ESM_1. Example R code for estimating length-mass relationships by fitting log-linear models.

Length-mass relationships for lake macroinvertebrates corrected for back transformation and preservation effects.

Marlen Mährlein, Marlene Pätzig, Mario Brauns, Andrew M. Dolman

Corresponding author: Marlen Mährlein, e-mail: marlen.maehrlein@igb-berlin.de Leibniz-Institute of Freshwater Ecology and Inland Fisheries (IGB), Müggelseedamm 301, 12587 Berlin, Germany

Simulating data

In this first code block we simulate some fake length-mass data to illustrate some of the issues when estimating length-mass relationships with log-linear regression models.

```r
# load require packages
library(ggplot2)
library(knitr)
library(gridExtra)
library(dplyr)

# set seed of random number generator
# so that the simulated data set is always the same
set.seed(1)

# assign values to parameters of the length-mass relationship
a <- 0.5
b <- 3

# sample size
n <- 50

# standard deviation of simulated biological variation
sd <- 0.2

# standard deviation of simulated measurement error
sd_measure <- 1

# randomly draw lengths uniformly between 0.1, and 4 mm
dat <- data.frame(
length = runif(n, 0.1, 4)
)

# simulate observed mass given multiplicative biological variation
# and additive measurement error
dat$mass <- 10^((log10(a) + log10(dat$length) * b) + rnorm(n, mean = 0, sd = sd)) +
  rnorm(n, 0, sd_measure)

# sort data set shortest to longest
dat <- arrange(dat, length)

# print first 10 rows of data set
```

```
kable(head(dat, 10), caption = "First 10 rows of the simulated
      length-mass data, ordered by increasing length")
```

Table 1: First 10 rows of the simulated length-mass data, ordered by increasing length

length	mass
0.1522223	0.0434463
0.1909917	1.3455509
0.3409665	0.6044584
0.5209801	1.5294185
0.5896649	-1.1272670
0.7885713	1.2675724
0.8262486	0.7508307
0.8865595	-0.1671617
0.9033008	0.6373670
0.9273558	-0.8049359

Problems with negative mass estimates

Measured mass esimates can sometimes be zero or negative for the smallest individuals when their mass approaches the limits of the equipment.

Exclude mass <= 0 mg

In code chunk 2 we exclude all data points that are <= 0 and then fit a log-linear regression model.

```
dat_sub_neg <- subset(dat, mass > 0)

# fit log-linear regression model
lm1 <- lm(log10(mass) ~ log10(length), data = dat_sub_neg)

# get estimated parameter values
a_est <- 10^coef(lm1)[1]
b_est <- coef(lm1)[2]

# get predicted "fitted" masses
dat_sub_neg$fttd <- fitted(lm1)

# plot the data + fitted regression on original and log10 transformed scales
p1 <- dat %>%
  ggplot(aes(x = length, y = mass)) %>%
  + geom_point() %>%
  + scale_x_continuous(expression(Length~"[mm]")) %>%
  + scale_y_continuous(expression(Mass~"[mg]")) %>%
  + theme_bw() %>%
  + theme(aspect.ratio = 1) %>%
  + geom_line(data = dat_sub_neg, aes(x = length, y = 10^fttd))
```

```
p2 <- p1 %>%
  + scale_x_continuous(expression(Length~"[mm]"), trans = "log10", breaks = c(0.25, 1, 4)) %>%
  + scale_y_continuous(expression(Mass~"[mg]"), trans = "log10") %>%
  + expand_limits(x = 0.25, y = c(0.1, 100))

p12 <- arrangeGrob(p1, p2, nrow = 1)
p12
```

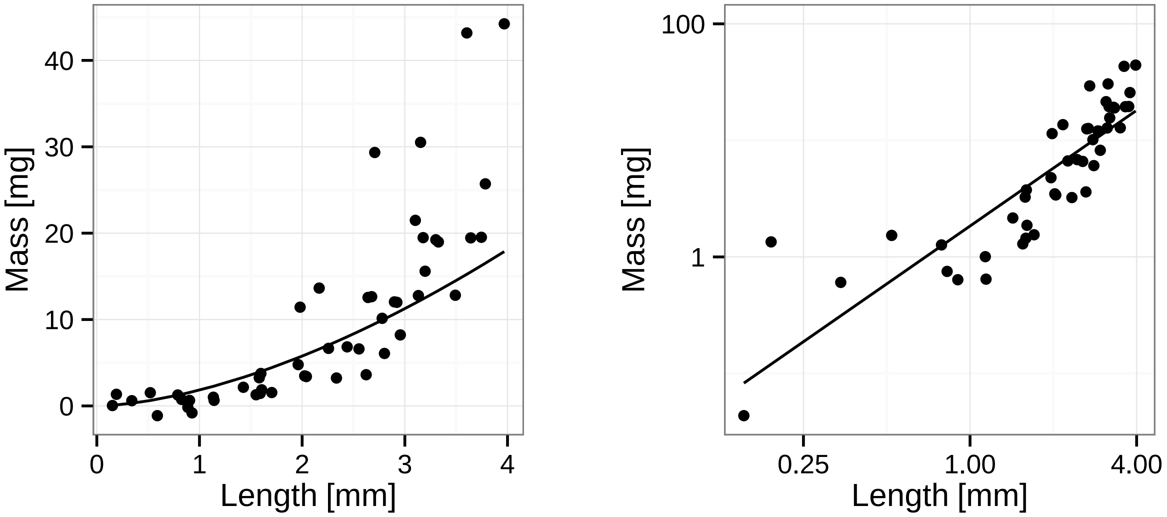

Figure 1: Simulated length-mass data on original scales (left) and $\log_{10}$ transformed scales (right). The regression line shows a log-linear model fit to the data with zero or negative mass estimates removed.

If zero and negative values are simply removed the resulting data set is distorted for short individuals. This is because only those whose mass was by chance overestimated are retained.

The estimated values of the parameters a and b are very different from the values assigned in code block 1.

a_est = 1.8409244

b_est = 1.6487659

Exclude length < 1 mm

In this code chunk we instead exclude all individuals whose lengths are below 1 mm. Then we refit the log-linear model.

```
dat_sub_short <- subset(dat, length >= 1)

lm2 <- lm(log10(mass) ~ log10(length), data = dat_sub_short)

a_est <- 10^coef(lm2)[1]
b_est <- coef(lm2)[2]

dat_sub_short$fttd <- fitted(lm2)
```

```
p3 <- dat_sub_short %>%
  ggplot(aes(x = length, y = mass)) %>%
  + geom_point() %>%
  + scale_x_continuous(expression(Length~"[mm]")) %>%
  + scale_y_continuous(expression(Mass~"[mg]")) %>%
  + theme_bw() %>%
  + theme(aspect.ratio = 1) %>%
  + geom_line(data = dat_sub_short, aes(x = length, y = 10^fttd))



p4 <- p3 %>%
  + scale_x_continuous(expression(Length~"[mm]"), trans = "log10", breaks = c(1, 2, 4)) %>%
  + scale_y_continuous(expression(Mass~"[mg]"), trans = "log10") %>%
  + expand_limits(x = 1, y = 100)

p34 <- arrangeGrob(p3, p4, nrow = 1)
p34
```

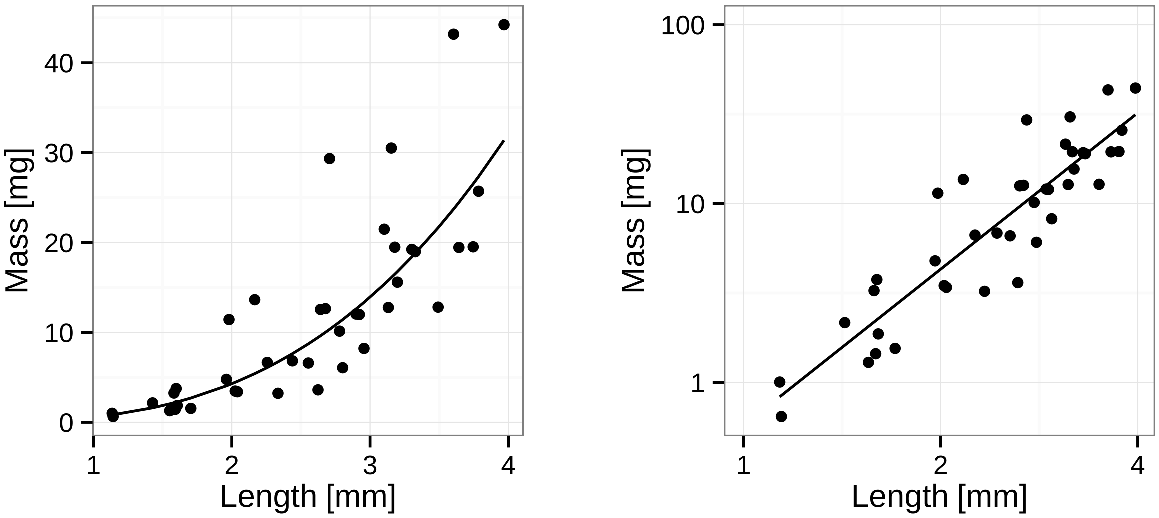

Figure 2: Simulated length-mass data with individuals < 1 mm long removed.

The estimated parameter values are now much closer to the values assigned in code block 1

a_est = 0.5744776

b_est = 2.90222

Correcting bias in mass estimates from log-linear models

The fitted length-mass relationship is now a good fit all the range of lengths but nevertheless, if an estimate is made of the total mass of the sample it will be a biased underestimate.

```r
sample_mass <- sum(dat_sub_short$mass)
est_sample_mass <- sum(10^dat_sub_short$fttd)
```

Mass of complete sample = 487 mg

Estimated mass of complete sample = 442 mg

In the following two code chunks we calculate the smearing correction factor and apply it to the predicted masses from the log-linear model

```r
# Calculate the smearing correction factor

smear_factor <- mean(10^(residuals(lm2)))

# Multiply the predicted masses by the smearing factor
dat_sub_short$fttd_smear <- 10^dat_sub_short$fttd * smear_factor
```

Smearing correction factor = 1.12

```r
sample_mass <- sum(dat_sub_short$mass)
est_sample_mass <- sum(10^dat_sub_short$fttd)
bias_corrected_est_sample_mass <- sum(dat_sub_short$fttd_smear)
```

Mass of complete sample = 487 mg

Estimated mass of complete sample = 442 mg

Bias corrected mass of complete sample = 494 mg

Finally we plot the uncorrected and bias corrected length-mass relationships on the original untransformed scale.

```r
p5 <- p3 %>%
  + geom_line(data = dat_sub_short, aes(x = length, y = fttd_smear), colour = "Red")
p5
```

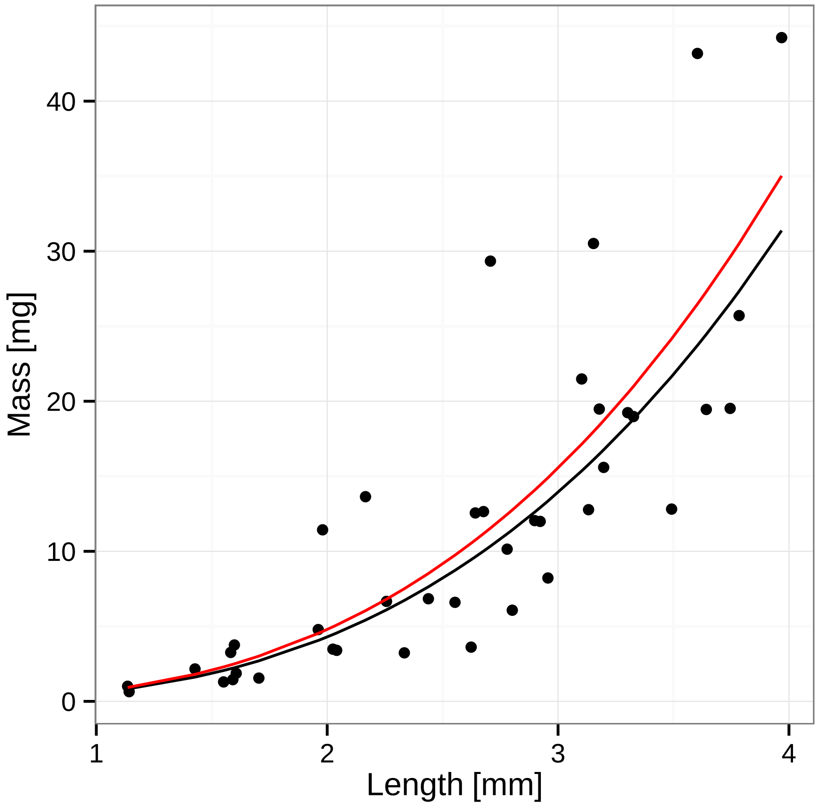

Figure 3: Simulated length-mass data with fitted log-linear regression model (black) and bias-corrected log-linear model (red).