Marginal Bone Level on Teeth Implants [implant]

Assignment

Problem

The goal of the analysis is to model evolution of a marginal bone level (quality of a bone, variable `bl`) in a jaw under an installed tooth implant during the years after the installation of the implant. It should also be evaluated whether and how this evolution depends on the following factors: (i) gender (variables `gender` and `fgender`), (ii) age at implant installation (variable `age`), (iii) implant length (`implen`), (iv) position of the implant (variables `jawsite` and `fjawsite`): on the *maxillary* (upper) or *mandibular* (lower) jaw, (v) type of the implant (variables `proscon` and `fproscon`): *freestanding* or *connected*. The marginal bone level was measured at the time of the implant installation, the follow-up visits (variable `year`) were scheduled to happen every half a year with a total follow-up of at most 7 years.

Specifications

1. Find a suitable function describing how the expectation of marginal bone level varies with time since the installation of the implant.
2. Investigate whether the bone level and/or its association with time depends on the factors mentioned in section *Problem*. If there is a dependence on these factors, describe it.

Perform a concise descriptive analysis (descriptive tables, figures) and comment what can be learned from it. Formulate the model used to address the study questions (include the model formula, list the assumptions and evaluate their validity). When replying to Specification (ii), show estimated effects of important parameters with confidence intervals; include p-values for relevant hypotheses. Describe the methods used to obtain the main results. Prepare a **short** discussion of the analysis methods (their strengths and weaknesses) and the results.

Dataset The dataset can be downloaded from
https://www2.karlin.mff.cuni.cz/~komarek/vyuka/2021_22/nmst432/AdvRegr_6_implant.RData

The dataframe is called `implant`. It contains 3 496 longitudinal measurements of a marginal bone level (variable `bl`) related to 829 teeth implants (variable `impid`).

Remark. Even though in many cases different implants recorded in the dataset belong to one patient, treat (for the purpose of this problem) different implants (indicated by different values of a variable `impid`) as independently behaving units. Nevertheless, be aware of possible limitations of your analysis.

Variable list: See Table 1.

Variable coding table

Table 1: Variable coding table

Variable Name	Variable Label	Variable Coding
`impid`	Implant id	integer
`year`	Time since implant installation	years
`bl`	Marginal bone level	mm, non-positive value (dental convention)
`age`	Age of patient at implant installation	years
`implen`	Length of implant	mm
`proscon`	Type of implant	0 = *Freestanding*, 1 = *Connected*
`jawsite`	Position of implant	0 = *Maxilla* (upper jaw), 1 = *Mandible* (lower jaw)
`gender`	Gender of a patient	0 = *Male*, 1 = *Female*
`fproscon`	Type of implant	factor
`fjawsite`	Position of implant	factor
`fgender`	Gender of a patient	factor

Instructions, hints

This document was prepared using Sweave (Leisch, 2002) in R (R Core Team, 2022), version 4.2.0 (2022-04-22). Additionally, the contributed packages nlme (Pinheiro et al., 2022) and mixAK (Komárek, 2009) were used.

The rest of the document provides commented R code that provides some steps of the analyzis which finally leads to a solution to the Problem. Note that not all output is shown in the document below. It is assumed that you run the code by yourself, supplement it by additional commands if needed and use this document only as a guidance through the code and output (that you create).

Initial operations

```
> setwd("/home/komarek/teach/mff_2021/nmst432_AdvRegr/Problem_6/")
> #
> (load("AdvRegr_6_implant.RData"))
> #
> dim(implant)                              ## 3496, 11
> with(implant, length(unique(impid)))      ## 829
> #
> head(implant)
> summary(implant)
> with(implant, table(year))
> #
> summary(implant[, "age"])
> hist(implant[, "age"], xlab = "Age", ylab = "Density",
  +      main = "", prob = TRUE, col = "olivedrab3")
> #
> summary(implant[, "implen"])
> hist(implant[, "implen"], xlab = "Length of implant [mm]", ylab = "Density",
  +      main = "", prob = TRUE, col = "olivedrab3")
> table(implant[, "implen"], useNA = "ifany")
```

Packages, basic transformations

```
> library("mixAK")
> library("mffSM")
> library("nlme")
> library("splines")
> implant <- transform(implant,
  +    fyear = factor(year),
  +    year0 = 1 * (year == 0),
  +    fage  = cut(age, breaks = c(18, 45, 50, 55, 60, 80),
  +                labels = c("<=45", "45-50", "50-55", "55-60", ">60")),
  +    fimplen = cut(implen,
  +                  breaks = c(5, 9, 12.5, 19), labels = c("<10", "10-12", ">12")),
  +    lbl = log(1 - bl))
```

```
> #
> with(implant, table(fage, useNA = "ifany"))
> with(implant, table(fimplen, useNA = "ifany"))
> #
> levels(implant[, "fyear"])
> levels(implant[, "fimplen"])
> #
> YLAB <- "Marginal bone level [mm]"
```

Exploratory plots

```
> ### Boxplots of bl by year
> plot(bl ~ fyear, data = implant, ylab = YLAB, xlab = "Time [years]",
  +      col = "palegreen")
>           ### Does not look normal (given year)...
> #
> ### Distribution of bl by year
> par(mfrow = c(3, 5), mar = c(4, 3, 4, 1) + 0.1)
> for (yr in levels(implant[, "fyear"])){
  +   hist(subset(implant, fyear == yr)[, "bl"], prob = TRUE,
  +        xlab = "Marg. bone level", ylab = "Density",
  +        main = paste("Year ", yr, sep = ""), col = "salmon")
  + }
> par(mfrow = c(1, 1), mar = c(5, 4, 4, 1) + 0.1)
>           ### Does not look normal (given year)...
> #
> ### Boxplots of log(1 - bl) by year
> plot(lbl ~ fyear, data = implant,
  +      lab = "Log(1 - bone level)", xlab = "Time [years]",
  +      col = "palegreen")
>           ### Also not really normal...
> #
> ### Distribution of log(1 - bl) by year
> par(mfrow = c(3, 5), mar = c(4, 3, 4, 1) + 0.1)
> for (yr in levels(implant[, "fyear"])){
  +   hist(subset(implant, fyear == yr)[, "lbl"], prob = TRUE,
  +        xlab = "Log(1 - bone level)", ylab = "Density",
  +        main = paste("Year ", yr, sep = ""), col = "salmon")
  + }
> par(mfrow = c(1, 1), mar = c(5, 4, 4, 1) + 0.1)
>           ### Also not really normal...
> #
> ### Grouped data
> gimplant <- groupedData(bl ~ year | impid,
  +     outer = ~ age + implen + fproscon + fjawsite + fgender, data = implant)
> #plot(gimplant)       ## Don't do it, too many subjects...
> #
> summary(gimplant)
> #gsummary(gimplant)   ## Don't do it, too many subjects...
```

```
> #
> ### Random subset (for plotting)
> ID <- unique(implant[, "impid"])
> (N <- length(ID))
> #
> set.seed(19730911)
> subID <- sample(x = ID, size = 100, replace = FALSE)
> subID <- subID[order(subID)]
> print(subID)
> #
> gimplantSub <- groupedData(bl ~ year | impid,
  +     outer = ~ age + implen + fproscon + fjawsite + fgender + fage + fimplen,
  +     data = subset(implant, impid %in% subID))
> #
> plot(gimplantSub)                       ### not really informative
> plot(gimplantSub, outer = ~fproscon, key = FALSE, aspect = "fill")
> plot(gimplantSub, outer = ~fjawsite, key = FALSE, aspect = "fill")
> plot(gimplantSub, outer = ~fgender, key = FALSE, aspect = "fill")
> #
> plot(gimplantSub, outer = ~fage, key = FALSE, aspect = "fill")
> plot(gimplantSub, outer = ~fimplen, key = FALSE, aspect = "fill")
> #
> ### Plots using the mixAK package
> ### Profiles (extraction)
> ip <- getProfiles(implant, t = "year",
  +    y = c("bl", "lbl", "age", "implen", "fproscon", "fjawsite", "fgender",
  +          "fage", "fimplen"),
  +    id = "impid")
> #
> ### Profiles (plot)
> plotProfiles(ip, data = implant, var = "bl", tvar = "year",
  +       xlab = "Year", ylab = YLAB)
> plotProfiles(ip, data = implant, var = "bl", tvar = "year",
  +       xlab = "Year", ylab = YLAB, points = TRUE,
  +       pch = 23, col = "red3", bg = "salmon", lcol = "lightblue")
>            ## this one is not really useful here
> #
> plotProfiles(ip, data = implant, var = "bl", tvar = "year",
  +       xlab = "Year", ylab = "Log(1 - bone level)")
> #
> ### Profiles + selected highlighted
> (N <- length(ip))
> nSelect <- 5
> set.seed(20010911)
> (iSamp  <- sample.int(N, size = nSelect))
> iSamp <- iSamp[order(iSamp)]
> #
> plotProfiles(ip, data = implant, var = "bl", tvar = "year", xlab = "Year", ylab = YLAB)
> for (i in iSamp){
  +   idata <- ip[[i]]
  +   lines(bl ~ year, data = idata, col = "red3", lwd = 2)
```

```
  +   points(idata[, "year"], idata[, "bl"], pch = 23, col = "red3", bg = "salmon")
  + }
> #
> ### Profiles + by-year means
> (meanBL <- with(implant, tapply(bl, fyear, mean)))
> (Year <- as.numeric(names(meanBL)))
> #
> plotProfiles(ip, data = implant, var = "bl", tvar = "year", xlab = "Year", ylab = YLAB)
> lines(Year, meanBL, col = "red3", lwd = 3)
> points(Year, meanBL, col = "red3", bg = "salmon", pch = 23, cex = 2)
> #
> ### Profiles + by-year means categorized by factor outer covariates
> #variable <- "fgender"
> plotProfCat <- function(variable)
  + {
  +   nLev <- length(levels(implant[, variable]))
  +   if (nLev > 5) stop("not (yet) implemented for factors with more than five levels.")
  +   COL <- c("darkblue", "red3", "darkgreen", "violetred4", "orange")[1:nLev]
  +   BG <- c("cadetblue3", "lightsalmon", "palegreen", "violet", "lightgoldenrod1")[1:nLev]
  +   PCH <- (21:25)[1:nLev]
  +   names(COL) <- names(BG) <- names(PCH) <- levels(implant[, variable])
  +
  +   plotProfiles(ip, data = implant, var = "bl", tvar = "year", gvar = variable,
  +         xlab = "Year", ylab = YLAB, col = BG, auto.layout = FALSE)
  +   for (ll in 1:nLev){
  +     subimpl <- implant[implant[, variable] == levels(implant[, variable])[ll],]
  +     lmeanBL <- tapply(subimpl[, "bl"], subimpl[, "fyear"], mean)
  +     lines(as.numeric(names(lmeanBL)), lmeanBL, col = COL[ll], lwd = 2)
  +     points(as.numeric(names(lmeanBL)), lmeanBL, col = COL[ll], bg = BG[ll], pch = PCH[ll],
  +   }
  +   legend(0, ifelse(nLev > 2, -4.5, -5.5), legend = levels(implant[, variable]),
  +          lty = 1, lwd = 2, pch = PCH, pt.bg = BG, col = COL)
  +   title(main = variable)
  +
  +   return(invisible(variable))
  + }
> layout(matrix(c(1,1,2,2,3,3, 0,4,4,5,5,0), nrow = 2, byrow = TRUE))
> plotProfCat("fgender")
> plotProfCat("fjawsite")
> plotProfCat("fproscon")
> #
> plotProfCat("fage")
> plotProfCat("fimplen")
> par(mfrow = c(1, 1))
>
> ### What have you learned by now?
```

Marginal linear models

First, we will fit some linear models to check suitable *marginal* models. Even though, given quite clear exploratory part, fitting standard linear models could perhaps be omitted. In any case, all tests are only partly informative!

```
> ### Spline for piecewise linear
> Knots <- c(0, 0.5, 6.5, 7)
> nKn <- length(Knots)
> Degree <- 1
> yspl <- bs(implant[, "year"], knots = Knots[-c(1, length(Knots))],
  +            Boundary.knots = Knots[c(1, length(Knots))], degree = Degree,
  +            intercept = TRUE)
> #
>     ### Spline basis
> source("plotSplines.R")
> xgrid <- seq(0, 7, by = 0.5)
> plotSplineBasis(xgrid, Knots = Knots, Degree = Degree, Intercept = TRUE)
>   ### Saturated model
> fit0 <- lm(bl ~ fyear, data = implant)
> summary(fit0)
> plotLM(fit0)
>     ### Clearly not-normal...
>
>   ### Piecewise linear (changes at 0.5 and 6.5)
> fit1 <- lm(bl ~ yspl - 1, data = implant)
> summary(fit1)
> plotLM(fit1)
> #
> plotSpline(xgrid, coef(fit1), Knots = Knots, Degree = Degree, Intercept = TRUE)
> #
> plotProfiles(ip, data = implant, var = "bl", tvar = "year", xlab = "Year", ylab = YLAB)
> plotSpline(xgrid, coef(fit1), col = "red1", Knots = Knots, Degree = Degree,
  +            Intercept = TRUE, add = TRUE)
> #
>   ### Some value at 0, line from 0.5
> fit2 <- lm(bl ~ year0 + year, data = implant)
> summary(fit2)
> plotLM(fit2)
> #
>   ### Two constants (at 0 and then different one)
> fit3 <- lm(bl ~ year0, data = implant)
> summary(fit3)
> plotLM(fit3)
> #
>   ### Formal submodel tests are only very informative!!!
> anova(fit3, fit2, fit1, fit0)
> #
> anova(fit1, fit0)
> anova(fit2, fit0)
```

```
> anova(fit3, fit0)
>   ### Everything worse than saturated model.
>   ### But the p-values are most likely anti-conservative
>   ### (smaller than they should be).
> #
> deviance(fit3)
> deviance(fit2)
> deviance(fit1)
> deviance(fit0)
```

Linear models by implant while assuming common residual variability

---

```
> source("plotLMMres.R")
> #
>   ### Add spline basis for the piecewise linear model
>   ### to the data (lmList needs all variables used in the model
>   ### in one data.frame)
> implant <- transform(implant, s1 = yspl[,1], s2 = yspl[,2], s3 = yspl[,3], s4 = yspl[,4])
>   ### Saturated model
> #ifit0 <- lmList(bl ~ fyear | impid, data = implant)
>     ### not feasible (no residual variability left)
>
>   ### Piecewise linear with changes at 0.5 and 6.5
> ifit1 <- lmList(bl ~ s1 + s2 + s3 + s4 - 1 | impid, data = implant)
> print(ifit1)
> plot(ifit1)
> #help("residuals.lmList")
> sres1 <- residuals(ifit1, type = "pooled.pearson")
>               ## uses common sigma when doing standardization
> sfit1 <- fitted(ifit1)
> plot(sfit1, sres1, pch = 23, col = "darkblue", bg = "cadetblue3")
> lws1 <- lowess(sfit1, sres1)
> lines(lws1$x, lws1$y, col = "red2")
> #
> plotLMMres(ifit1, resid.type = "pooled.pearson")
>   ### Some value at 0, line from 0.5
> ifit2 <- lmList(bl ~ year0 + year | impid, data = implant)
> print(ifit2)
> plot(ifit2)
> plotLMMres(ifit2, resid.type = "pooled.pearson")
> pairs(ifit2)
>   ### Two constants (at 0 and then the other one)
> ifit3 <- lmList(bl ~ year0 | impid, data = implant)
> print(ifit3)
> plot(ifit3)
> plotLMMres(ifit3, resid.type = "pooled.pearson")
> pairs(ifit3)
> #
> be3 <- coef(ifit3)
```

```
> print(be3[1:20,])
> subset(implant, impid == 161)    ## reason for NA of estimated year0 coefficient?
> #
> be3all <- be3[!is.na(be3[,2]),]
> plot(be3all[,1], be3all[,2], xlab = "Intercept", ylab = "year0",
  +      pch = 21, col = "red3", bg = "cadetblue2")
> abline(a = 0, b = -1, col = "red3", lwd = 2)
> #
>   ### Intercept + year0 = Y_{i,0} --> do you know why?
> impidall <- as.numeric(rownames(be3all))
> bl0 <- subset(implant, impid %in% impidall & year == 0)[, "bl"]
> sumbe3all <- be3all[,1] + be3all[,2]
> plot(bl0, sumbe3all, xlab = "Y[i,0]", ylab = "Sum of coefs.",
  +      pch = 21, col = "red3", bg = "cadetblue2")
```

Linear mixed model

First, let us consider some marginal model for evolution over time and just random intercept.

```
>   ### Different means for each time, random intercept
>   ### ++++++++++++++++++++++++++++++++++++++++++++++++
> lmm0   <- lme(bl ~ fyear, random = ~ 1 | impid, data = implant)
> lmm0ml <- lme(bl ~ fyear, random = ~ 1 | impid, data = implant, method = "ML")
> #
> summary(lmm0)
> anova(lmm0)           ### mean over time is not constant (not really surprising...)
> plot(lmm0)
> plotLMMres(lmm0)
> #
>    ### Where is that strange line coming from?
>    ### Simliar reason as above...
> res0 <- residuals(lmm0, type = "pearson")
> fit0 <- fitted(lmm0)
> res0.0 <- res0[implant[, "bl"] == 0]       ### mostly (but not always) year = 0
> fit0.0 <- fit0[implant[, "bl"] == 0]
> res0.1 <- res0[implant[, "bl"] < 0]
> fit0.1 <- fit0[implant[, "bl"] < 0]
> par(mfrow = c(1, 2))
> plot(fit0.0, res0.0, pch = 23, col = "darkblue", bg = "cyan3",
  +      xlab = "Fitted", ylab = "Res", main = "bl = 0")
> plot(fit0.1, res0.1, pch = 23, col = "darkblue", bg = "cyan3",
  +      xlab = "Fitted", ylab = "Res", main = "bl < 0")
> par(mfrow = c(1, 1))
> #
>   ### Piecewise linear (changes at 0.5 and 6.5), random intercept
>   ### ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> lmm1 <- lme(bl ~ yspl - 1, random = ~ 1 | impid, data = implant)
> lmm1ml <- lme(bl ~ yspl - 1, random = ~ 1 | impid, data = implant, method = "ML")
> #
> summary(lmm1)
```

```
> anova(lmm1)            ### not really useful here...
> plotLMMres(lmm1)
> #
>   ### Some value at 0, line from 0.5, random intercept
>   ### ++++++++++++++++++++++++++++++++++++++++++++++++
> lmm2 <- lme(bl ~ year0 + year, random = ~ 1 | impid, data = implant)
> lmm2ml <- lme(bl ~ year0 + year, random = ~ 1 | impid, data = implant, method = "ML")
> #
> summary(lmm2)
> anova(lmm2)      ### !!! still type I (sequential) tests !!!
> #?anova.lme
> #
> plotLMMres(lmm2)
> #
> #
>   ### Two constants (at 0 and then different one), random intercept
>   ### ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
> lmm3 <- lme(bl ~ year0, random = ~1 | impid, data = implant)
> lmm3ml <- lme(bl ~ year0, random = ~1 | impid, data = implant, method = "ML")
> summary(lmm3)
> #
> #
>   ### Submodel tests, structured models versus saturated
>   ### (via likelihood-ratio - technically easier)
> anova(lmm1ml, lmm0ml)      ### piecewise linear (changes at 0.5 and 6.5) versus saturated
> anova(lmm2ml, lmm0ml)      ### some value at 0, line from 0.5 versus saturated
> anova(lmm3ml, lmm0ml)      ### two constants versus saturated
> anova(lmm3ml, lmm2ml)
> #
> anova(lmm2ml, lmm1ml)      ### piecewise linear (3 pieces) versus 0/line from 0.5
> #
> #  Constant and line versus two constants
> fixef(lmm2)
> fixef(lmm3)
> #
> anova(lmm3ml, lmm2ml)          ### LR test
> anova(lmm2ml, L = c(0, 0, 1))  ### Wald based on ML fit
> anova(lmm2, L = c(0, 0, 1))    ### Wald based on REML fit
> #
>   ## In this case, the Wald test was seen also from:
> summary(lmm2ml)
> summary(lmm2)
```

TASK

At the end of the exploratory part, we agreed that reasonable starting mixed model for further development is the following:

$$Y_{i,j} = \beta_0 + \beta_1\, t_{i,j} + \beta_2\, \mathbb{I}[t_{i,j} = 0] + b_{i,0} + b_{i,1}\, t_{i,j} + b_{i,2}\, \mathbb{I}[t_{i,j} = 0] + \varepsilon_{i,j} \tag{1}$$

with $t_{i,j}$ denoting time of measurement (since implant installation) and otherwise standard assumptions as in the lecture notes. Before going further, try to understand meaning of the model which roughly states that each subject has certain mean value of response at time 0 (equal to $\beta_0 + b_{i,0} + \beta_2 + b_{i,2}$) and subject-specific mean trajectory starting from time 0.5 (we have no data in between 0 and 0.5 so we cannot say anything about evolution here) follows a line whose intercept is $\beta_0 + b_{i,0}$ and slope is $\beta_1 + b_{i,1}$. As a homework, try to use either Wald tests or likelihood-ratio tests on submodels (models must be fitted using the ML and not REML method if LR tests are to be used) and explore how coefficients β_0, β_1 and β_2 depend on remaining covariates (gender, implant length etc.). That is, explore (by using "standard" model building strategy using above mentioned tests) whether it is useful to replace

- β_0 in the model formula by $\beta_0 + \beta_{0,1}\, \texttt{age}_i + \beta_{0,2}\, \texttt{gender}_i + \cdots$
  $\equiv$ which covariates should be included as main effects among the fixed effects of the linear mixed model;
- β_1 in the model formula by $\beta_1 + \beta_{1,1}\, \texttt{age}_i + \beta_{1,2}\, \texttt{gender}_i + \cdots$
  $\equiv$ which covariates should be included in interactions with time among the fixed effects of the linear mixed model;
- β_2 in the model formula by $\beta_2 + \beta_{2,1}\, \texttt{age}_i + \beta_{1,2}\, \texttt{gender}_i + \cdots$
  $\equiv$ which covariates should be included in interactions with $\mathbb{I}[t_{i,j} = 0]$ among the fixed effects of the linear mixed model.

Report the final model and perhaps state briefly how you derived it. On top of that, report your findings related to validity of the model assumptions (model diagnostics).

References

Komárek, A. (2009). A new R package for Bayesian estimation of multivariate normal mixtures allowing for selection of the number of components and interval-censored data. *Computational Statistics & Data Analysis*, **53**(12), 3932–3947. doi: 10.1016/j.csda.2009.05.006.

Leisch, F. (2002). Dynamic generation of statistical reports using literate data analysis. In Härdle, W. and Rönz, B., editors, *COMPSTAT 2002 – Proceedings in Computational Statistics*, pages 575–580, Heidelberg, 2002. Physica-Verlag.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., and R Core Team (2022). *nlme: Linear and Nonlinear Mixed Effects Models*. URL https://CRAN.R-project.org/package=nlme. R package version 3.1-155.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.