

## Box 2. Step-by-step instructions for determining if a population is at equilibrium.

2025-01-10

```
knitr::opts_chunk$set(echo = TRUE)

# load libraries
library(tidyverse) # data management
library(cowplot) # prettier default plots
library(mgcv) # fit gams
library(gratia) # visualize derivative
# library(itsadug) # another useful package for visualizing gams, but not used here
library(ggokabeito)
library(tidygam)

# set prettier plot theme
theme_set(theme_cowplot())

# load and manipulate data
test_df <- read_csv("beetle_test_df.csv") %>%
  mutate(ID = factor(ID)) # necessary to convert character variables
  # to factors for fitting random effects
```

A system is at equilibrium (in the strictest and most classic sense of ‘constancy’) when there is no net change in its state variable(s) (e.g. population abundance) over time (Box 1). Mathematically, this is when the first derivative of the variable of interest with respect to time is equal to zero. Here, we describe how to use generalized additive mixed models (GAMMs) to statistically determine whether a system is outside equilibrium or not. GAMMs provide a flexible framework for modeling timeseries data. In particular, different types of variables can be modeled (e.g. counts, presence/absence, biomass, etc.), and complex experimental designs (i.e. crossed/nested random effects), residual temporal autocorrelation, and irregularly spaced timeseries can be accommodated (Pedersen et al. 2019). In addition, they are well suited for detecting periods of change in timeseries data (Simpson 2018; Pedersen et al. 2020). Pedersen et al. (2019) (Pedersen et al. 2019) provide an excellent introduction to fitting and comparing GAMMs, and we adopt their approach here.

To illustrate this approach, we used a GAMM to analyze the population dynamics of flour beetles (*Tribolium castaneum*) sampled weekly over a period of 11 weeks from three independent populations experiencing the same environmental conditions. Full R code is available at [https://github.com/mabarbour/Equilibrium\\_conundrum\\_code.git](https://github.com/mabarbour/Equilibrium_conundrum_code.git). The variable of interest is the natural log abundance ( $\ln N$ ) of beetles over time. The first derivative of this variable with respect to time corresponds to the per-capita population growth rate ( $d\ln N(t)/dt$ ). When  $d\ln N(t)/dt = 0$ , the beetle population is at equilibrium. We first fit our GAMM with the following code:

```
GAMM <- mgcv::gam(log(Abundance) ~ s(Week) + s(ID, bs="re") +
  s(Week, ID, bs="re"),
  data = test_df, method = "REML")
```

Here, we have modeled the natural log abundance of the beetle population over time ( $s(\text{Week})$ ) and allowed each independent population to have a different intercept ( $s(\text{ID}, \text{bs}="re")$ ) and slope ( $s(\text{Week}, \text{ID}, \text{bs}="re")$ )

(i.e. random intercept and slope). This model fit the data well (Fig. 1/3a), outperformed other potential models (e.g. random intercept only, random smooths), and there was no evidence of residual autocorrelation (see supplementary code online).

From this GAMM, we calculated the mean and 95% confidence interval of the first derivative of the model's fit throughout the entire experiment with the following code:

```
derivs <- gratia::derivatives(GAMM)
```

This allowed us to determine the time periods of the experiment where the beetle population was at equilibrium (red thicker lines in Fig. 1/3b below). Specifically, we can say that there is no clear change in the beetle population over these time periods, because the 95% confidence interval overlaps with zero (dashed line in Fig. 1/3a below). From this figure, we can also see that there are multiple transient periods of zero detectable change. Knowing whether the study system is potentially in a transient state requires knowledge about the study system in question and careful reflection. For the flour beetles, we conclude that the population has reached an equilibrium by week 11, because the model indicates that there was no net change for two consecutive time points, whereas previous transient states always occurred between sampling periods.

One challenge with our approach is that the equilibrium state is the null model. In other words, we default to concluding the system is at equilibrium if the confidence intervals of the estimate of the first derivative overlap with zero. This makes our approach susceptible to concluding that a system is at equilibrium at low sample sizes when greater uncertainty widens the confidence intervals. Therefore, we always recommend plotting the model and inferred change in the system (Fig. 1/3 below) and using knowledge of the system to conclude whether there is sufficient data to make an assessment of whether the system is at equilibrium or not. Mixed models are helpful in this regard because they efficiently pool information from multiple replicates. Analysis of individual timeseries, in contrast, would require more sampling points to determine whether they are at equilibrium or not.

More detailed code for fitting alternative models and plotting the results is provided below:

```
## fit model following model types from Fig. 4 of Pedersen et al. 2019, PeerJ

# global smoother with no random effects
model_G_noRE <- gam(log(Abundance) ~ s(Week, k=11, bs="tp", m=2),
  data = test_df, method = "REML", family="gaussian")

# global smoother with random intercept for each cage (ID)
model_G <- gam(log(Abundance) ~ s(Week, k=11, bs="tp", m=2) +
  s(ID, k=3, bs="re"),
  data = test_df, method = "REML", family="gaussian")

# global smoother with random intercepts and slopes for each cage
model_G_RS <- gam(log(Abundance) ~ s(Week, k=11, bs="tp", m=2) +
  s(ID, k=3, bs="re") + s(Week, ID, k=3, bs="re"),
  data = test_df, method = "REML", family="gaussian")

# global smoother with different group-levels smoothers with the same "wiggleness"
model_GS <- gam(log(Abundance) ~ s(Week, k=11, bs="tp", m=2) +
  s(Week, ID, k=11, m=2, bs="fs"),
  data = test_df, method = "REML", family="gaussian")

## Warning in gam.side(sm, X, tol = .Machine$double.eps^0.5): model has repeated
## 1-d smooths of same variable.

# global smoother with different group-level smoothers with differing "wiggleness"
model_GI <- gam(log(Abundance) ~ s(Week, k=11, bs="tp", m=2) +
  s(Week, by=ID, k=11, m=1, bs="tp") + s(ID, k=3, bs="re"),
```

```

        data = test_df, method = "REML", family="gaussian")

# group-specific smoothers without a global smoother, but all smoothers have the same "wiggliness"
model_S <- gam(log(Abundance) ~ s(Week, ID, k=11, m=2, bs="fs"),
               data = test_df, method = "REML", family="gaussian")

# group-specific smoothers with different levels of "wiggliness"
model_I <- gam(log(Abundance) ~ s(Week, by=ID, k=11, m=2, bs="tp") +
               s(ID, k=3, bs="re"),
               data = test_df, method = "REML", family="gaussian")

# AIC model comparison
AIC(model_G_noRE, model_G, model_G_RS, model_GS, model_GI, model_S, model_I)

##              df              AIC
## model_G_noRE 11.28315 -37.207941
## model_G      13.50394 -40.928518
## model_G_RS   13.56590 -45.628953
## model_GS     14.94028 -45.113972
## model_GI     25.40912 -83.160855
## model_S      14.51303  5.564392
## model_I      13.76720  3.268248

# model_GI seems to be the clear winner, but there is residual autocorrelation.
# AIC and diagnostics suggest that model_G_RS (random slope) is best
# so we proceed with this one.

# choose model for subsequent evaluation
model_choice <- model_G_RS

## assess model fit
summary(model_choice)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(Abundance) ~ s(Week, k = 11, bs = "tp", m = 2) + s(ID, k = 3,
##      bs = "re") + s(Week, ID, k = 3, bs = "re")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.16721    0.03459   120.5  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(Week)      9.005e+00  9.777 124.888 <2e-16 ***
## s(ID)        3.634e-05  2.000   0.000  0.4440
## s(Week,ID)   1.583e+00  2.000   3.793  0.0193 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

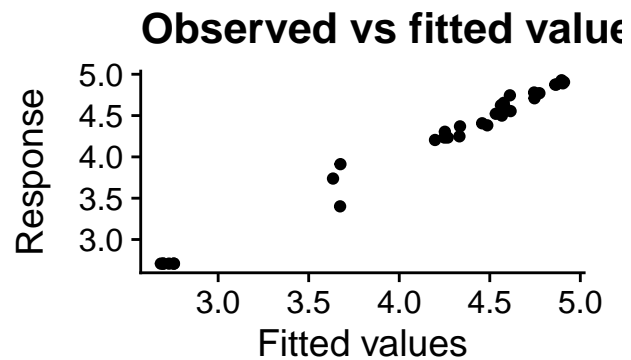
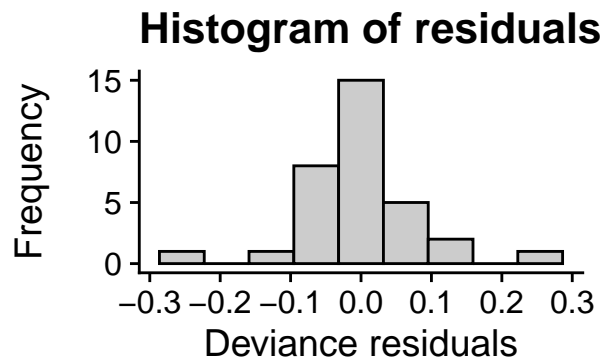
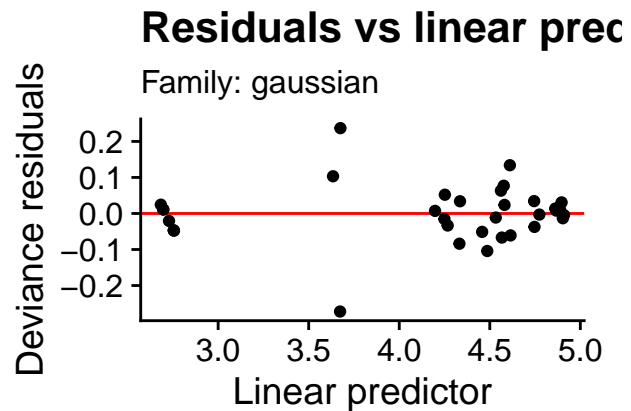
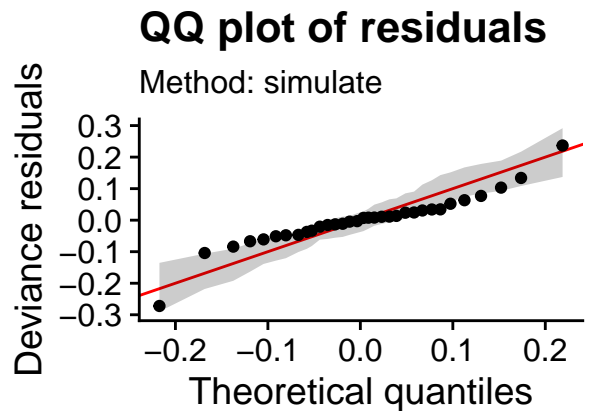
```

```
## R-sq.(adj) = 0.984   Deviance explained = 98.9%
## -REML = -7.0188   Scale est. = 0.00995   n = 33
```

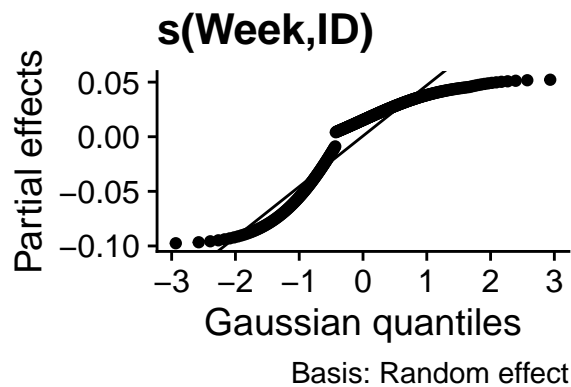
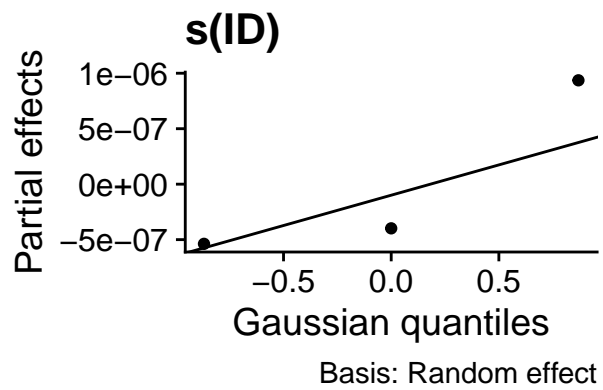
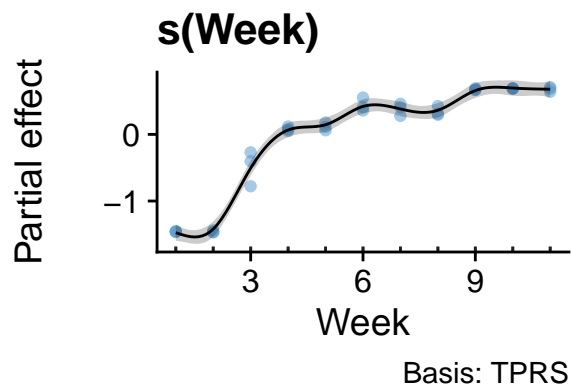
```
#gam.check(model_choice)
k.check(model_choice)
```

```
##           k'           edf k-index p-value
## s(Week)    10 9.005366e+00 1.195972 0.825
## s(ID)       3 3.633808e-05          NA    NA
## s(Week,ID)  3 1.582742e+00          NA    NA
```

```
appraise(model_choice, method = "simulate")
```



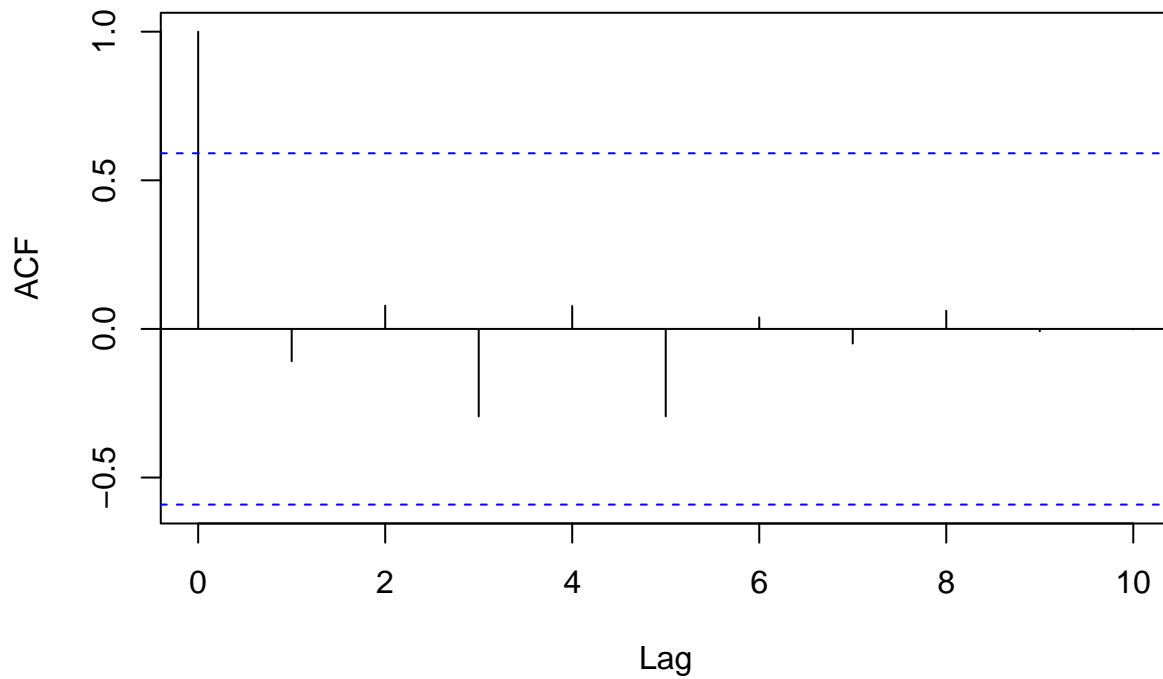
```
draw(model_choice, residuals = T)
```



```
## check for residual autocorrelation
# add residuals from model
resid_df <- test_df %>%
  ungroup() %>%
  mutate(resids = residuals(model_choice))

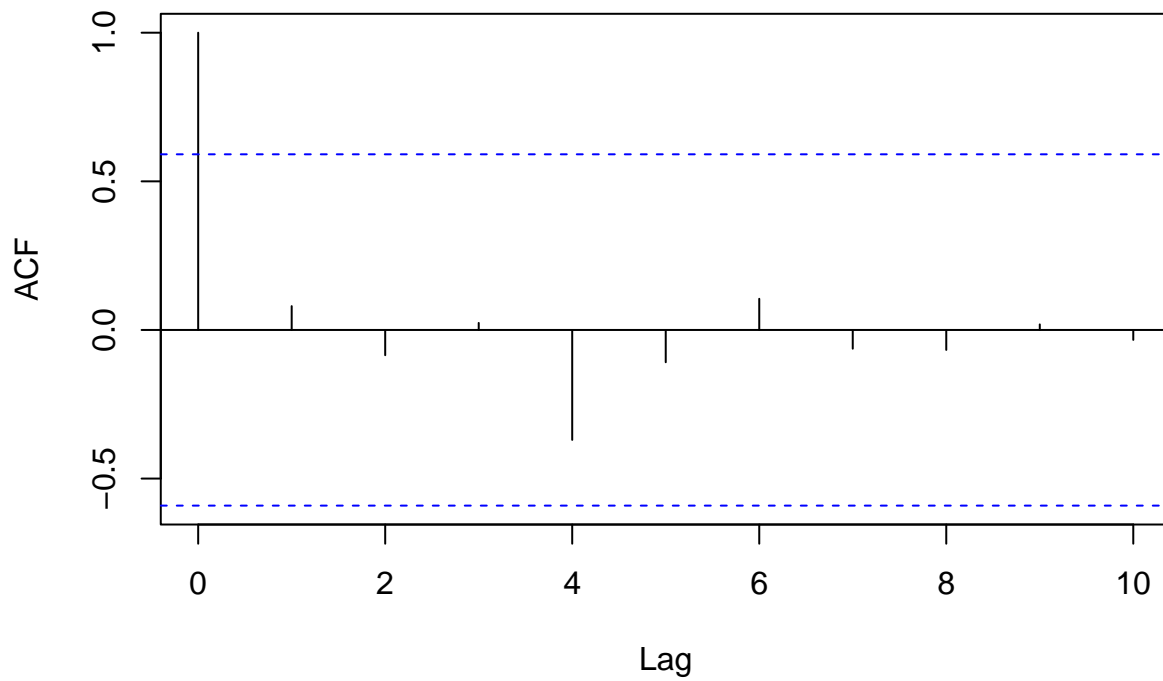
# check residual autocorrelation within each cage,
# because it doesn't make sense to check across cages
IDs <- unique(test_df$ID)
acf(filter(resid_df, ID == IDs[1])$resids)
```

**Series filter(resid\_df, ID == IDs[1])\$resids**



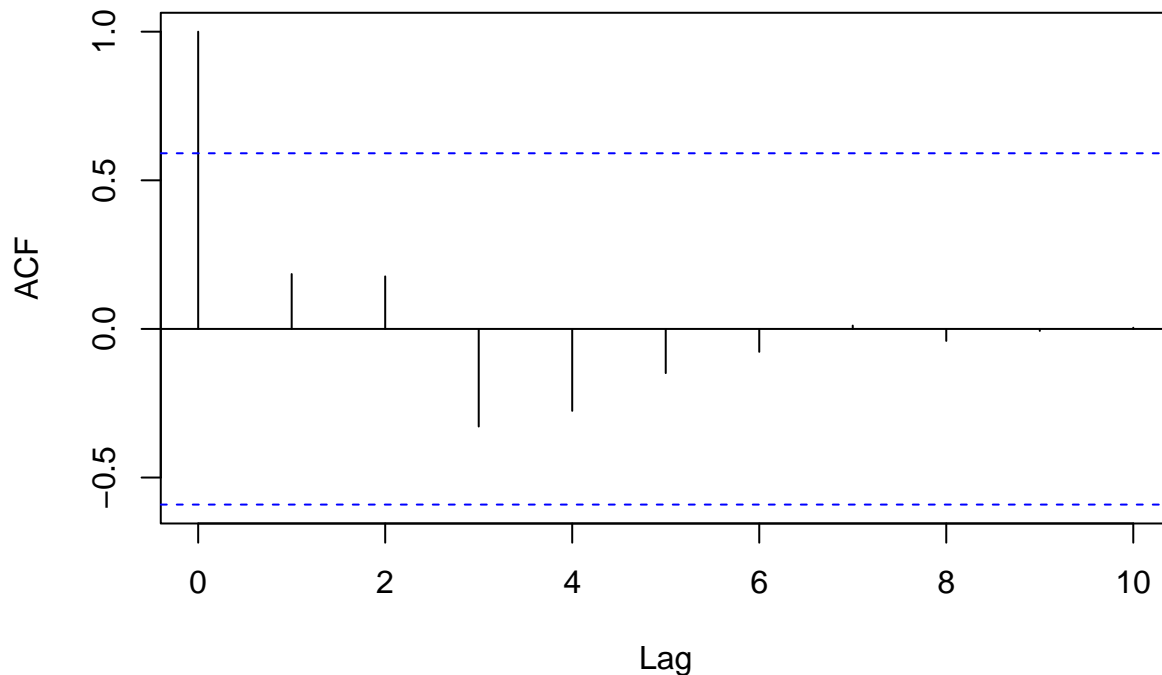
```
acf(filter(resid_df, ID == IDs[2])$resids)
```

**Series filter(resid\_df, ID == IDs[2])\$resids**



```
acf(filter(resid_df, ID == IDs[3])$resids)
```

### Series filter(resid\_df, ID == IDs[3])\$resids



```
# no clear evidence of autocorrelation for model_G_RS
```

```
# plot predictions
```

```
pred_df <- predict_gam(model_choice, series = "Week", length_out = 100, exclude_terms = c("s(ID)", "s(Week)"))
```

```
pred_plot <- ggplot(pred_df, aes(x = Week)) +
  geom_ribbon(aes(ymin = lower_ci, ymax = upper_ci, x = Week),
    alpha = 0.2) +
  geom_line(aes(y = Abundance)) +
  geom_point(data = test_df, aes(x = Week, y = Abundance), alpha = 0.5, color = "steelblue") +
  scale_x_continuous(breaks = 1:11, name = "") + # Week of experiment
  scale_y_continuous(name = bquote(Abundance~(italic("N"))),
    breaks = c(10,20,50,100,150), transform = "log")# +
```

```
# plot first derivative to detect equilibrium.
```

```
# Note that it will be the same for all three replicates because there is no random smooth term
```

```
deriv_plot <- derivatives(model_choice, order = 1) %>%
  mutate(equilib = ifelse(.upper_ci < 0 & .derivative < 0, NA,
    ifelse(.lower_ci > 0 & .derivative > 0, NA, .derivative))) %>%
  ggplot(aes(x = Week)) +
  geom_line(aes(y = .derivative)) +
  geom_ribbon(aes(ymax = .upper_ci, ymin = .lower_ci), alpha = 0.2) +
  geom_line(aes(x = Week, y = equilib), size = 2, color = "red", inherit.aes = F) +
  scale_x_continuous(breaks = 1:11, name = "Week of experiment") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  ylab(bquote(~italic("d")*ln*italic("N(t)"/italic("d")*t)))
```

```
# merge plots together
```

```
pred_deriv_plot <- plot_grid(pred_plot, deriv_plot, ncol = 1, align = "v", labels = "AUTO")
```

pred\_deriv\_plot

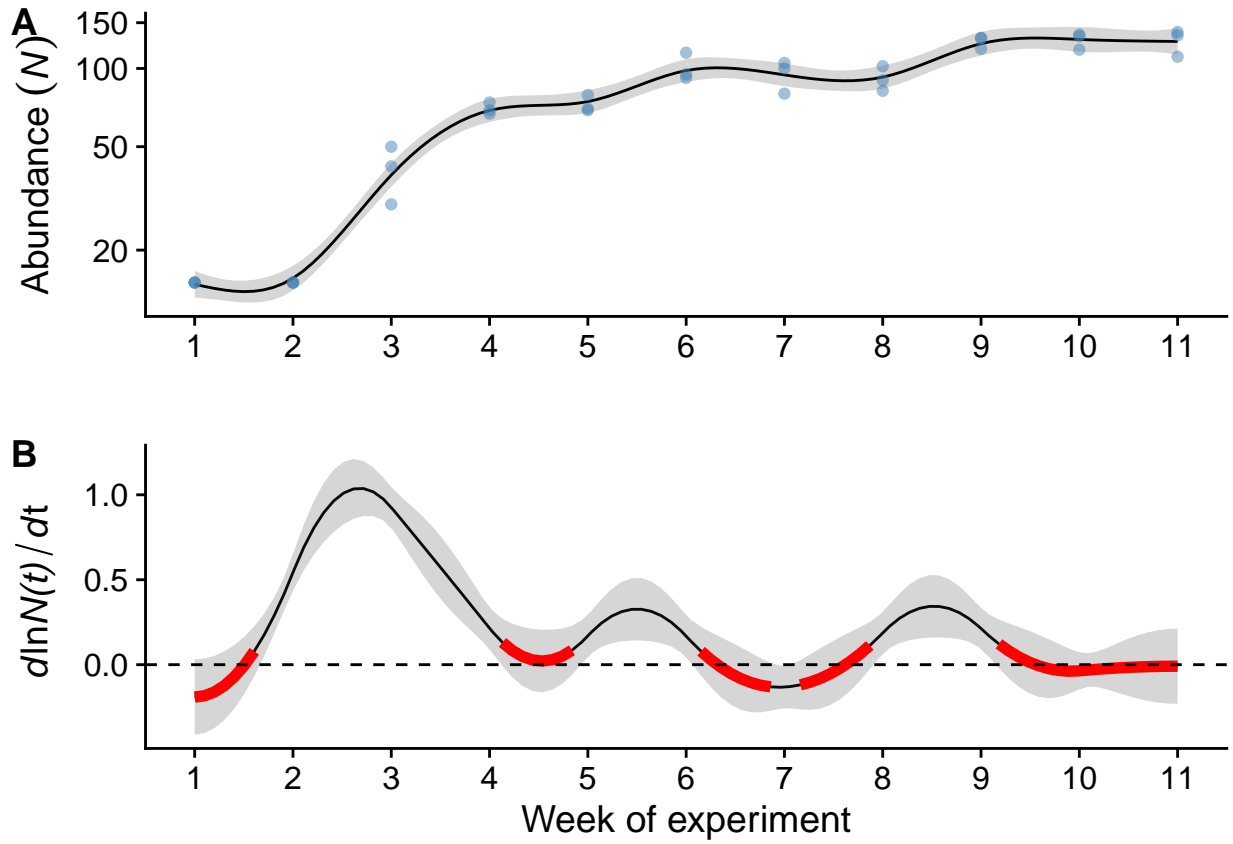


Figure 1: (Figure 3 in manuscript). Visualization of a method to whether populations are at equilibrium, applied here to populations of flour beetles (*T. castaneum*). Panel (a) shows a generalized additive mixed-model (GAMM) fit to the flour beetle abundance data (data from three populations). Panel (b) shows the estimated first derivative of the model fit, which is used to determine whether the flour beetle populations are at equilibrium or not. Red lines in panel b show areas of beetle population dynamics where there is no detectable change in per-capita population growth rates.