Statistics Tips

Øystein Sørensen

2022-01-27

Contents

Chapter 1

About

This book is intended to be an ever-growing repository of statistics tips and tricks for the Center for Lifespan Changes in Brain and Cognition. I may not be able to add appropriate references everywhere, but in general the books Wood [2017] and Pinheiro and Bates [2000] have been particularly useful for my own understanding.

Chapter 2

Generalized Additive Mixed Models

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
theme_set(theme_bw())
library(mgcv)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

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

```
## This is mgcv 1.8-38. For overview type 'help("mgcv-package")'.
```

The utility of GAMMs for estimating lifespan brain trajectories is described in Fjell et al. [2010] and Sørensen et al. [2021]. The main R packages for GAMMs are `mgcv` and `gamm4`.

2.1 Scanner/batch effects

A common problem is that longitudinal data have been collected on different scanners. There can be systematic differences between values estimated on different scanners, and they can have different noise levels. This chapter shows how to correct for both of these effects, meaning that such scanner difference won't have any biasing effect on the parameter estimates.

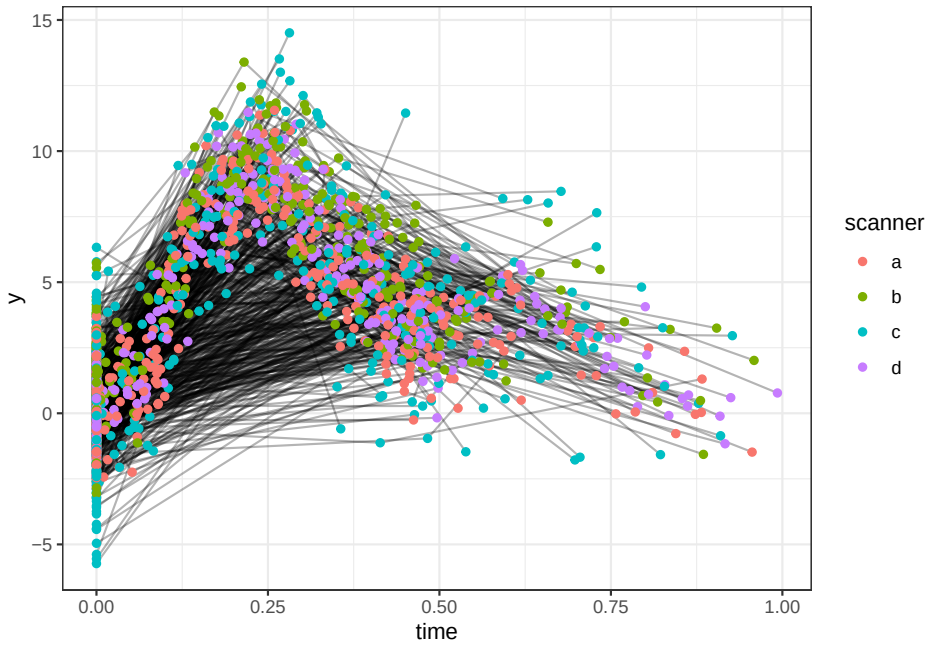
We will simulate some data to illustrate the problem.

```
scanners <- letters[1:4]
scanner_bias <- c(0, 1, .4, .2)
scanner_noise <- c(1, 1, 2, .5)
names(scanner_bias) <- names(scanner_noise) <- scanners
n <- 1000

set.seed(9988)
dat <- tibble(
  id = seq_len(n),
  time = 0,
  random_intercept = rnorm(n)
  ) %>%
  mutate(num_observations = sample(1:3, size = nrow(.), replace = TRUE)) %>%
  uncount(num_observations) %>%
  group_by(id) %>%
  mutate(timepoint = row_number()) %>%
  ungroup() %>%
  mutate(
    time = if_else(timepoint == 1, time, runif(nrow(.), max = .5)),
    scanner = factor(sample(scanners, size = nrow(.), replace = TRUE))
    ) %>%
  group_by(id) %>%
  mutate(time = cumsum(time)) %>%
  ungroup() %>%
  mutate(
    noise = rnorm(nrow(.), sd = scanner_noise[scanner]),
    bias = scanner_bias[scanner],
    y = 0.2 * time^11 * (10 * (1 - time))^6 + 10 *
      (10 * time)^3 * (1 - time)^10 + bias + noise + random_intercept
  ) %>%
  select(-noise, -bias, -timepoint)
```

Here is a spaghetti plot of the data.

```
ggplot(dat, aes(x = time, y = y, group = id)) +
  geom_line(alpha = .3) +
  geom_point(aes(color = scanner))
```

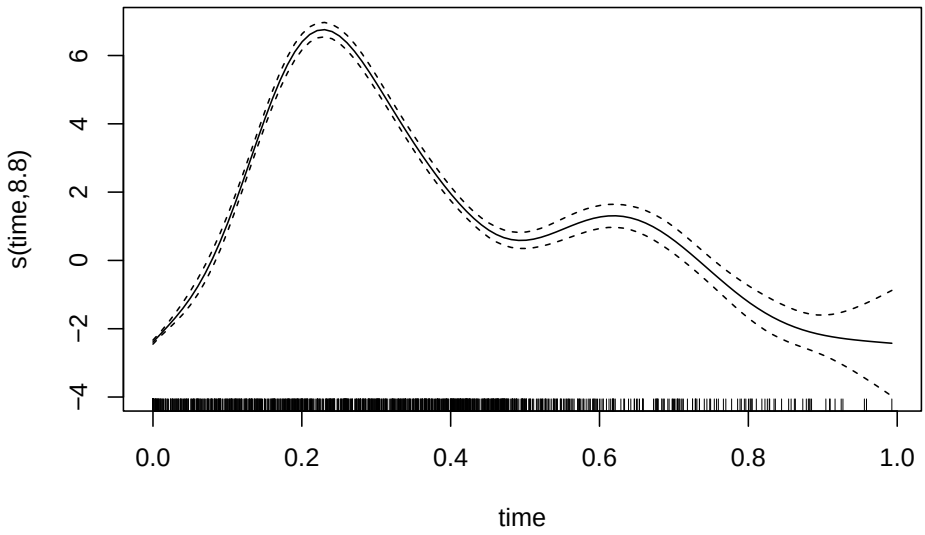


There are two ways of correcting for scanner bias. We can either include scanner as a fixed effect, or we can include it as a random effect. With as few as 4 scanners this will not make much of a difference in practice, but the interpretations of the models are a bit difference. With *fixed effects* we are interested in the specific scanners in this study, and want to estimater *their* bias. With *random effects* we would consider scanners as samples from some population of scanners, and our interest would be in the variation between scanners. Given the limited number of scanners, we use fixed effects in this example.

```
mod1 <- gamm(y ~ s(time) + scanner, random = list(id =~ 1),
             data = dat)
```

We can plot the model fit.

```
plot(mod1$gam)
```



And inspect the output. We that the `scanner` term has discovered that there are systematic differences between the scanners. It won't be exact, since this is a randoom sample, but it points in the right directions.

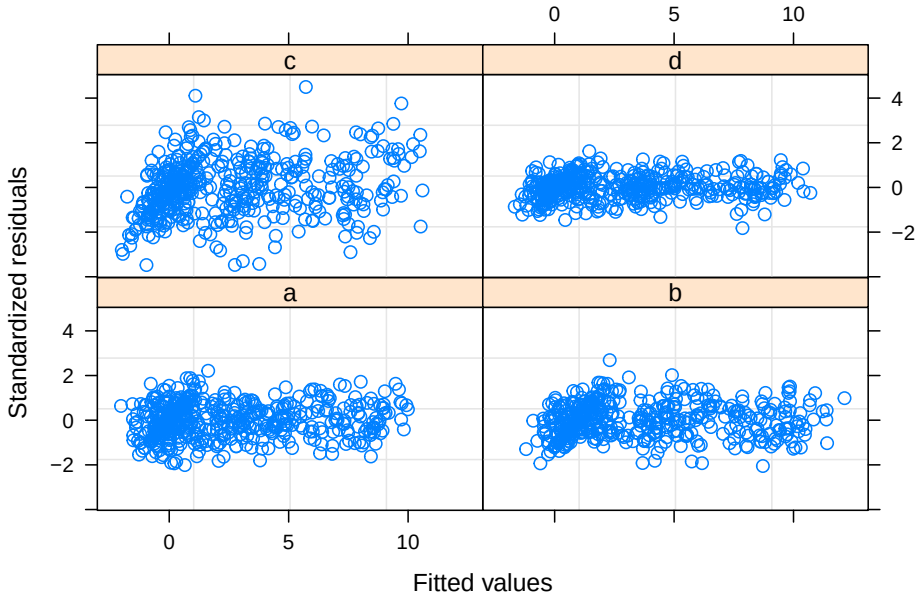
```
summary(mod1$gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## y ~ s(time) + scanner
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.36789    0.07091  33.392   <2e-16 ***
## scannerb     0.98792    0.09460  10.443   <2e-16 ***
## scannerc     0.14460    0.09211   1.570   0.1166
## scannerd     0.16303    0.09376   1.739   0.0822 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df    F p-value
## s(time) 8.795  8.795 1100  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.775
```

```
##   Scale est. = 1.6241    n = 2026
```

The model does however assume identical residuals, regardless of scanner. We can produce a diagnostic plot showing the residuals by scanner, which shows that this assumption is not correct (as we already new). In particular, scanner d has much lower residuals than scanner c.

```
plot(mod1$lme, form = resid(., type = "pearson") ~ fitted(.) | scanner)
```



We can allow the residual standard deviation to differ between scanners.

```
mod2 <- gamm(y ~ s(time) + scanner, random = list(id =~ 1),
             weights = varIdent(form = ~ 1  | scanner), data = dat)
```

Looking at the model output, under `Variance function:`, we see the multipliers for each scanner.

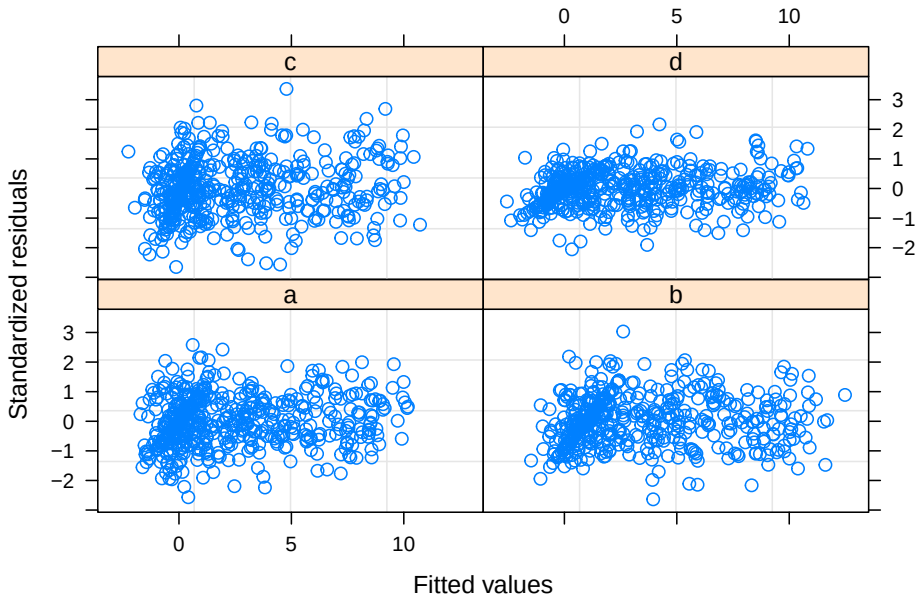
```
mod2$lme
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: strip.offset(mf)
##   Log-likelihood: -3574.53
##   Fixed: y.0 ~ X - 1
## X(Intercept)    Xscannerb    Xscannerc    Xscannerd  Xs(time)Fx1
##    2.3640655    0.9799803    0.1639192    0.1511269    2.3047793
##
## Random effects:
##  Formula: ~Xr - 1 | g
##  Structure: pdIdnot
```

```
##              Xr1      Xr2      Xr3      Xr4      Xr5      Xr6      Xr7      Xr8
## StdDev: 24.80835 24.80835 24.80835 24.80835 24.80835 24.80835 24.80835 24.80835
##
##  Formula: ~1 | id %in% g
##         (Intercept) Residual
## StdDev:    1.038648 1.036847
##
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | scanner
##  Parameter estimates:
##         a         b         c         d
## 1.0000000 0.9911869 1.9116612 0.4746387
## Number of Observations: 2026
## Number of Groups:
##         g id %in% g
##         1      1000
```

The diagnostic plot looks more reasonable now.

```
plot(mod2$lme, form = resid(., type = "pearson") ~ fitted(.) | scanner)
```



We can formally compare the models, and the second model wins.

```
anova(mod1$lme, mod2$lme)
```

```
##           Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## mod1$lme      1  8 7594.727 7639.637 -3789.363
```

```
## mod2$lme     2 11 7171.059 7232.811 -3574.530 1 vs 2 429.6675  <.0001
```

These corrections will only work if there is some amount of mixing between scanner and age/time. In contrast, if we had a longitudinal setting in which people at age 40 were scanned with a given scanner, and came back at age 45 for another session with a new scanner, then there would be no overlap between age and scanner, and we would not be able to distinguish age effects from scanner effects.

Bibliography

Anders. M. Fjell, Kristine B. Walhovd, Lars T. Westlye, Ylva Østby, Christian K. Tamnes, Terry L. Jernigan, Anthony Gamst, and Anders M. Dale. When does brain aging accelerate? Dangers of quadratic fits in cross-sectional studies. *NeuroImage*, 50(4):1376–1383, May 2010. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2010.01.061.

Jose Pinheiro and Douglas Bates. *Mixed-Effects Models in S and S-PLUS*. Statistics and Computing. Springer, 2000. ISBN 978-0-387-98957-0.

Øystein Sørensen, Kristine B. Walhovd, and Anders M. Fjell. A recipe for accurate estimation of lifespan brain trajectories, distinguishing longitudinal and cohort effects. *NeuroImage*, 226:117596, February 2021. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2020.117596.

S.N Wood. *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, second edition, 2017.