Workshop

Exercise # 4: Temporal trends in fantail darter abundance

Goal

The goal of this exercise is to gain experience fitting a Bayesian hierarchical model using the `rstanarm` package and to practice using **R** functions learned during the workshop. The model will be a varying intercept, varying slope regression model.

The dataset

For this exercise we will model catch of the fantail darter sampled at multiple locations and across years in PA. Specifically, we are interested if there are trends in fantail darter catch (I will refer to the catch as "abundance" [N] even though the data do not represent actual abundance estimates). However, we are not interested in trends at any given site – there is not enough data to make site-specific inferences, rather we are interested if there are trends at the HUC 8 watershed level. This will also allow us to assess if there is spatial variability in these "regional" trends.

HUC 8 data summaries

There are a total of 33 HUC 8 watersheds in our data set. However, the sample size per HUC varies substantially. The number of observations per HUC ranges from 1 to 345. Table 1 shows the sample size per HUC for 15 HUCs sorted based on sample size. **We will retain the HUCs that have > 50 observations** (shown in red in Table 1; note that this is a purely arbitrary cutoff for the purpose of the exercise). The time-frame of this data set ranges from 1933 – 2013; however, no HUCs have samples in every year.

Table 1: Summary of observations per HUC 8

HUC 8	Number of observations
5020005	345
5030106	251
5010004	230
5010001	218
5010003	218
5010006	108
4120101	89
5010007	66
5030101	64
5010009	56
2050202	47
5030102	47
5020006	40
2070004	37
5010005	33

Getting started

1. Open **R** from the `.proj` file associated with this workshop (`R_Workshop.Rproj`), create a new R script, and save it (give it name of your choice) to the folder called *Ex_4* located in the folder *07_Exercises*. To create a new script, follow File –> New File –> R Script.

2. Load **R** packages. You can just copy-and-paste these from the `exercise_3.R` script, but they are also listed below.

```
library(dplyr) # data management
library(tidyverse) # data management
library(ggplot2) # plot
library(lubridate) # dates
library(stringr) # manipulate character stings
library(sf) # map creation
library(spData) # spatial data
library(car)  # logit function
library(qs) # save and read in large files
library(cmdstanr) # Bayesian estimation using stan
library(rstanarm) # Bayesian models using R functions and stan
library(mcmcplots) # as.mcmc function
library(ggrepel)
library(kableExtra) # tables
library(bayesplot)
```

3. Read in the data using `read_csv`. The data set is called `fantaildarter.csv`. The data are located in the directory `02_Data/Fischer_data`. Name the data frame `dart_dat`.

Data cleaning/manipulation

I will provide less code for this exercise. Refer to previous exercises if you need additional code guidance. Use pipes to accomplish the following:

1. Select the following columns:

```
RecordID, Date, County, Basin, N_Detected, Lat, Long, HUC_8, HUC_12
```

2. Change `Date` to a date data type.
3. Convert `County`, `Basin`, `HUC_8`, `HUC_12` to factors.
4. Remove any `NA` values in the column `N_Detected`
5. Make all column names lower case.

Your data frame should look like this:

```
str(dart_dat)
```

```
tibble [2,062 x 10] (S3: tbl_df/tbl/data.frame)
 $ recordid  : num [1:2062] 19107 19099 19119 19134 19051 ...
 $ date      : Date[1:2062], format: "1933-10-27" "1934-04-28" ...
 $ county    : Factor w/ 22 levels "adams","allegheny",..: 17 17 13 5 5 5 6 17 13 13 ...
 $ basin     : Factor w/ 8 levels "Allegheny","Erie",..: 6 6 6 6 6 6 6 6 6 6 ...
 $ n_detected: num [1:2062] 4 7 1 3 1 3 1 2 9 4 ...
 $ lat       : num [1:2062] 41.3 41.3 41 40.6 40.6 ...
 $ long      : num [1:2062] -80.3 -80.4 -80.3 -80.2 -80.5 ...
 $ huc_8     : Factor w/ 40 levels "2050103","2050104",..: 36 36 36 35 35 35 39 36 36 36 ...
 $ huc_12    : Factor w/ 521 levels "20501030802",..: 476 476 488 455 464 449 497 482 488 488
 $ year      : num [1:2062] 1933 1934 1934 1934 1934 ...
```

Try this!

Try making a summary table of the number of observations per HUC 8 (similar to Table 1, but don't worry about color formatting etc.). We can use the `group_by()` and `summarize()` functions to accomplish this.

Subset HUCs

As previously mentioned, we want to only retain those HUCs with > 50 observations. If you were able to summarize the data as in Table 1 (number of observations for each HUC) then we need to select those HUCs with > 50 observations by simply adding one more line of code piped in as follows (i.e., `filter(n > 50)`:

```
dart_subset <- dart_dat %>%
  group_by(huc_8) %>%
  summarize(n = n()) %>%
  filter(n > 50) %>%  # retain HUCs with > 50 observations
  arrange(-n) # we can sort by n as well
```

We now have a total of 10 HUCs with are desired sample size.

```
dim(dart_subset)
```

```
[1] 10  2
```

```
head(dart_subset)
```

```
# A tibble: 6 x 2
  huc_8       n
  <fct>   <int>
1 5020005   345
2 5030106   251
3 5010004   230
4 5010001   218
5 5010003   218
6 5010006   108
```

Notice that `dart_subset` has 10 rows, one row for each HUC that met our sample size criterion. The next step is to select all the data from `dart_dat` for these 10 HUCs we identified in `dart_subset`. To accomplish this we can use handy syntax for selecting data from one data frame (i.e., `dart_dat`) that match the IDs (a vector of HUC 8 codes in our case) contained in another data frame (i.e., `dart_subset`). While we are at it, let's natural log transform abundance (`n_detected`) for use as our response variable in modeling. We need to add a constant to accommodate zero abundances during log-transformation. We will add a constant of 1 prior to log-transformation. The code is below:

```
dart_dat <- dart_dat %>%
  subset(huc_8 %in% dart_subset$huc_8) %>%
  mutate(log_n = log(n_detected+1)) %>% # log-transform N (add 1 for 0's)
  droplevels() # drop unused levels from a factor (i.e., the huc_8 IDs we removed)
```

Useful subsetting syntax

Note the syntax in the `subset()` function above. The first argument `huc_8` is referencing the `huc_8` column in our `dart_dat` data frame. The IN operator (`%in%`) is checking if the values of the first vector (`huc_8` in `dart_dat`) are present in the second vector (`dart_subset$huc_8`) and returning a logical vector indicating if there is a match or not. This way we can select all

> the rows in `dart_dat` that have `huc_8` codes contained in `dart_subset$huc_8`.

Our data frame `dart_dat` now only has data for our 10 target HUCs.

```
unique(dart_dat$huc_8)
```

```
 [1] 5030101 5010001 5010004 5010003 5020005 5010007 5010006 4120101 5030106
[10] 5010009
10 Levels: 4120101 5010001 5010003 5010004 5010006 5010007 5010009 ... 5030106
```

The structure of `dart_dat` is:

```
str(dart_dat)
```

```
tibble [1,645 x 11] (S3: tbl_df/tbl/data.frame)
 $ recordid  : num [1:1645] 19134 19051 19113 19440 19183 ...
 $ date      : Date[1:1645], format: "1934-06-16" "1934-06-18" ...
 $ county    : Factor w/ 16 levels "allegheny","Armstrong",..: 3 3 3 8 5 5 5 6 6 6 ...
 $ basin     : Factor w/ 5 levels "Allegheny","Erie",..: 5 5 5 1 1 1 1 1 1 1 ...
 $ n_detected: num [1:1645] 3 1 3 4 2 3 2 3 1 10 ...
 $ lat       : num [1:1645] 40.6 40.6 40.6 41.8 41.5 ...
 $ long      : num [1:1645] -80.2 -80.5 -80.3 -78.5 -80.1 ...
 $ huc_8     : Factor w/ 10 levels "4120101","5010001",..: 9 9 9 2 4 4 4 4 4 4 ...
 $ huc_12    : Factor w/ 227 levels "41201010102",..: 211 219 206 19 114 100 98 112 110 94 ...
 $ year      : num [1:1645] 1934 1934 1934 1935 1935 ...
 $ log_n     : num [1:1645] 1.386 0.693 1.386 1.609 1.099 ...
```

> Try this!
>
> Create a map of all the locations (lat/longs) of our sample sites after we subsetted the data. Make it however you want/can, but an example is in Figure 1.

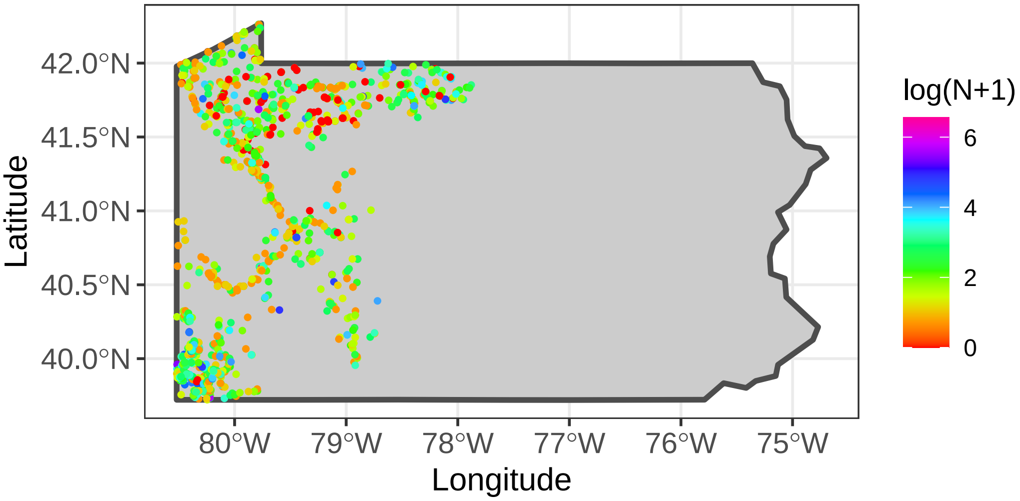


Figure 1: Locations of sample sites after retaining HUC 8s with > 50 observations (n = 10 HUC 8s). Points are colored according the log(abundance).

Temporal trends

The next step is to fit a model to examine temporal trends in each HUC 8. Table 2 is a quick-and-dirty table summarizing our 10 HUC 8's, along with temporal dynamics based on annual average abundance.

We can see from Table 2 that HUC 8s vary in average abundance and temporal dynamics are variable. We can also plot the actual data (not taking annual means) as shown in Figure 2.

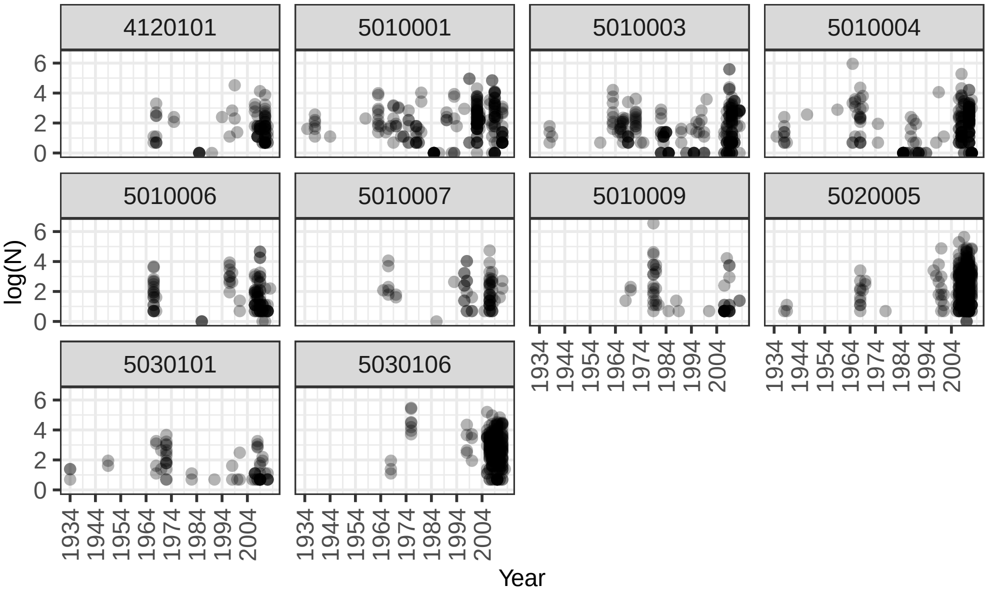
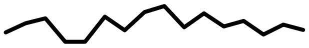
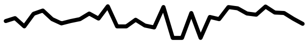
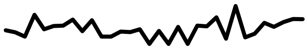
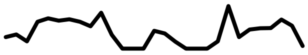
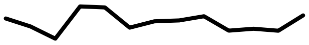
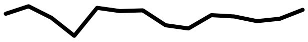
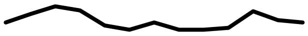
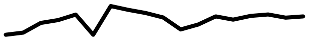
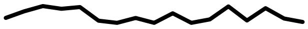
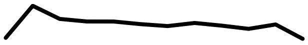


Figure 2: Fantail darter abundance over time for 10 HUC 8 watersheds.

Table 2: Summary of fantail darter mean abundance and temporal trends.

HUC 8	Mean abundance	Temporal trend (log[N])
4120101	7.5	
5010001	13.7	
5010003	9.9	
5010004	11.3	
5010006	10.1	
5010007	12.1	
5010009	25.0	
5020005	19.9	
5030101	6.2	
5030106	25.6	

The statistical model

We will fit a simple trend model and investigate the presence of monotonic (linear) trends in abundance among HUCs.

The model is as follows:

$$y_i \sim N\left(\alpha_{j(i)} + \beta_{j(i)} \cdot x_i, \sigma^2\right), \text{for } i, \dots n$$

$$\begin{pmatrix}\alpha_j \\ \beta_j\end{pmatrix} \sim MVN\left(\begin{pmatrix}\mu_\alpha \\ \mu_\beta\end{pmatrix}, \begin{pmatrix}\sigma_\alpha^2 & \sigma_{\alpha\beta} \\ \sigma_{\alpha\beta} & \sigma_\beta^2\end{pmatrix}\right), \text{ for } j = 1, \dots J$$

In this model, y_i is the log_e-transformed fantail darter abundance for site i, and x_i is the standardized (mean = 0, SD = 1) year predictor variable. Again, it is good practice to standardize all predictors prior to analysis such that all variables have a mean of zero and a standard deviation of one. Standardizing not only facilitates parameter interpretation, but will also aid in the convergence of hierarchical models. The parameters α_j and β_j are HUC 8-specific intercepts and slopes, and σ^2 is the model error term variance. The HUC-specific intercepts and slopes are assumed to come from a multivariate normal distribution (MVN), where μ_α and μ_β are the population-average intercept and slope parameters describing the average temporal trend across all data and HUCs. The parameters σ_α^2, σ_β^2, and $\sigma_{\alpha\beta}$ are the variances among intercepts and slopes, and the covariance, respectively. We will use default priors in `stan_glmer` when we fit this model. In terms of inferences about temporal trends, the β_j parameters will tell us about HUC-specific trends and μ_β gives us an estimate of the overall trends, across all HUCS.

Fitting the model

We will use the `stan_glmer` function to fit this model. The code is in the following **R** chunk. Note that we will first standardized (mean = 0, SD = 1) the `year` predictor variable and call it `z_year`.

```r
# Standardize year predictor
dart_dat <- dart_dat %>%
  mutate(z_year = as.numeric(scale(year)))

######## --------------------------------
# Fit varying intercept and slope model
m1 <- stan_glmer(formula = log_n ~ 1 + z_year + (1 + z_year | huc_8),
                 family = gaussian,
                 data = dart_dat,
                 iter = 1500, chains = 3)
# print(m1, digits=3)
```

The random effects part of the model is contained in the `(1 + z_year | huc_8)`, where `1` indicates a varying (random) intercept, `z_year` indicates a varying slope for `z_year` and `| huc_8` indicates that the intercepts (α_j) and slopes (β_j) vary according to HUC 8. The fixed intercept (μ_α) and slope (μ_β) are identified in `~ 1 + z_year`.

This model may take several minutes to fit, depending on your CPU.

Model convergence

Examine trace plots for a few parameters (we would want to look at all trace plots.)

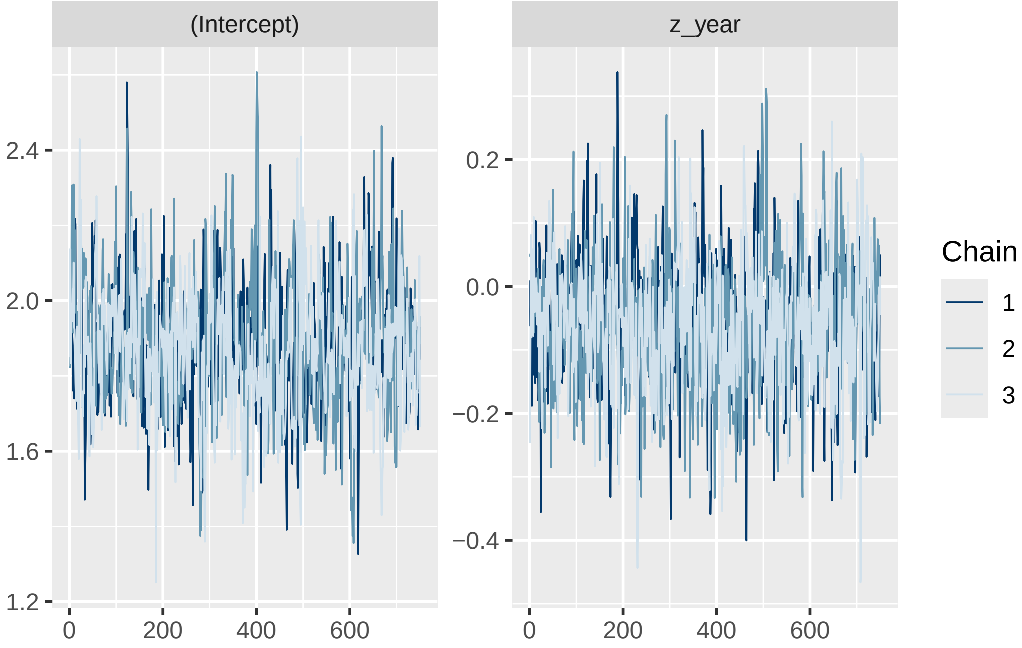


Figure 3: Trace plots for population-average parameters.

Table 3: Posterior summaries

Parameter	Posterior mean	SD	Lower 95% CI	Upper 95% CI
(Intercept)	1.88	0.16	1.57	2.22
z_year	-0.06	0.10	-0.26	0.14
b[(Intercept) huc_8:4120101]	-0.26	0.20	-0.67	0.10
b[z_year huc_8:4120101]	0.11	0.14	-0.15	0.40
b[(Intercept) huc_8:5010001]	0.15	0.18	-0.22	0.49
b[z_year huc_8:5010001]	0.16	0.11	-0.06	0.40
b[(Intercept) huc_8:5010003]	-0.32	0.18	-0.69	0.01
b[z_year huc_8:5010003]	0.08	0.11	-0.16	0.30
b[(Intercept) huc_8:5010004]	-0.27	0.18	-0.63	0.07
b[z_year huc_8:5010004]	0.10	0.11	-0.12	0.32
b[(Intercept) huc_8:5010006]	-0.17	0.19	-0.55	0.20
b[z_year huc_8:5010006]	-0.03	0.13	-0.30	0.23
b[(Intercept) huc_8:5010007]	0.10	0.21	-0.32	0.50
b[z_year huc_8:5010007]	-0.08	0.16	-0.41	0.23
b[(Intercept) huc_8:5010009]	-0.19	0.21	-0.60	0.23
b[z_year huc_8:5010009]	-0.37	0.17	-0.73	-0.05
b[(Intercept) huc_8:5020005]	0.40	0.18	0.05	0.74
b[z_year huc_8:5020005]	0.29	0.13	0.03	0.56
b[(Intercept) huc_8:5030101]	-0.41	0.21	-0.83	-0.03
b[z_year huc_8:5030101]	-0.10	0.13	-0.36	0.14
b[(Intercept) huc_8:5030106]	0.94	0.19	0.57	1.31
b[z_year huc_8:5030106]	-0.18	0.17	-0.53	0.15
sigma	1.14	0.02	1.10	1.18
Sigma[huc_8:(Intercept),(Intercept)]	0.23	0.14	0.08	0.57
Sigma[huc_8:z_year,(Intercept)]	0.00	0.05	-0.10	0.10
Sigma[huc_8:z_year,z_year]	0.08	0.06	0.02	0.24

Table 3 contains posterior summaries for all parameters.

Trend inferences

The population-average estimated temporal trend is negative (μ_β = -0.059), but the 95% credible interval overlaps with zero (95% credible interval = -0.250, 0.113). The posterior probability of a negative population-average trend is 0.76.

Visulaize HUC 8-specific model fits and trends

Figure 4 shows the HUC-specific trends. We can see that trends (direction and magnitude) vary across HUCs and that support of a decline (represented by the posterior probability of β_j <0) also varys.

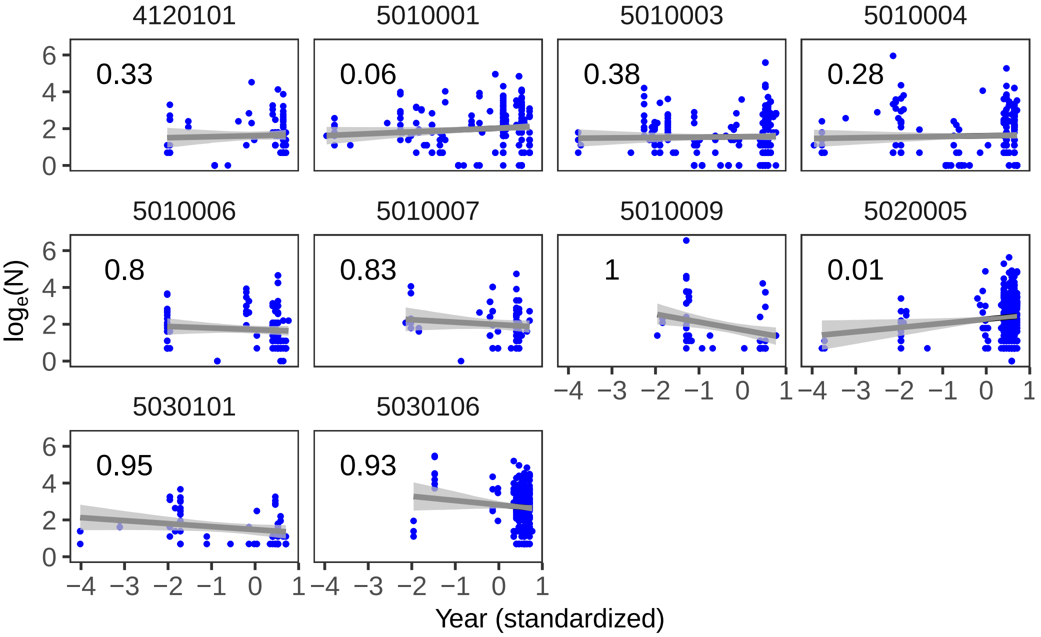


Figure 4: HUC-specific trends in fantail darter abundance. Fitted line is the posterior mean and shaded area is 95% credible region. Numbers are the posterior probability of a negative trend in abundance.