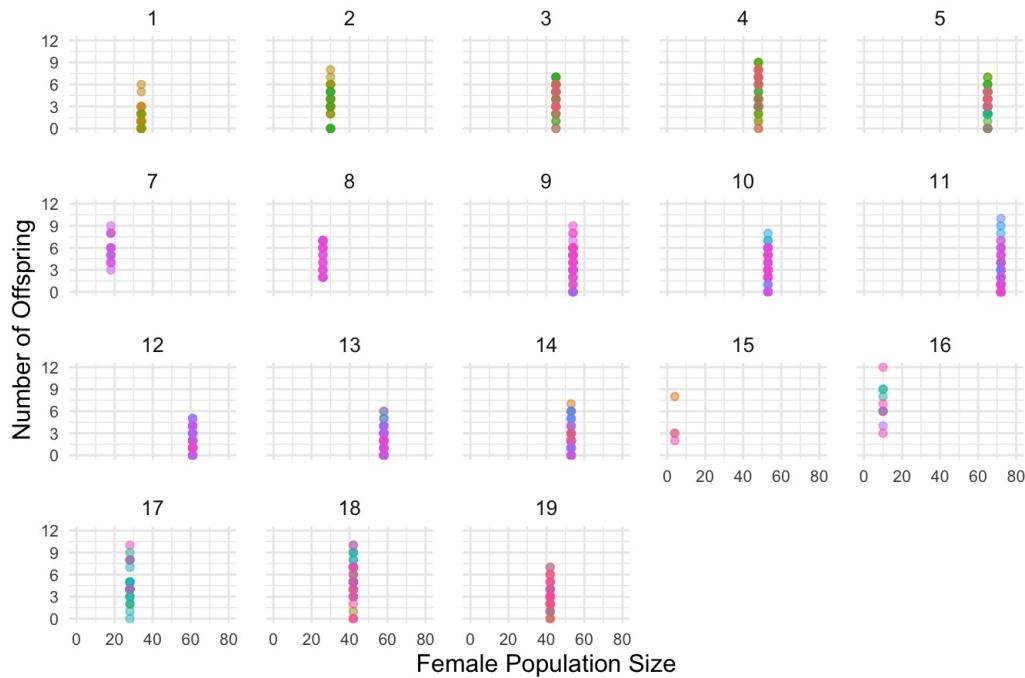
```
grid.arrange(p5, p6, p7, p8, ncol = 2)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

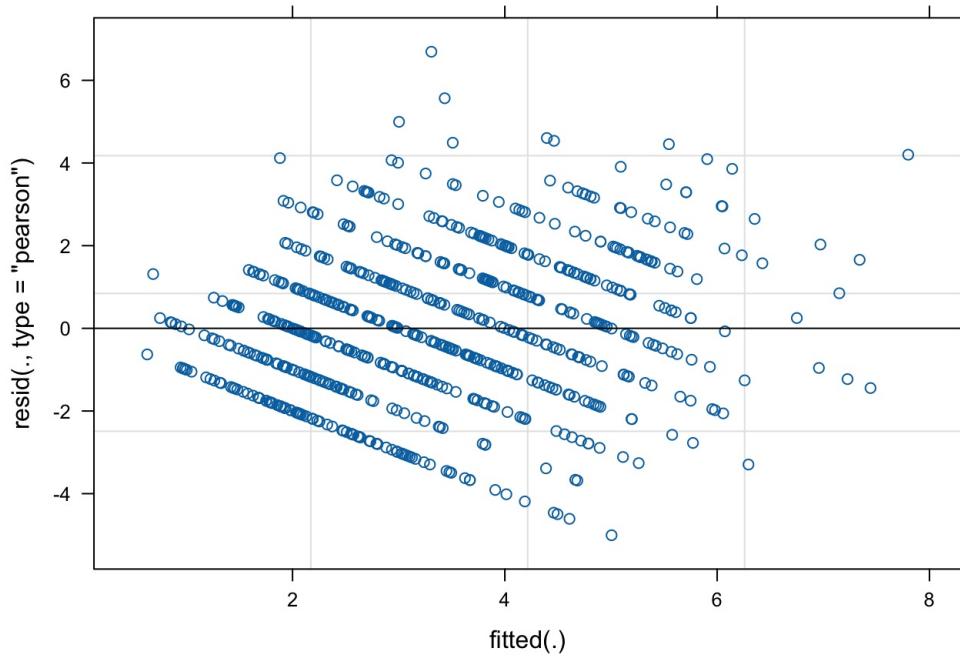


```
print(p9)
```

Density Dependence by Year



```
par(mfrow = c(2,2))
plot(combined_model)
```



```
par(mfrow = c(1,1))
```

Hierarchical model

```
library(lme4)
library(lmerTest)
library(MuMIn)
```

```
## Registered S3 method overwritten by 'MuMIn':
##   method      from
##   nobs.multinom broom
```

```

library(ggplot2)
library(dplyr)

data$x_scaled <- scale(data$x)
data$y_scaled <- scale(data$y)
data$fpop_scaled <- scale(data$fpop)

# Model 1
m1 <- lmer(spf ~ fpop_scaled + x_scaled + y_scaled +
            (1|year) + (1|band),
            data = data)

# Model 2
m2 <- lmer(spf ~ fpop_scaled + x_scaled + y_scaled +
            (1 + fpop_scaled|year) + (1|band),
            data = data)

## boundary (singular) fit: see help('isSingular')

# comparison
cat("\nModel Comparison:\n")

## 
## Model Comparison:

print(anova(m1, m2))

## refitting model(s) with ML (instead of REML)

## Data: data
## Models:
## m1: spf ~ fpop_scaled + x_scaled + y_scaled + (1 | year) + (1 | band)
## m2: spf ~ fpop_scaled + x_scaled + y_scaled + (1 + fpop_scaled | year) + (1 | band)
##   npar   AIC   BIC logLik deviance Chisq Df Pr(>Chisq)
## m1    7 3180.9 3213.1 -1583.5   3166.9
## m2    9 3181.0 3222.4 -1581.5   3163.0 3.8763  2      0.144

cat("\nFull Model Summary:\n")

## 
## Full Model Summary:

print(summary(m2))

```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method 
## lmerModLmerTest]
## Formula: spf ~ fpop_scaled + x_scaled + y_scaled + (1 + fpop_scaled | 
##   year) + (1 | band)
## Data: data
##
## REML criterion at convergence: 3172.4
##
## Scaled residuals:
##   Min     1Q Median     3Q    Max 
## -2.4802 -0.6516 -0.0795  0.6534  3.6235 
##
## Random effects:
##   Groups      Name        Variance Std.Dev. Corr
##   band        (Intercept) 0.5478   0.7401  
##   year        (Intercept) 0.7674   0.8760  
##             fpop_scaled 0.1598   0.3997  -1.00 
##   Residual      3.6106   1.9002  
## Number of obs: 736, groups: band, 356; year, 18
##
## Fixed effects:
##   Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)  3.36646   0.22421 17.11030 15.015 2.76e-11 ***
## fpop_scaled -0.70400   0.19778 11.91268 -3.560  0.00397 **  
## x_scaled    -0.22307   0.08305 279.07303 -2.686  0.00766 **  
## y_scaled     0.00604   0.08114 396.97854  0.074  0.94070  
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr) fpp_sc x_scld
## fpop_scaled -0.472
## x_scaled    0.007 -0.009
## y_scaled    0.007  0.004 -0.108
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

```

```

r2 <- r.squaredGLMM(m2)
cat("\nR2 values:\n")

```

```

## 
## R2 values:

```

```

cat("Marginal R2 (fixed effects):", round(r2[1,1], 3), "\n")

```

```

## Marginal R2 (fixed effects): 0.098

```

```

cat("Conditional R2 (fixed + random effects):", round(r2[1,2], 3), "\n")

```

```

## Conditional R2 (fixed + random effects): 0.359

```

```

# ICC
icc <- function(model) {
  vc <- VarCorr(model)
  var_year <- attr(vc$year, "stddev")[1]^2
  var_band <- attr(vc$band, "stddev")[1]^2
  var_residual <- attr(vc, "sc")^2

  icc_year <- var_year/(var_year + var_band + var_residual)
  icc_band <- var_band/(var_year + var_band + var_residual)

  return(list(year = icc_year, band = icc_band))
}

cat("\nIntraclass Correlation Coefficients:\n")

```

```

## 
## Intraclass Correlation Coefficients:

```

```

icc_values <- icc(m2)
cat("ICC for year:", round(icc_values$year, 3), "\n")

```

```
## ICC for year: 0.156
```

```
cat("ICC for band:", round(icc_values$band, 3), "\n")
```

```
## ICC for band: 0.111
```

```
# plots
p1 <- ggplot(data.frame(fitted = fitted(m2),
                        resid = residuals(m2)),
              aes(x = fitted, y = resid)) +
  geom_point(alpha = 0.5) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  labs(title = "Residuals vs Fitted",
       x = "Fitted values",
       y = "Residuals") +
  theme_minimal()

ranef_year <- data.frame(effects = ranef(m2)$year[,1],
                           type = "Year")
ranef_band <- data.frame(effects = ranef(m2)$band[,1],
                           type = "Band")
ranef_all <- rbind(ranef_year, ranef_band)

p2 <- ggplot(ranef_all, aes(sample = effects)) +
  geom_qq() +
  geom_qq_line(color = "red") +
  facet_wrap(~type) +
  labs(title = "Random Effects Q-Q Plots") +
  theme_minimal()

new_data <- expand.grid(
  fpop_scaled = seq(min(data$fpop_scaled),
                     max(data$fpop_scaled),
                     length.out = 50),
  x_scaled = mean(data$x_scaled),
  y_scaled = mean(data$y_scaled),
  year = unique(data$year)
)

new_data$pred <- predict(m2, new_data, re.form = ~(1 + fpop_scaled|year))

p3 <- ggplot(new_data, aes(x = fpop_scaled, y = pred, group = year)) +
  geom_line(alpha = 0.3) +
  geom_smooth(aes(group = 1), method = "lm", color = "red") +
  labs(title = "Predicted Offspring Count vs Female Population Size",
       subtitle = "Lines show year-specific predictions",
       x = "Standardized Female Population Size",
       y = "Predicted Offspring Count") +
  theme_minimal()

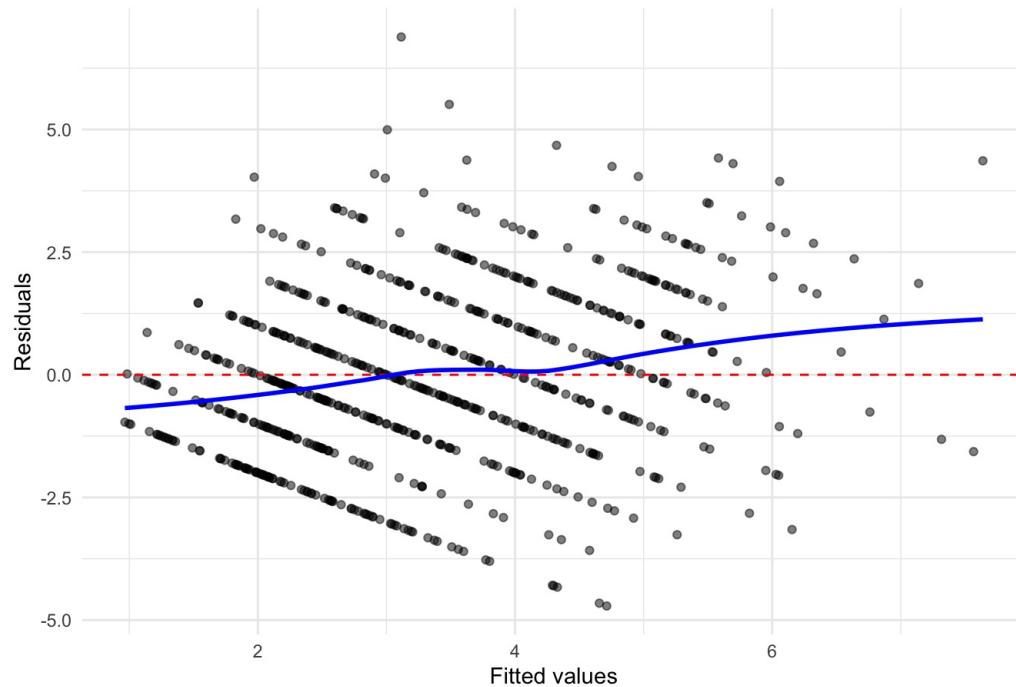
bird_effects <- data.frame(
  band = rownames(ranef(m2)$band),
  effect = ranef(m2)$band[,1]
)

p4 <- ggplot(bird_effects, aes(x = effect)) +
  geom_histogram(bins = 30, fill = "lightblue", color = "black") +
  labs(title = "Distribution of Bird Random Effects",
       x = "Random Effect Size",
       y = "Count") +
  theme_minimal()

print(p1)
```

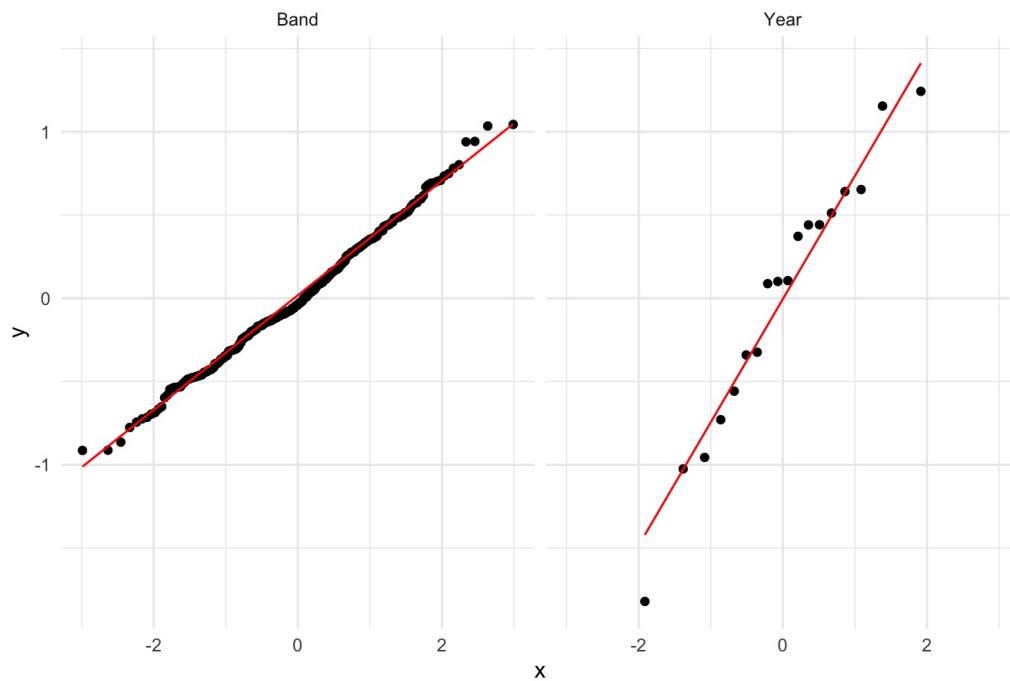
```
## `geom_smooth()` using formula = 'y ~ x'
```

Residuals vs Fitted



```
print(p2)
```

Random Effects Q-Q Plots

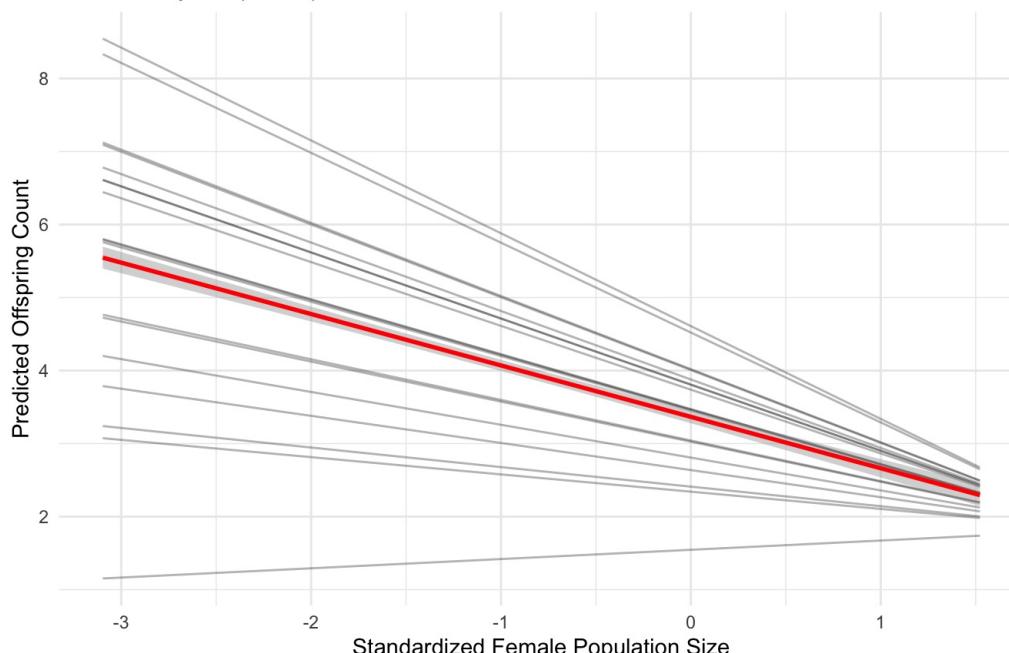


```
print(p3)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

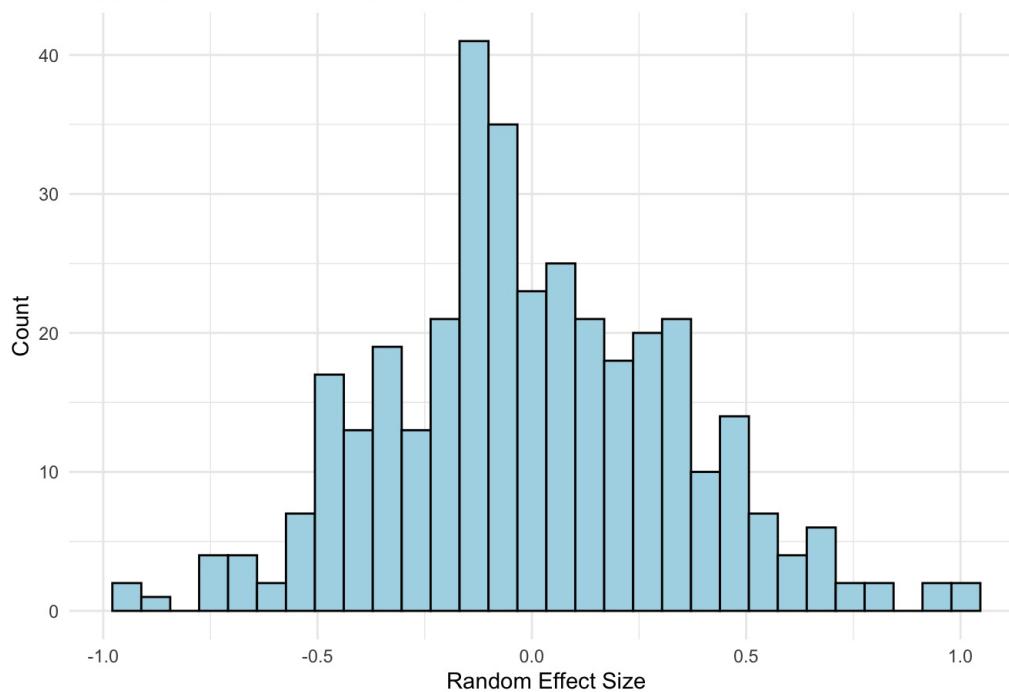
Predicted Offspring Count vs Female Population Size

Lines show year-specific predictions



```
print(p4)
```

Distribution of Bird Random Effects



```
cat("\nFixed Effects Summary:\n")
```

```
##  
## Fixed Effects Summary:
```

```
print(summary(m2)$coefficients)
```

```
##           Estimate Std. Error      df   t value  Pr(>|t|)  
## (Intercept) 3.366454859 0.22421085 17.11030 15.01468339 2.763769e-11  
## fpop_scaled -0.703999776 0.19777614 11.91268 -3.55957898 3.969766e-03  
## x_scaled    -0.223065103 0.08304731 279.07303 -2.68600049 7.664672e-03  
## y_scaled     0.006040394 0.08114149 396.97854  0.07444273 9.406956e-01
```

```
cat("\nRandom Effects Summary:\n")
```

```
##  
## Random Effects Summary:
```

```
print(VarCorr(m2))
```

```
## Groups     Name      Std.Dev. Corr
## band       (Intercept) 0.74015
## year        (Intercept) 0.87604
##             fpop_scaled 0.39972 -1.000
## Residual           1.90016
```

```
# Year-specific summaries
year_effects <- data.frame(
  year = rownames(ranef(m2)$year),
  intercept = ranef(m2)$year[,1],
  slope = ranef(m2)$year[,2]
)
cat("\nYear-specific Random Effects (top 5):\n")
```

```
## 
## Year-specific Random Effects (top 5):
```

```
print(head(year_effects, 5))
```

```
##   year intercept      slope
## 1    1 -1.8206442  0.8307231
## 2    2 -0.3404652  0.1553474
## 3    3  0.4403946 -0.2009431
## 4    4  1.2427926 -0.5670611
## 5    5  0.5118545 -0.2335489
```

```
cat("\nBird-specific Random Effects (top 5):\n")
```

```
## 
## Bird-specific Random Effects (top 5):
```

```
print(head(bird_effects, 5))
```

```
##   band      effect
## 1 2009 -0.07103456
## 2 2020 -0.02845905
## 3 2026 -0.16293832
## 4 2034 -0.18473288
## 5 2071 -0.10737566
```

```
# validation
cat("\nModel Validation:\n")
```

```
## 
## Model Validation:
```

```
cat("Convergence:", !isSingular(m2), "\n")
```

```
## Convergence: FALSE
```

```
cat("Number of observations:", nobs(m2), "\n")
```

```
## Number of observations: 736
```

```
cat("Number of groups - Year:", ngrps(m2)[ "year"], "\n")
```

```
## Number of groups - Year: 18
```

```
cat("Number of groups - Band:", ngrps(m2)[ "band"], "\n")
```

```
## Number of groups - Band: 356
```

other models

```
library(nlme)

## 
## Attaching package: 'nlme'

## The following object is masked from 'package:lme4':
## 
##     lmList

## The following object is masked from 'package:dplyr':
## 
##     collapse

library(lme4)
library(gLmmTMB)
library(mgcv)

## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.

library(MASS)

## 
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
## 
##     select

extract_fit_stats <- function(model) {
  aic <- AIC(model)
  bic <- BIC(model)
  loglik <- as.numeric(logLik(model))
  df <- attr(logLik(model), "df")

  data.frame(
    AIC = aic,
    BIC = bic,
    logLik = loglik,
    df = df
  )
}

# GLMM with Negative Binomial
nb_model <- glmer.nb(spf ~ fpop_scaled + x_scaled + y_scaled +
  (1|year) + (1|band),
  data = data)

## Warning in theta.ml(Y, mu, weights = object@resp$weights, limit = limit, :
## iteration limit reached

# GAM
gam_model <- gam(spf ~ s(fpop_scaled) +
  s(x_scaled, y_scaled) +
  s(year, bs = "re") +
  s(band, bs = "re"),
  family = nb(),
  data = data)

# Spatial cluster model
data$spatial_cluster <- paste0(
  cut(data$x_scaled, breaks = 5),
  "_",
  cut(data$y_scaled, breaks = 5)
)

spatial_model <- glmer.nb(spf ~ fpop_scaled + x_scaled + y_scaled +
  (1|year) + (1|band) + (1|spatial_cluster),
  data = data)
```

```
## Warning in theta.ml(Y, mu, weights = object@resp$weights, limit = limit, :  
## iteration limit reached
```

```
# Compare  
model_stats <- list(  
  "Negative Binomial GLMM" = extract_fit_stats(nb_model),  
  "GAM with Spatial Smooth" = extract_fit_stats(gam_model),  
  "Spatial Cluster Model" = extract_fit_stats(spatial_model)  
)  
  
model_comparison <- do.call(rbind, model_stats)  
model_comparison$Model <- rownames(model_comparison)  
rownames(model_comparison) <- NULL  
  
# AIC  
model_comparison$deltaAIC <- model_comparison$AIC - min(model_comparison$AIC)  
model_comparison <- model_comparison[, c("Model", "AIC", "deltaAIC", "BIC", "logLik", "df")]  
cat("\nModel Comparison:\n")
```

```
##  
## Model Comparison:
```

```
print(model_comparison, row.names = FALSE)
```

```
##           Model     AIC   deltaAIC     BIC   logLik     df  
##  Negative Binomial GLMM 3131.344  0.00000000 3163.552 -1558.672  7.00000  
##  GAM with Spatial Smooth 3150.364 19.01988702 3527.413 -1493.237 81.94535  
##  Spatial Cluster Model 3131.373  0.02919054 3168.183 -1557.687  8.00000
```

```
# best model  
best_model_name <- model_comparison$Model[which.min(model_comparison$AIC)]  
cat("\nBest model based on AIC:", best_model_name, "\n")
```

```
##  
## Best model based on AIC: Negative Binomial GLMM
```

```
best_model <- switch(best_model_name,  
  "Negative Binomial GLMM" = nb_model,  
  "GAM with Spatial Smooth" = gam_model,  
  "Spatial Cluster Model" = spatial_model  
)  
  
cat("\nBest Model Summary:\n")
```

```
##  
## Best Model Summary:
```

```
print(summary(best_model))
```

```

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(255.8929)  ( log )
## Formula: spf ~ fpop_scaled + x_scaled + y_scaled + (1 | year) + (1 | band)
## Data: data
##
##      AIC      BIC  logLik deviance df.resid
##  3131.3  3163.6 -1558.7   3117.3     729
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.0792 -0.7149 -0.0726  0.6533  3.9292
##
## Random effects:
## Groups Name        Variance Std.Dev.
## band   (Intercept) 0.05286  0.2299
## year   (Intercept) 0.10120  0.3181
## Number of obs: 736, groups: band, 356; year, 18
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.120713  0.086568 12.946 < 2e-16 ***
## fpop_scaled -0.182911  0.065652 -2.786 0.00534 **
## x_scaled    -0.066620  0.024484 -2.721 0.00651 **
## y_scaled     0.002959  0.024103  0.123 0.90230
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) fpp_sc x_scl
## fpop_scaled  0.365
## x_scaled     0.021 -0.004
## y_scaled     0.001 -0.010 -0.098

```

```

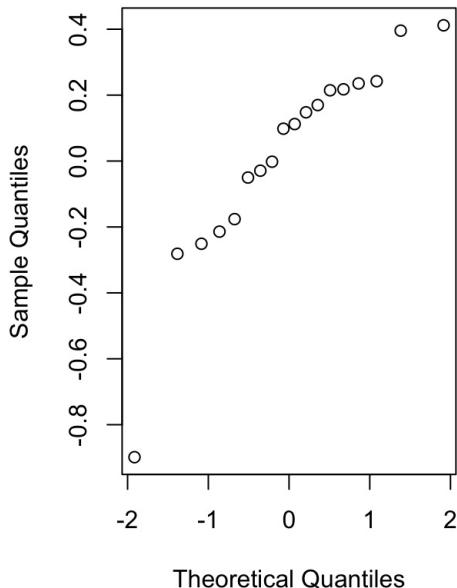
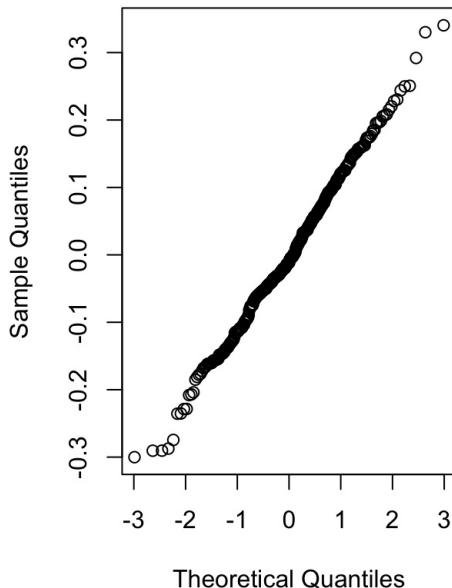
if(inherits(best_model, "glmerMod")) {
  par(mfrow = c(2,2))
  plot(best_model)
  par(mfrow = c(1,1))

  ranef_year <- ranef(best_model)$year
  ranef_band <- ranef(best_model)$band

  par(mfrow = c(1,2))
  qqnorm(ranef_year[[1]], main = "Year Random Effects")
  qqnorm(ranef_band[[1]], main = "Bird Random Effects")
  par(mfrow = c(1,1))

} else if(inherits(best_model, "gam")) {
  gam.check(best_model)
  plot(best_model, pages = 1)
}

```

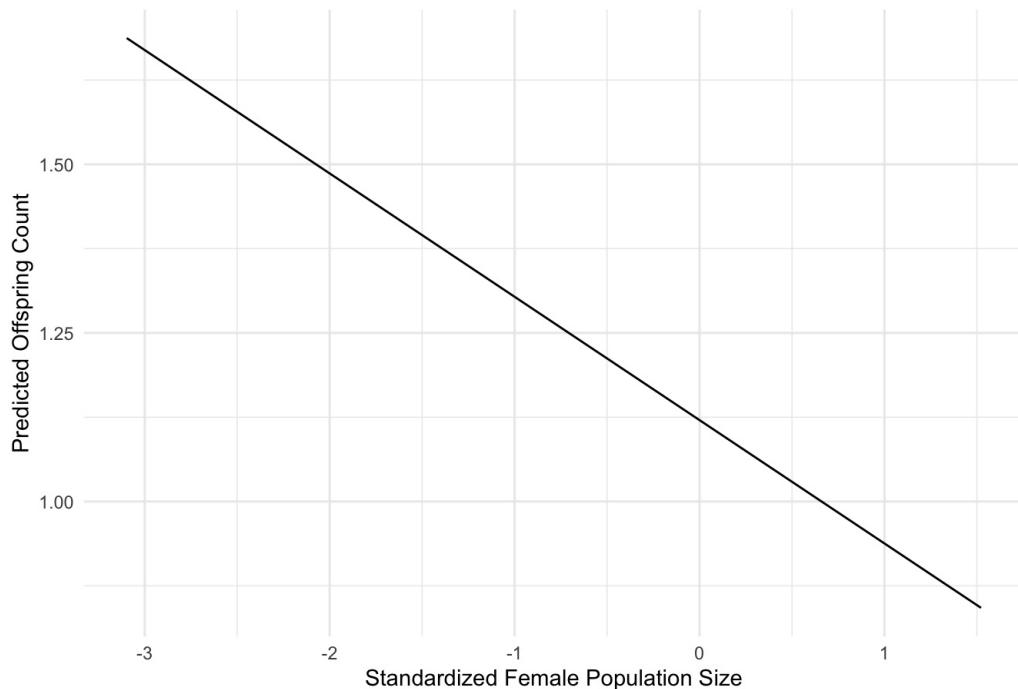
Year Random Effects**Bird Random Effects**

```
# predictions
newdata <- expand.grid(
  fpop_scaled = seq(min(data$fpop_scaled), max(data$fpop_scaled), length.out = 100),
  x_scaled = mean(data$x_scaled),
  y_scaled = mean(data$y_scaled)
)

if(inherits(best_model, "glmerMod")) {
  newdata$pred <- predict(best_model, newdata, re.form = NA)
} else if(inherits(best_model, "gam")) {
  newdata$pred <- predict(best_model, newdata, type = "response")
}

# plot
pred_plot <- ggplot(newdata, aes(x = fpop_scaled, y = pred)) +
  geom_line() +
  labs(title = "Model Predictions",
       x = "Standardized Female Population Size",
       y = "Predicted Offspring Count") +
  theme_minimal()

print(pred_plot)
```

Model Predictions

```

if(inherits(best_model, "glmerMod")) {
  cat("\nRandom Effects Variances:\n")
  print(VarCorr(best_model))

  vc <- VarCorr(best_model)
  total_var <- sum(sapply(vc, function(x) attr(x, "stddev")^2)) +
    attr(vc, "sc")^2

  cat("\nIntraclass Correlation Coefficients:\n")
  for(re in names(vc)) {
    icc <- attr(vc[[re]], "stddev")^2 / total_var
    cat(re, "ICC:", round(icc, 3), "\n")
  }
}

```

```

## 
## Random Effects Variances:
##   Groups Name      Std.Dev.
##   band   (Intercept) 0.22992
##   year   (Intercept) 0.31811
##
## Intraclass Correlation Coefficients:
##   band ICC: 0.046
##   year ICC: 0.088

```

Model and data analysis interpretation

```
library(DHARMa)
```

```
## This is DHARMa 0.4.7. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa')
```

```

library(mgcv)
library(ggplot2)
library(performance)
library(ade4)
library(spdep)

```

```
## Loading required package: spData
```

```

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`

```

```
## Loading required package: sf
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```

## 
## Attaching package: 'spdep'

## The following object is masked from 'package:ade4':
## 
##     mstree

```

```

check_spatial_autocorrelation <- function(model) {
  resids <- residuals(model)
  coords <- cbind(data$x_scaled, data$y_scaled)
  dist_matrix <- dist(coords)
  resid_dist <- dist(matrix(resids, ncol=1))
  mantel_result <- mantel.rtest(dist_matrix, resid_dist, nrepet = 999)
  plot(as.vector(dist_matrix),
       as.vector(resid_dist),
       xlab = "Spatial Distance",
       ylab = "Residual Distance",
       main = "Spatial Correlation in Residuals")
  abline(lm(as.vector(resid_dist) ~ as.vector(dist_matrix)),
         col = "red")
  return(mantel_result)
}

cat("\nChecking Assumptions for Best Model:\n")

```

```

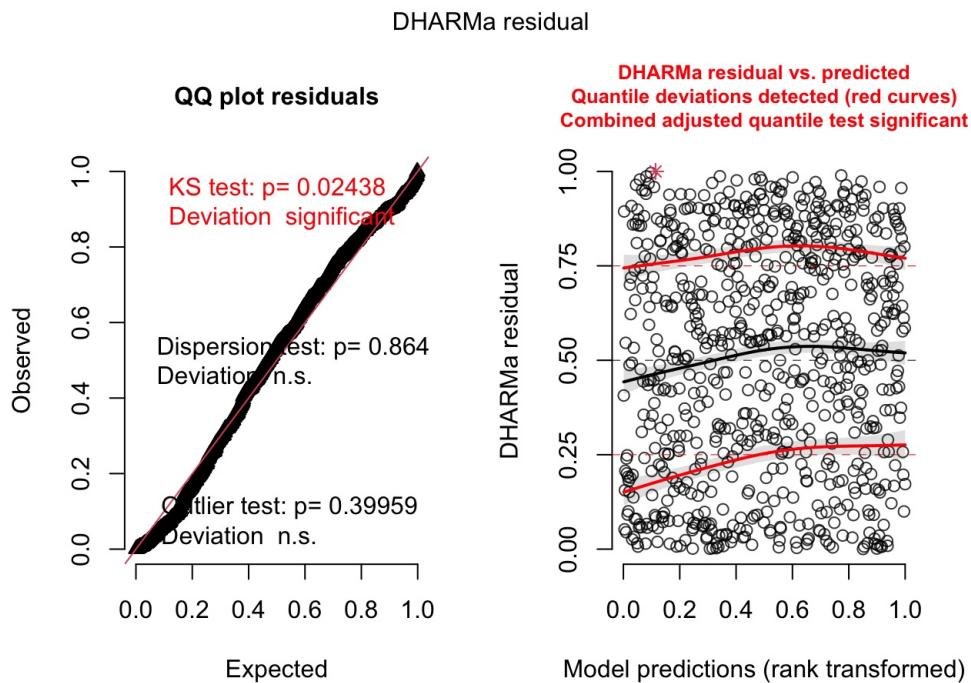
##  
## Checking Assumptions for Best Model:

```

```

# assumptions using DHARMA  
sim_resid <- simulateResiduals(best_model)  
plot(sim_resid)

```



```
cat ("\nDHARMA Tests:\n")
```

```

##  
## DHARMA Tests:

```

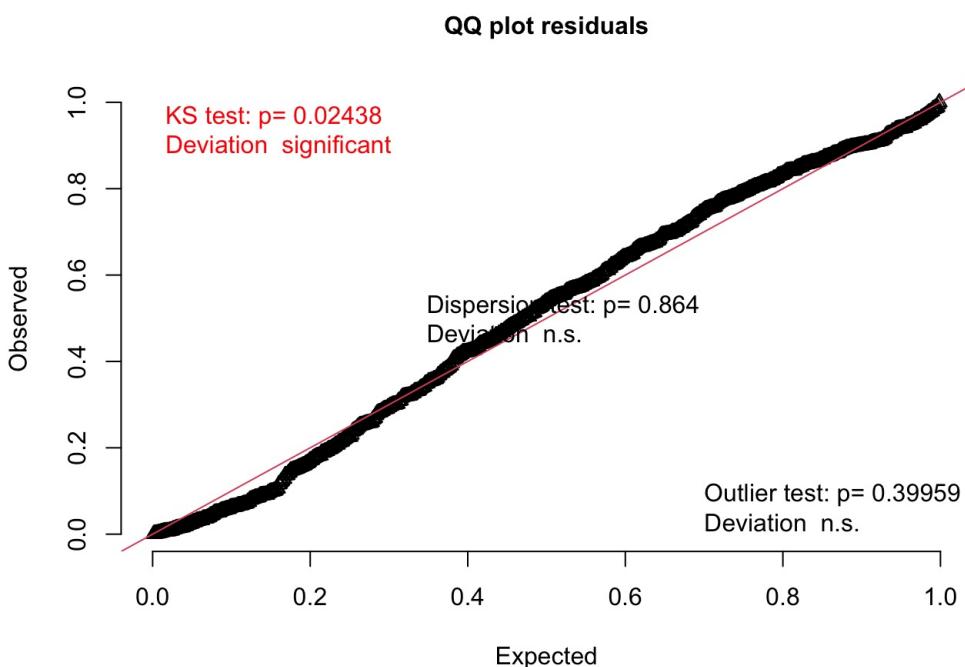
```
cat ("\n1. Uniformity Test:\n")
```

```

##  
## 1. Uniformity Test:

```

```
print(testUniformity(sim_resid))
```



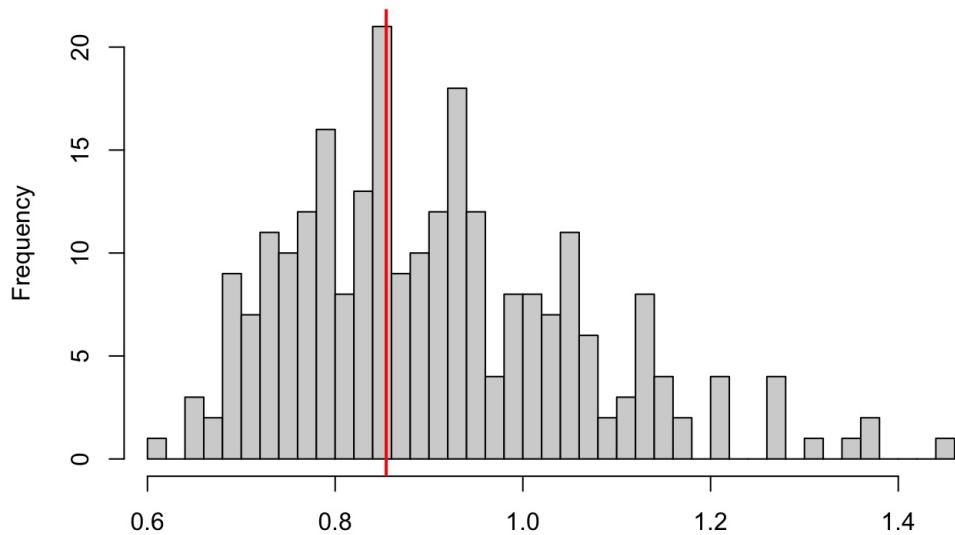
```
##  
##  Asymptotic one-sample Kolmogorov-Smirnov test  
##  
## data: simulationOutput$scaledResiduals  
## D = 0.054716, p-value = 0.02438  
## alternative hypothesis: two-sided
```

```
cat("\n2. Dispersion Test:\n")
```

```
##  
## 2. Dispersion Test:
```

```
print(testDispersion(sim_resid))
```

DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



```

##  

## DHARMA nonparametric dispersion test via sd of residuals fitted vs.  

## simulated  

##  

## data: simulationOutput  

## dispersion = 0.94215, p-value = 0.864  

## alternative hypothesis: two.sided

```

```
cat("\n3. Zero-inflation Test:\n")
```

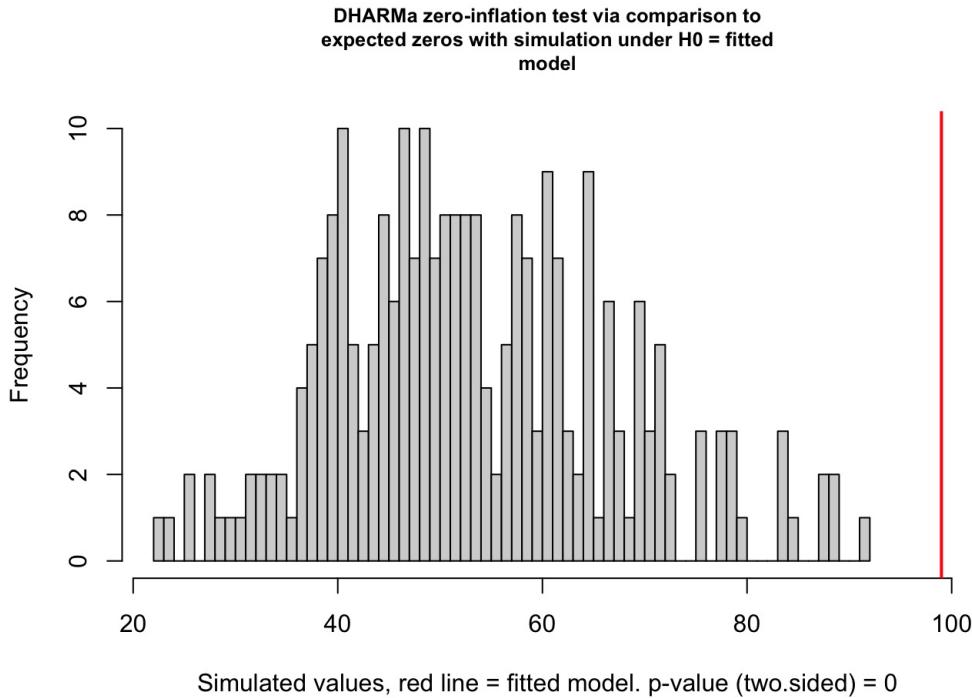
```

##  

## 3. Zero-inflation Test:

```

```
print(testZeroInflation(sim_resid))
```



```

##  

## DHARMA zero-inflation test via comparison to expected zeros with  

## simulation under H0 = fitted model  

##  

## data: simulationOutput  

## ratioObsSim = 1.8455, p-value < 2.2e-16  

## alternative hypothesis: two.sided

```

```
# spatial autocorrelation  
cat("\nSpatial Autocorrelation Check:\n")
```

```

##  

## Spatial Autocorrelation Check:

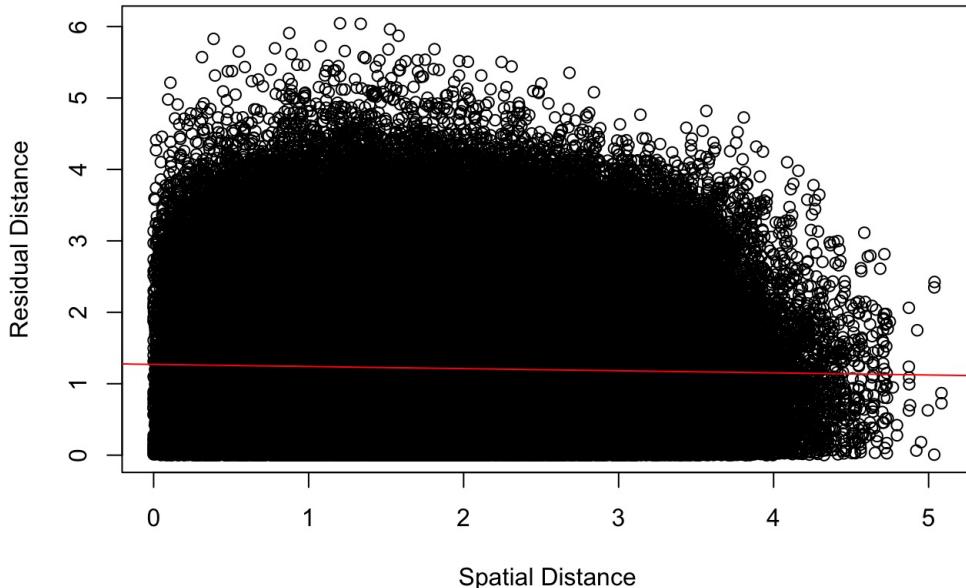
```

```
spatial_autocorr <- check_spatial_autocorrelation(best_model)
```

```
## Warning in is.euclid(m1): Zero distance(s)
```

```
## Warning in is.euclid(distmat): Zero distance(s)
```

Spatial Correlation in Residuals



```
print(spatial_autocorr)
```

```
## Monte-Carlo test
## Call: mantel.rtest(m1 = dist_matrix, m2 = resid_dist, nrep = 999)
##
## Observation: -0.02840851
##
## Based on 999 replicates
## Simulated p-value: 0.988
## Alternative hypothesis: greater
##
##      Std.Obs    Expectation     Variance
## -2.267808e+00 -8.165783e-05  1.560213e-04
```

```
# overdispersion
overdisp <- check_overdispersion(best_model)
cat("\nOverdispersion Check:\n")
```

```
## 
## Overdispersion Check:
```

```
print(overdisp)
```

```
## # Overdispersion test
## 
## dispersion ratio = 0.942
##      p-value = 0.864
```

```
## No overdispersion detected.
```

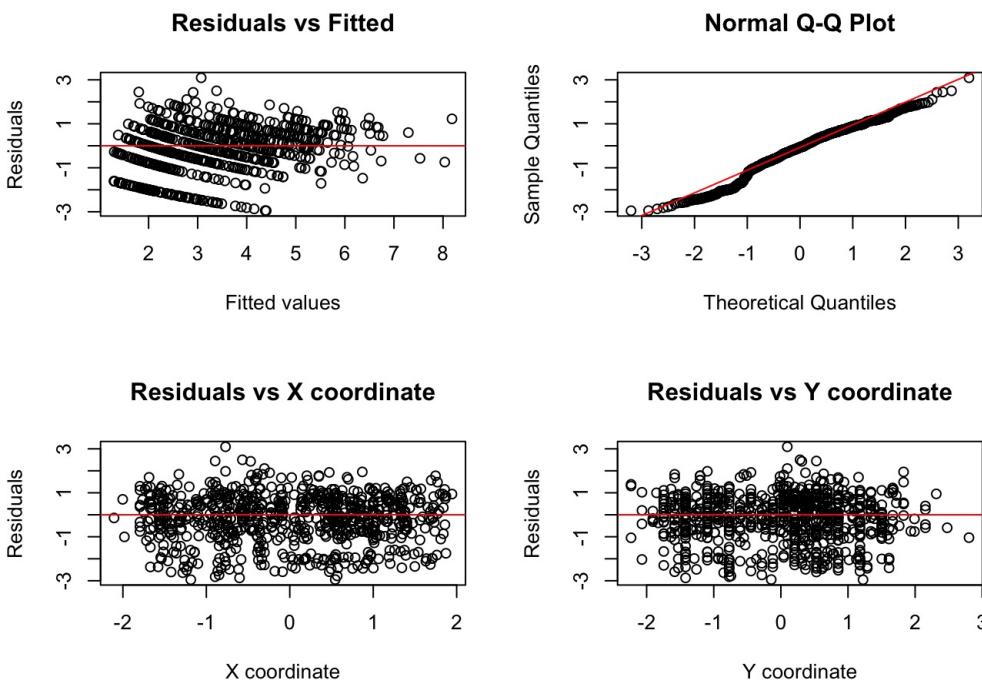
```
# plots
par(mfrow=c(2,2))

plot(fitted(best_model), residuals(best_model),
      main="Residuals vs Fitted",
      xlab="Fitted values",
      ylab="Residuals")
abline(h=0, col="red")

qqnorm(residuals(best_model))
qqline(residuals(best_model), col="red")

plot(data$x_scaled, residuals(best_model),
      main="Residuals vs X coordinate",
      xlab="X coordinate",
      ylab="Residuals")
abline(h=0, col="red")

plot(data$y_scaled, residuals(best_model),
      main="Residuals vs Y coordinate",
      xlab="Y coordinate",
      ylab="Residuals")
abline(h=0, col="red")
```



```
par(mfrow=c(1,1))

cat("\nSummary of Model Diagnostics:\n")

##  
## Summary of Model Diagnostics:

# major assumptions
if(overdisp$dispersion_ratio > 1.2) {
  cat("\n- Overdispersion detected (ratio =",
      round(overdisp$dispersion_ratio, 2), ")")
}

if(spatial_autocorr$pvalue < 0.05) {
  cat("\n- Significant spatial autocorrelation detected (p =",
      round(spatial_autocorr$pvalue, 3), ")")
}

shapiro_test <- shapiro.test(residuals(best_model))
if(shapiro_test$p.value < 0.05) {
  cat("\n- Non-normal residuals detected (Shapiro-Wilk p =", 
      round(shapiro_test$p.value, 3), ")")
}
```

```
##  
## - Non-normal residuals detected (Shapiro-Wilk p = 0 )
```