Signal Tandmobiel® Data
Time to caries analysis using `bayesBisurvreg`

Arnošt Komárek

June 29, 2005

This document describes the analysis of the Signal Tandmobiel® data presented in

Komárek, A. and Lesaffre, E.
Bayesian semiparametric accelerated failure time model for paired doubly-interval-censored data.

This article will be refered as Komárek and Lesaffre (2006b)and can be found in the `doc` directory of the package `bayesSurv` as `KomarekLesaffre2006b.pdf`. All R commands presented in this document are available in the same directory as `tandmobPA.R`.

The model considered in this document is the following:

$$\log(U_{i,l} - 5.0) = \boldsymbol{\delta}'\boldsymbol{z}_{i,l} + \zeta_{i,l}, \qquad i = 1, \ldots, N, \quad l = 1, 2, \tag{1}$$

$$\log(V_{i,l} - U_{i,l}) = \log(T_{i,l}) = \boldsymbol{\beta}'\boldsymbol{x}_{i,l} + \varepsilon_{i,l}, \qquad i = 1, \ldots, N, \quad l = 1, 2, \tag{2}$$

where $U_{i,l}$ is the emergence age of the lth tooth and $V_{i,l}$ is the age when the lth tooth was attacked by caries for the first time.

In the initial model (which is not shown in this document but it is discussed by Komárek and Lesaffre, 2006b), the covariate vectors were composed of

- $\boldsymbol{z}_{i,l}$ ... Girl, Girl:Maxilla;
- $\boldsymbol{x}_{i,l}$ ... Girl, Brush, Plaque.1, Plaque.2, Seal, Prim5d, Prim5m, Prim5f, Girl:Maxilla, Brush:Maxilla, Plaque.1:Maxilla, Plaque.2:Maxilla, Seal:Maxilla, Prim5d:Maxilla, Prim5m:Maxilla, Prim5f:Maxilla.

In the final model, the following covariates were used:

- $\boldsymbol{z}_{i,l}$ ... Girl;
- $\boldsymbol{x}_{i,l}$ ... Girl, Brush, Plaque, Seal, Prim5;

See below for the explanation of the covariates.

1 Initial operations

• Set the directories.

```
> anadir <- "/home/komari/win/work/papers/gsplineTand/anaDMFsymm/"
> chaindir.PA <- paste(anadir, "chains/modelPA", c("1646", "2636"),
+     sep = "")
> initdir.PA <- paste(anadir, "chains/end_PA", c("1646", "2636"),
+     sep = "")
> plotdir.PA <- paste(anadir, "summPlots/modelPA/", sep = "")
> resultdir <- paste(anadir, "results/", sep = "")
> predCurvesdir <- paste(resultdir, "predCurves/", sep = "")
> figuredir <- "/home/komari/win/work/papers/gsplineTand/RforCRAN/figuresPA/"
```

• We load the package and the data.

```
> library(bayesSurv)
> data(tandmobRoos)
```

In the following we create two separate datasets. One dataset for a pair of teeth 16 and 46 (right first molars) and one dataset for a pair of teeth 26 and 36 (left first molars). In Komárek and Lesaffre (2006b), only the analysis of the right first molars is shown. In each dataset we will have one row per tooth and only children who have all covariate information non-missing for both considered teeth are included.

• Separate datasets for pairs 16-46 and 26-36 from the original data:

```
> pair <- list(pair1 = c(16, 46), pair2 = c(26, 36))
> dmfp <- list()
> for (pii in 1:length(pair)) {
+     tt1 <- pair[[pii]][1]
+     tt2 <- pair[[pii]][2]
+     remove <- (is.na(tandmobRoos[, paste("FBEG.", tt1, sep = "")]) |
+         is.na(tandmobRoos[, paste("FBEG.", tt2, sep = "")]) |
+         (tandmobRoos[, paste("TOOTH.", tt1, sep = "")] == 0) |
+         (tandmobRoos[, paste("TOOTH.", tt2, sep = "")] == 0) |
+         is.na(tandmobRoos[, paste("T", tt1 + 39, "d", sep = "")]) |
+         is.na(tandmobRoos[, paste("T", tt1 + 39, "m", sep = "")]) |
+         is.na(tandmobRoos[, paste("T", tt1 + 39, "f", sep = "")]) |
+         is.na(tandmobRoos[, paste("T", tt1 + 39, "s", sep = "")]) |
+         is.na(tandmobRoos[, paste("T", tt2 + 39, "d", sep = "")]) |
+         is.na(tandmobRoos[, paste("T", tt2 + 39, "m", sep = "")]) |
+         is.na(tandmobRoos[, paste("T", tt2 + 39, "f", sep = "")]) |
+         is.na(tandmobRoos[, paste("T", tt2 + 39, "s", sep = "")]) |
+         is.na(tandmobRoos[, paste("SEAL.", tt1, sep = "")]) |
+         is.na(tandmobRoos[, paste("SEAL.", tt2, sep = "")]) |
+         is.na(tandmobRoos[, "FREQ.BR"]) | is.na(tandmobRoos[,
+         paste("PLAQUE.", tt1, ".1", sep = "")]) | is.na(tandmobRoos[,
+         paste("PLAQUE.", tt1, ".2", sep = "")]) | is.na(tandmobRoos[,
+         paste("PLAQUE.", tt2, ".1", sep = "")]) | is.na(tandmobRoos[,
+         paste("PLAQUE.", tt2, ".2", sep = "")]))
+     dmfp[[pii]] <- tandmobRoos[!remove, ]
+ }
> names(dmfp) <- c("1646", "2636")
> n.sample <- sapply(dmfp, nrow)
> print(n.sample)

1646 2636
2827 2843
```

• Create datasets with one row per tooth. Subtract also 5 from each time variable (time zero in our analysis was 5 years of age).

```
> time.0 <- 5
```

• Emergence times:

```
> Ebeg <- list()
> Eend <- list()
> for (pii in 1:length(pair)) {
+     tt1 <- pair[[pii]][1]
+     tt2 <- pair[[pii]][2]
+     Ebeg[[pii]] <- cbind(dmfp[[pii]][, paste("EBEG.", tt1, sep = "")],
+         dmfp[[pii]][, paste("EBEG.", tt2, sep = "")])
+     Eend[[pii]] <- cbind(dmfp[[pii]][, paste("EEND.", tt1, sep = "")],
+         dmfp[[pii]][, paste("EEND.", tt2, sep = "")])
+     Ebeg[[pii]] <- as.vector(t(Ebeg[[pii]])) - time.0
+     Eend[[pii]] <- as.vector(t(Eend[[pii]])) - time.0
+     Ebeg[[pii]][Ebeg[[pii]] <= 0] <- NA
+ }
```

• Caries times:

```
> Fbeg <- list()
> Fend <- list()
> for (pii in 1:length(pair)) {
+     tt1 <- pair[[pii]][1]
+     tt2 <- pair[[pii]][2]
+     Fbeg[[pii]] <- cbind(dmfp[[pii]][, paste("FBEG.", tt1, sep = "")],
+         dmfp[[pii]][, paste("FBEG.", tt2, sep = "")])
+     Fend[[pii]] <- cbind(dmfp[[pii]][, paste("FEND.", tt1, sep = "")],
+         dmfp[[pii]][, paste("FEND.", tt2, sep = "")])
+     Fbeg[[pii]] <- as.vector(t(Fbeg[[pii]])) - time.0
+     Fend[[pii]] <- as.vector(t(Fend[[pii]])) - time.0
+ }
```

• Covariates. These are:

- `Idnr`: identification number of a child;
- `Maxilla`: binary covariate, 1 = upper (maxillary) tooth, 0 = lower (mandibular) tooth;
- `Girl`: binary covariate, 1 = girl, 0 = boy;
- `Gender`: factor, girl/boy;
- `Brush`: binary covariate, 1 = brushing at least once a day, 0 = brushing less than once a day;
- `fBrush`: factor, daily = brushing at least once a day, not.daily = brushing less than once a day;
- `Plaque`: binary covariate, 1 = plaque either in pits and fissures or on total occlusal surface, 0 = no plaque;
- `Plaque.1`: dummy, 1 = plaque in pits and fissures, 0 = either no plaque or plaque on total occlusal surface;
- `Plaque.2`: dummy, 1 = plaque on total occlusal surface, 0 = either no plaque or plaque in pits and fissures;
- `fPlaque`: ordered factor, none = no plaque, pits.fiss = plaque in pits and fissures, total = plaque on total occlusal surface;

- `Seal`: binary covariate, 1 = pits and fissures sealing, 0 = no sealing;
- `fSeal`: factor, `yes` = pits and fissures sealing, `no` = no sealing;
- `Prim5`: binary covariate concerning the status of the adjacent primary second molar (tooth 55 is adjacent to 16, tooth 65 is adjacent to 26, tooth 75 is adjacent to 36, tooth 85 is adjacent to 46), 1 = either **d**ecayed or **m**issing due to carries or **f**illed, 0 = sound;
- `Prim5d`: dummy for the status of the adjacent primary second molar, 1 = **d**ecayed;
- `Prim5m`: dummy for the status of the adjacent primary second molar, 1 = **m**issing due to caries;
- `Prim5f`: dummy for the status of the adjacent primary second molar, 1 = **f**illed;
- `fPrim5`: factor for the status of the adjacent primary second molar, sound/decayed/filled/ missing.

```
> Idnr <- list()
> Maxilla <- list()
> Girl <- list()
> Gender <- list()
> Brush <- list()
> fBrush <- list()
> Plaque <- list()
> Plaque.1 <- list()
> Plaque.2 <- list()
> fPlaque <- list()
> Seal <- list()
> fSeal <- list()
> Prim5 <- list()
> Prim5d <- list()
> Prim5m <- list()
> Prim5f <- list()
> fPrim5 <- list()
> for (pii in 1:length(pair)) {
+     tt1 <- pair[[pii]][1]
+     tt2 <- pair[[pii]][2]
+     Idnr[[pii]] <- rep(dmfp[[pii]][, "IDNR"], rep(2, dim(dmfp[[pii]])[1]))
+     Maxilla[[pii]] <- rep(c(1, 0), dim(dmfp[[pii]])[1])
+     Girl[[pii]] <- rep(dmfp[[pii]][, "GIRL"], rep(2, dim(dmfp[[pii]])[1]))
+     Brush[[pii]] <- rep(dmfp[[pii]][, "FREQ.BR"], rep(2, dim(dmfp[[pii]])[1]))
+     Plaque.1[[pii]] <- as.vector(t(cbind(dmfp[[pii]][, paste("PLAQUE.",
+         tt1, ".1", sep = "")], dmfp[[pii]][, paste("PLAQUE.",
+         tt2, ".1", sep = "")])))
+     Plaque.2[[pii]] <- as.vector(t(cbind(dmfp[[pii]][, paste("PLAQUE.",
+         tt1, ".2", sep = "")], dmfp[[pii]][, paste("PLAQUE.",
+         tt2, ".2", sep = "")])))
+     Seal[[pii]] <- as.vector(t(cbind(dmfp[[pii]][, paste("SEAL.",
+         tt1, sep = "")], dmfp[[pii]][, paste("SEAL.", tt2, sep = "")])))
+     Prim5d[[pii]] <- as.vector(t(cbind(dmfp[[pii]][, paste("T",
+         tt1 + 39, "d", sep = "")], dmfp[[pii]][, paste("T", tt2 +
+         39, "d", sep = "")])))
+     Prim5m[[pii]] <- as.vector(t(cbind(dmfp[[pii]][, paste("T",
+         tt1 + 39, "m", sep = "")], dmfp[[pii]][, paste("T", tt2 +
+         39, "m", sep = "")])))
+     Prim5f[[pii]] <- as.vector(t(cbind(dmfp[[pii]][, paste("T",
+         tt1 + 39, "f", sep = "")], dmfp[[pii]][, paste("T", tt2 +
+         39, "f", sep = "")])))
+     Gender[[pii]] <- factor(1 * (Girl[[pii]] == 1), levels = 0:1,
+         labels = c("boy", "girl"))
+     fBrush[[pii]] <- factor(1 * (Brush[[pii]] == 1), levels = 0:1,
+         labels = c("not.daily", "daily"))
```

```
+     fPlaque[[pii]] <- ordered(0 * (Plaque.1[[pii]] == 0 & Plaque.2[[pii]] ==
+         0) + 1 * (Plaque.1[[pii]] == 1 & Plaque.2[[pii]] == 0) +
+         2 * (Plaque.1[[pii]] == 0 & Plaque.2[[pii]] == 1), levels = 0:2,
+         labels = c("none", "pits.fiss", "total"))
+     fSeal[[pii]] <- factor(1 * (Seal[[pii]] == 1), levels = 0:1,
+         labels = c("no", "yes"))
+     fPrim5[[pii]] <- factor(0 * (Prim5d[[pii]] == 0 & Prim5f[[pii]] ==
+         0 & Prim5m[[pii]] == 0) + 1 * (Prim5d[[pii]] == 1 & Prim5f[[pii]] ==
+         0 & Prim5m[[pii]] == 0) + 2 * (Prim5d[[pii]] == 0 & Prim5f[[pii]] ==
+         1 & Prim5m[[pii]] == 0) + 3 * (Prim5d[[pii]] == 0 & Prim5f[[pii]] ==
+         0 & Prim5m[[pii]] == 1), levels = 0:3, labels = c("sound",
+         "decayed", "filled", "missing"))
+     Plaque[[pii]] <- 0 * (fPlaque[[pii]] == "none") + 1 * (fPlaque[[pii]] ==
+         "pits.fiss" | fPlaque[[pii]] == "total")
+     Prim5[[pii]] <- 0 * (fPrim5[[pii]] == "sound") + 1 * (fPrim5[[pii]] ==
+         "decayed" | fPrim5[[pii]] == "filled" | fPrim5[[pii]] ==
+         "missing")
+ }
```

• Final data frames for the analysis:

```
> data.PA <- list()
> for (pii in 1:length(pair)) {
+     data.PA[[pii]] <- data.frame(Idnr = Idnr[[pii]], Ebeg = Ebeg[[pii]],
+         Eend = Eend[[pii]], Fbeg = Fbeg[[pii]], Fend = Fend[[pii]],
+         Maxilla = Maxilla[[pii]], Girl = Girl[[pii]], Gender = Gender[[pii]],
+         Brush = Brush[[pii]], fBrush = fBrush[[pii]], Plaque = Plaque[[pii]],
+         Plaque.1 = Plaque.1[[pii]], Plaque.2 = Plaque.2[[pii]],
+         fPlaque = fPlaque[[pii]], Seal = Seal[[pii]], fSeal = fSeal[[pii]],
+         Prim5 = Prim5[[pii]], Prim5d = Prim5d[[pii]], Prim5m = Prim5m[[pii]],
+         Prim5f = Prim5f[[pii]], fPrim5 = fPrim5[[pii]])
+ }
> names(data.PA) <- c("1646", "2636")
> rm(list = c("dmfp", "Idnr", "Ebeg", "Eend", "Fbeg", "Fend", "Maxilla",
+     "Girl", "Gender", "Brush", "fBrush", "Plaque", "Plaque.1",
+     "Plaque.2", "fPlaque", "Seal", "fSeal", "Prim5", "Prim5d",
+     "Prim5m", "Prim5f", "fPrim5"))
> rm(list = c("pii", "remove", "tt1", "tt2"))
```

• Show first few rows of the data frame for the left teeth 16 and 46.

```
> print(data.PA$"1646"[1:10, ])
```

```
   Idnr Ebeg Eend Fbeg Fend Maxilla Girl Gender Brush    fBrush Plaque Plaque.1
1     1  1.7  2.7  6.2   NA       1    1   girl     0 not.daily      0        0
2     1  1.7  2.7  6.2   NA       0    1   girl     0 not.daily      0        0
3     2   NA  2.4  2.4  3.4       1    0    boy     0 not.daily      0        0
4     2   NA  2.4  2.4  3.4       0    0    boy     0 not.daily      0        0
5     3  0.5  1.5  1.5  2.7       1    1   girl     0 not.daily      0        0
6     3   NA  1.5  5.5   NA       0    1   girl     0 not.daily      0        0
7     7   NA  1.9  6.3   NA       1    0    boy     1     daily      1        0
8     7   NA  1.9  6.3   NA       0    0    boy     1     daily      1        0
9     8   NA  2.1  6.5   NA       1    0    boy     0 not.daily      0        0
10    8   NA  2.1  6.5   NA       0    0    boy     0 not.daily      0        0
   Plaque.2 fPlaque Seal fSeal Prim5 Prim5d Prim5m Prim5f  fPrim5
1         0    none    1   yes     0      0      0      0   sound
2         0    none    1   yes     0      0      0      0   sound
3         0    none    0    no     0      0      0      0   sound
```

4	0	none	0	no	1	1	0	0	decayed
5	0	none	0	no	0	0	0	0	sound
6	0	none	0	no	0	0	0	0	sound
7	1	total	0	no	1	1	0	0	decayed
8	1	total	0	no	1	1	0	0	decayed
9	0	none	0	no	0	0	0	0	sound
10	0	none	0	no	0	0	0	0	sound

2 Initial estimates

In this section we compute initial estimates using the maximum-likelihood fits with correctly interval-censored emergence times and the mid-point imputed caries times. A parametric log-logistic AFT model will be used. We will then use these estimates to start the MCMC.

Note that we have to include the main effect of Maxilla in both the initial emergence and caries models. In the subsequent Bayesian model, the main effect of Maxilla is implicitely included in different intercepts for the upper and lower tooth.

• Initial fit for the emergence.

```
> fit.aft <- function(data) {
+     vbeg <- data[, "Ebeg"]
+     vend <- data[, "Eend"]
+     vbeg[vbeg <= 0] <- NA
+     tfit <- survreg(Surv(vbeg, vend, type = "interval2") ~ Maxilla +
+         Girl, dist = "loglogistic", data = data)
+     return(tfit)
+ }
> emerg.PA.aft <- list()
> emerg.PA.aft$"1646" <- fit.aft(data = data.PA$"1646")
> emerg.PA.aft$"2636" <- fit.aft(data = data.PA$"2636")
```

• Initial fit for caries.

```
> fit.aft2 <- function(data) {
+     vebeg <- data[, "Ebeg"]
+     veend <- data[, "Eend"]
+     vfbeg <- data[, "Fbeg"]
+     vfend <- data[, "Fend"]
+     vebeg[vebeg <= 0] <- NA
+     left <- is.na(vebeg)
+     interv <- !is.na(vebeg) & !is.na(veend)
+     right <- is.na(veend)
+     onset <- vebeg
+     onset[left] <- 0.5 * (0 + veend[left])
+     onset[right] <- vebeg[right]
+     onset[interv] <- 0.5 * (vebeg[interv] + veend[interv])
+     vfbeg2 <- vfbeg - onset
+     vfbeg2[vfbeg2 < 0] <- NA
+     vfend2 <- vfend - onset
+     tfit <- survreg(Surv(vfbeg2, vfend2, type = "interval2") ~
+         Maxilla + Girl + Brush + Plaque + Seal + Prim5, dist = "loglogistic",
+         data = data)
+     return(tfit)
+ }
> caries.PA.aft <- list()
> caries.PA.aft$"1646" <- fit.aft2(data = data.PA$"1646")
> caries.PA.aft$"2636" <- fit.aft2(data = data.PA$"2636")
```

• Results of the initial emergence fits.

```
> lapply(emerg.PA.aft, summary)

$"1646"

Call:
survreg(formula = Surv(vbeg, vend, type = "interval2") ~ Maxilla +
```

```
    Girl, data = data, dist = "loglogistic")
             Value Std. Error       z       p
(Intercept)  0.4259    0.00726   58.64 0.00000
Maxilla     -0.0224    0.00851   -2.63 0.00843
Girl        -0.0173    0.00849   -2.03 0.04216
Log(scale)  -2.2881    0.02233 -102.47 0.00000

Scale= 0.101

Log logistic distribution
Loglik(model)= -2067.6   Loglik(intercept only)= -2073.1
        Chisq= 11.01 on 2 degrees of freedom, p= 0.0041
Number of Newton-Raphson Iterations: 5
n= 5654


$"2636"

Call:
survreg(formula = Surv(vbeg, vend, type = "interval2") ~ Maxilla +
    Girl, data = data, dist = "loglogistic")
              Value Std. Error        z        p
(Intercept)  0.41540    0.00725   57.279 0.000000
Maxilla     -0.03204    0.00862   -3.718 0.000201
Girl        -0.00692    0.00859   -0.806 0.420314
Log(scale)  -2.28294    0.02248 -101.571 0.000000

Scale= 0.102

Log logistic distribution
Loglik(model)= -2037   Loglik(intercept only)= -2044.3
        Chisq= 14.57 on 2 degrees of freedom, p= 0.00069
Number of Newton-Raphson Iterations: 5
n= 5686
```

• Results of the initial caries fits.

```
> lapply(caries.PA.aft, summary)

$"1646"

Call:
survreg(formula = Surv(vfbeg2, vfend2, type = "interval2") ~
    Maxilla + Girl + Brush + Plaque + Seal + Prim5, data = data,
    dist = "loglogistic")
              Value Std. Error       z         p
(Intercept)  2.1913     0.0574  38.200  0.00e+00
Maxilla      0.0106     0.0362   0.292  7.70e-01
Girl        -0.0712     0.0363  -1.961  4.99e-02
Brush        0.3185     0.0484   6.583  4.62e-11
Plaque      -0.2046     0.0395  -5.176  2.27e-07
Seal         0.0438     0.0412   1.062  2.88e-01
Prim5       -0.6108     0.0396 -15.441  8.66e-54
Log(scale)  -0.5956     0.0253 -23.557 1.07e-122

Scale= 0.551

Log logistic distribution
Loglik(model)= -4768.7   Loglik(intercept only)= -4945.5
```

```
        Chisq= 353.73 on 6 degrees of freedom, p= 0
Number of Newton-Raphson Iterations: 4
n= 5654


$"2636"

Call:
survreg(formula = Surv(vfbeg2, vfend2, type = "interval2") ~
    Maxilla + Girl + Brush + Plaque + Seal + Prim5, data = data,
    dist = "loglogistic")
              Value Std. Error       z         p
(Intercept)  2.1709     0.0555  39.127  0.00e+00
Maxilla      0.0347     0.0356   0.974  3.30e-01
Girl        -0.0696     0.0358  -1.944  5.19e-02
Brush        0.3379     0.0473   7.140  9.34e-13
Plaque      -0.2549     0.0388  -6.571  4.99e-11
Seal         0.0729     0.0409   1.785  7.43e-02
Prim5       -0.6160     0.0390 -15.803  2.95e-56
Log(scale)  -0.6225     0.0256 -24.304 1.78e-130

Scale= 0.537

Log logistic distribution
Loglik(model)= -4661.1   Loglik(intercept only)= -4852.3
        Chisq= 382.41 on 6 degrees of freedom, p= 0
Number of Newton-Raphson Iterations: 4
n= 5686
```

3 Prior distributions

In this section we specify the prior distributions.

3.1 Priors for the smoothed error distributions

In our model, we have two penalized Gaussian mixtures (G-splines) included. The first G-spline defines a distribution of the emergence error term $g_\zeta(\zeta_1, \zeta_2)$ and the second one a distribution of the time-to-caries error term $g_\varepsilon(\varepsilon_1, \varepsilon_2)$. Since we will not use any informative priors, the prior specifications for the two G-splines will be the same, namely

- $K_1 = K_2 = 15$;
- The middle knot is really in the middle, i.e. it is $\mu_{(0,0)}$;
- We use the univariate conditional autoregression prior (`uniCAR`) for the transformed mixture weights $\mathbb{A}$ with the differences of the 3rd order;
- The smoothing hyperparameters λ_1 and λ_2 are not forced to be equal;
- The distance between the two knots in each margin is equal to 1.5 times the basis standard deviation σ_1 and σ_2, respectively, i.e. the basis standard deviation is equal to 2/3 times the distance between the two knots;
- Prior distribution for the smoothing hyperparameters λ_1 and λ_2 is Gamma(1, 0.005);
- Prior distribution for the intercept terms α_1 and α_2 is $\mathcal{N}(0, 100)$;
- Prior distribution for the scale parameters τ_1 and τ_2 is Gamma(1, 0.005) on τ^{-2};
- Position of the middle knot (γ_1, γ_2) is fixed (this corresponds to `specification = 2`);
- The basis standard deviations σ_1 and σ_2 are fixed (this corresponds to `specification = 2`).

```
> prior.PA.gspl.emerg <- list(specification = 2, K = c(15, 15),
+     izero = c(0, 0), neighbor.system = "uniCAR", order = 3, equal.lambda = FALSE,
+     c4delta = c(1.5, 1.5), prior.lambda = c("gamma", "gamma"),
+     prior.intercept = c("normal", "normal"), prior.scale = c("gamma",
+         "gamma"), prior.gamma = c("fixed", "fixed"), prior.sigma = c("fixed",
+         "fixed"), shape.lambda = c(1, 1), rate.lambda = c(0.005,
+         0.005), mean.intercept = c(0, 0), var.intercept = c(100,
+         100), shape.scale = c(1, 1), rate.scale = c(0.005, 0.005))
> prior.PA.gspl.caries <- prior.PA.gspl.emerg
```

3.2 Priors for the regression parameters

For all regression parameters we use $\mathcal{N}(0, 100)$ priors.

```
> prior.PA.beta.emerg <- list(mean.prior = 0, var.prior = 100)
> prior.PA.beta.caries <- list(mean.prior = rep(0, 5), var.prior = rep(100,
+     5))
```

4 Initial values

In this section we specify the initial values for the MCMC sampling. Partially, we use the values obtained in Section 2. We specify separately initials for the right pair of teeth (16 and 46) and the left pair of teeth (26 and 36).

4.1 Emergence part of the model

The following initial values are used:

- $\lambda_1^\zeta = \lambda_2^\zeta = 3\,000$;
- $\alpha_1^\zeta = 0.40$, $\alpha_2^\zeta = 0.42$;
- $\tau_1^\zeta = \tau_2^\zeta = 0.10$;
- $\gamma_1^\zeta = \gamma_2^\zeta = 0$, i.e. the middle knot $\mu_{(0,\,0)}^\zeta$ is equal to (0, 0). Note that this value remains the same during the whole MCMC;
- $\sigma_1^\zeta = \sigma_2^\zeta = 0.20$ (the basis standard deviations). These values remain the same during the whole MCMC as well;
- $\boldsymbol{\delta} = (\delta_1) = (-0.01)$.

```
> iinit.PA.emerg <- list(lambda = c(3000, 3000), intercept = c(0.4,
+     0.42), scale = rep(0.1, 2), gamma = c(0, 0), sigma = c(0.2,
+     0.2), beta = -0.01)
> init.PA.emerg <- list(iinit.PA.emerg, iinit.PA.emerg)
> names(init.PA.emerg) <- c("1646", "2636")
```

4.2 Caries part of the model

The following initial values are used:

- $\lambda_1^\varepsilon = \lambda_2^\varepsilon = 3\,000$;
- $\alpha_1^\varepsilon = \alpha_2^\varepsilon = 2.20$;
- $\tau_1^\varepsilon = \tau_2^\varepsilon = 0.55$;
- $\gamma_1^\varepsilon = \gamma_2^\varepsilon = 0$, i.e. the middle knot $\mu_{(0,\,0)}^\varepsilon$ is equal to (0, 0). Note that this value remains the same during the whole MCMC;
- $\sigma_1^\varepsilon = \sigma_2^\varepsilon = 0.20$ (the basis standard deviations). These values remain the same during the whole MCMC as well;
- $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_5)' = (-0.07,\, 0.32,\, -0.20,\, 0.04,\, -0.61)'$.

```
> iinit.PA.caries <- list(lambda = c(3000, 3000), intercept = c(2.2,
+     2.2), scale = c(0.55, 0.55), gamma = c(0, 0), sigma = c(0.2,
+     0.2), beta = c(-0.07, 0.32, -0.2, 0.04, -0.61))
> init.PA.caries <- list(iinit.PA.caries, iinit.PA.caries)
> names(init.PA.caries) <- c("1646", "2636")
```

4.3 Read initial values from files

If we wish to restart the MCMC from some specific point we can read the initial values from files. In the following, the initials are taken from the first row of each file (after skiping possible header). It is assumed that the files for the analysis of the pair of teeth 16 and 46 are in the directory `initdir.PA[1]` and the files for the analysis of the pair of teeth 26 and 36 in the directory `initdir.PA[2]`. Note how the function `vecr2matr` is used to transform stored allocation variables.

```
> init.PA.emerg <- list()
> init.PA.caries <- list()
> for (k in 1:2) {
+     init.PA.emerg[[k]] <- list()
+     init.PA.emerg[[k]]$iter <- scan(paste(initdir.PA[k], "/iteration.sim",
+         sep = ""), skip = 1)
+     init.PA.emerg[[k]]$lambda <- scan(paste(initdir.PA[k], "/lambda.sim",
+         sep = ""), skip = 1)
+     init.PA.gspline <- scan(paste(initdir.PA[k], "/gspline.sim",
+         sep = ""), skip = 1)
+     init.PA.emerg[[k]]$gamma <- init.PA.gspline[1:2]
+     init.PA.emerg[[k]]$sigma <- init.PA.gspline[3:4]
+     init.PA.emerg[[k]]$intercept <- init.PA.gspline[7:8]
+     init.PA.emerg[[k]]$scale <- init.PA.gspline[9:10]
+     init.PA.emerg[[k]]$beta <- scan(paste(initdir.PA[k], "/beta.sim",
+         sep = ""), skip = 1)
+     init.PA.emerg[[k]]$a <- scan(paste(initdir.PA[k], "/mlogweight.sim",
+         sep = ""), skip = 0)
+     init.PA.emerg[[k]]$y <- matrix(scan(paste(initdir.PA[k],
+         "/Y.sim", sep = ""), skip = 0), ncol = 2, byrow = TRUE)
+     init.PA.r <- scan(paste(initdir.PA[k], "/r.sim", sep = ""),
+         skip = 0)
+     init.PA.emerg[[k]]$r <- vecr2matr(init.PA.r, prior.PA.gspl.emerg$K)
+     init.PA.caries[[k]] <- list()
+     init.PA.caries[[k]]$iter <- scan(paste(initdir.PA[k], "/iteration.sim",
+         sep = ""), skip = 1)
+     init.PA.caries[[k]]$lambda <- scan(paste(initdir.PA[k], "/lambda_2.sim",
+         sep = ""), skip = 1)
+     init.PA.gspline <- scan(paste(initdir.PA[k], "/gspline_2.sim",
+         sep = ""), skip = 1)
+     init.PA.caries[[k]]$gamma <- init.PA.gspline[1:2]
+     init.PA.caries[[k]]$sigma <- init.PA.gspline[3:4]
+     init.PA.caries[[k]]$intercept <- init.PA.gspline[7:8]
+     init.PA.caries[[k]]$scale <- init.PA.gspline[9:10]
+     init.PA.caries[[k]]$beta <- scan(paste(initdir.PA[k], "/beta_2.sim",
+         sep = ""), skip = 1)
+     init.PA.caries[[k]]$a <- scan(paste(initdir.PA[k], "/mlogweight_2.sim",
+         sep = ""), skip = 0)
+     init.PA.caries[[k]]$y <- matrix(scan(paste(initdir.PA[k],
+         "/Y_2.sim", sep = ""), skip = 0), ncol = 2, byrow = TRUE)
+     init.PA.r <- scan(paste(initdir.PA[k], "/r_2.sim", sep = ""),
+         skip = 0)
+     init.PA.caries[[k]]$r <- vecr2matr(init.PA.r, prior.PA.gspl.caries$K)
+ }
> names(init.PA.emerg) <- c("1646", "2636")
> names(init.PA.caries) <- c("1646", "2636")
```

• Show the initial values for the emergence process for the model with the right teeth (16 and 46) using the standard function `str`.

```
> str(init.PA.emerg$"1646")
```

```
List of 10
 $ iter     : num 250000
 $ lambda   : num [1:2] 47236 45559
 $ gamma    : num [1:2] 0 0
 $ sigma    : num [1:2] 0.2 0.2
 $ intercept: num [1:2] 0.630 0.638
 $ scale    : num [1:2] 0.104 0.102
 $ beta     : num -0.0219
 $ a        : num [1:961] 1.65 1.67 1.68 1.69 1.70 ...
 $ y        : num [1:2827, 1:2] 0.829 0.495 0.145 0.277 0.399 ...
 $ r        : num [1:2827, 1:2] 7 -4 -14 -11 -8 -8 0 -15 -10 -13 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "r1" "r2"
```

• Show the initial values for the caries process for the model with the right teeth (16 and 46) using the standard function `str`.

```
> str(init.PA.caries$"1646")
```

```
List of 10
 $ iter     : num 250000
 $ lambda   : num [1:2] 45937 43756
 $ gamma    : num [1:2] 0 0
 $ sigma    : num [1:2] 0.2 0.2
 $ intercept: num [1:2] 6.69 6.31
 $ scale    : num [1:2] 1.39 1.33
 $ beta     : num [1:5]  0.0550  0.1167 -0.2292  0.0782 -0.5739
 $ a        : num [1:961] 2.81 2.96 3.07 3.14 3.17 ...
 $ y        : num [1:2827, 1:2] 5.406 0.413 0.384 1.608 2.795 ...
 $ r        : num [1:2827, 1:2] -4 -15 -14 -10 -9 -11 -6 -13 -8 -5 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "r1" "r2"
```

5 MCMC sampling – function `bayesBisurvreg`

In this section we start the MCMC sampling (separately for each pair of teeth). The chains for the right pair of teeth (16 and 46) are stored in the directory `chaindir.PA[1]`, the chains for the left pair of teeth (26 and 36) are stored in the directory `chaindir.PA[2]`.

• Specify the length of the simulation.

```
> nsimul.PA <- list(niter = 250000, nthin = 3, nburn = 225000,
+     nwrite = 100)
```

• Sample.

```
> sides <- c("RIGHT", "LEFT")
> sample.PA <- list()
> for (k in 1:2) {
+     cat("\nPerforming ", sides[k], " part of the mouth\n", sep = "")
+     cat("=============================================\n")
+     data.now <- data.PA[[k]]
+     sample.PA[[k]] <- bayesBisurvreg(formula = Surv(Ebeg, Eend,
+         type = "interval2") ~ Girl + cluster(Idnr), formula2 = Surv(Fbeg,
+         Fend, type = "interval2") ~ Girl + Brush + Plaque + Seal +
+         Prim5 + cluster(Idnr), onlyX = FALSE, dir = chaindir.PA[k],
+         nsimul = nsimul.PA, prior = prior.PA.gspl.emerg, prior2 = prior.PA.gspl.caries,
+         prior.beta = prior.PA.beta.emerg, prior.beta2 = prior.PA.beta.caries,
+         init = init.PA.emerg[[k]], init2 = init.PA.caries[[k]],
+         store = list(a = FALSE, a2 = FALSE), data = data.now)
+ }
```

6 Posterior predictive survivor, density and hazard – function predictive2

Once the MCMC sampling is finished we can compute predictive survivor density and hazard functions for some specific combinations of covariates. In this section we show how the predictive quantities can be computed for the caries part of the model.

• Specify whether some sampled values should be skipped, how often will the information concerning the progress of computation be print on the screen. Further specify the grid of values in which the predictive quantities will be computed, specify probabilities of quantiles we want to compute.

```
> skip <- 0
> nwrite <- 5000
> pred.grid <- c(seq(0.1, 1.5, by = 0.05), seq(1.65, 6, by = 0.15))
> quants <- c(0.025, 0.5, 0.975)
> nquant <- length(quants)
```

6.1 Combinations of covariates

• Specify the combination of covariates for which the predictive quantities for each tooth will be computed.

```
> pGender <- factor(c("boy", "girl"))
> pBrush <- factor(c("not.daily", "daily"))
> pSeal <- factor(c("no", "yes"))
> pPlaque <- factor(c("none", "present"))
> pPrim5 <- factor(c("sound", "dmf"))
> pdata <- expand.grid(ffPrim5 = pPrim5, ffPlaque = pPlaque, fSeal = pSeal,
+     fBrush = pBrush, Gender = pGender)
> pdata$Prim5 <- 1 * (pdata$ffPrim5 == "dmf")
> pdata$Plaque <- 1 * (pdata$ffPlaque == "present")
> pdata$Seal <- 1 * (pdata$fSeal == "yes")
> pdata$Brush <- 1 * (pdata$fBrush == "daily")
> pdata$Girl <- 1 * (pdata$Gender == "girl")
> ncov <- nrow(pdata)
> pred.data <- data.frame(Idnr = 1:(2 * ncov), Maxilla = c(rep(1,
+     ncov), rep(0, ncov)), Dimension = c(rep(1, ncov), rep(2,
+     ncov)))
> pred.data <- cbind(pred.data, rbind(pdata, pdata))
```

• Look at the data frame we use for the prediction. Note how the variable `Dimension` which actually corresponds to the variable `Maxilla` is specified.

```
> print(pred.data)
```

```
   Idnr Maxilla Dimension ffPrim5 ffPlaque fSeal    fBrush Gender Prim5 Plaque
1     1       1         1   sound     none    no not.daily    boy     0      0
2     2       1         1     dmf     none    no not.daily    boy     1      0
3     3       1         1   sound  present    no not.daily    boy     0      1
4     4       1         1     dmf  present    no not.daily    boy     1      1
5     5       1         1   sound     none   yes not.daily    boy     0      0
6     6       1         1     dmf     none   yes not.daily    boy     1      0
7     7       1         1   sound  present   yes not.daily    boy     0      1
8     8       1         1     dmf  present   yes not.daily    boy     1      1
9     9       1         1   sound     none    no     daily    boy     0      0
10   10       1         1     dmf     none    no     daily    boy     1      0
11   11       1         1   sound  present    no     daily    boy     0      1
```

```
12   12       1         1     dmf  present    no     daily    boy     1       1
13   13       1         1   sound     none   yes     daily    boy     0       0
14   14       1         1     dmf     none   yes     daily    boy     1       0
15   15       1         1   sound  present   yes     daily    boy     0       1
16   16       1         1     dmf  present   yes     daily    boy     1       1
17   17       1         1   sound     none    no not.daily   girl     0       0
18   18       1         1     dmf     none    no not.daily   girl     1       0
19   19       1         1   sound  present    no not.daily   girl     0       1
20   20       1         1     dmf  present    no not.daily   girl     1       1
21   21       1         1   sound     none   yes not.daily   girl     0       0
22   22       1         1     dmf     none   yes not.daily   girl     1       0
23   23       1         1   sound  present   yes not.daily   girl     0       1
24   24       1         1     dmf  present   yes not.daily   girl     1       1
25   25       1         1   sound     none    no     daily   girl     0       0
26   26       1         1     dmf     none    no     daily   girl     1       0
27   27       1         1   sound  present    no     daily   girl     0       1
28   28       1         1     dmf  present    no     daily   girl     1       1
29   29       1         1   sound     none   yes     daily   girl     0       0
30   30       1         1     dmf     none   yes     daily   girl     1       0
31   31       1         1   sound  present   yes     daily   girl     0       1
32   32       1         1     dmf  present   yes     daily   girl     1       1
33   33       0         2   sound     none    no not.daily    boy     0       0
34   34       0         2     dmf     none    no not.daily    boy     1       0
35   35       0         2   sound  present    no not.daily    boy     0       1
36   36       0         2     dmf  present    no not.daily    boy     1       1
37   37       0         2   sound     none   yes not.daily    boy     0       0
38   38       0         2     dmf     none   yes not.daily    boy     1       0
39   39       0         2   sound  present   yes not.daily    boy     0       1
40   40       0         2     dmf  present   yes not.daily    boy     1       1
41   41       0         2   sound     none    no     daily    boy     0       0
42   42       0         2     dmf     none    no     daily    boy     1       0
43   43       0         2   sound  present    no     daily    boy     0       1
44   44       0         2     dmf  present    no     daily    boy     1       1
45   45       0         2   sound     none   yes     daily    boy     0       0
46   46       0         2     dmf     none   yes     daily    boy     1       0
47   47       0         2   sound  present   yes     daily    boy     0       1
48   48       0         2     dmf  present   yes     daily    boy     1       1
49   49       0         2   sound     none    no not.daily   girl     0       0
50   50       0         2     dmf     none    no not.daily   girl     1       0
51   51       0         2   sound  present    no not.daily   girl     0       1
52   52       0         2     dmf  present    no not.daily   girl     1       1
53   53       0         2   sound     none   yes not.daily   girl     0       0
54   54       0         2     dmf     none   yes not.daily   girl     1       0
55   55       0         2   sound  present   yes not.daily   girl     0       1
56   56       0         2     dmf  present   yes not.daily   girl     1       1
57   57       0         2   sound     none    no     daily   girl     0       0
58   58       0         2     dmf     none    no     daily   girl     1       0
59   59       0         2   sound  present    no     daily   girl     0       1
60   60       0         2     dmf  present    no     daily   girl     1       1
61   61       0         2   sound     none   yes     daily   girl     0       0
62   62       0         2     dmf     none   yes     daily   girl     1       0
63   63       0         2   sound  present   yes     daily   girl     0       1
64   64       0         2     dmf  present   yes     daily   girl     1       1
   Seal Brush Girl
1     0     0    0
2     0     0    0
3     0     0    0
4     0     0    0
5     1     0    0
```

6	1	0	0
7	1	0	0
8	1	0	0
9	0	1	0
10	0	1	0
11	0	1	0
12	0	1	0
13	1	1	0
14	1	1	0
15	1	1	0
16	1	1	0
17	0	0	1
18	0	0	1
19	0	0	1
20	0	0	1
21	1	0	1
22	1	0	1
23	1	0	1
24	1	0	1
25	0	1	1
26	0	1	1
27	0	1	1
28	0	1	1
29	1	1	1
30	1	1	1
31	1	1	1
32	1	1	1
33	0	0	0
34	0	0	0
35	0	0	0
36	0	0	0
37	1	0	0
38	1	0	0
39	1	0	0
40	1	0	0
41	0	1	0
42	0	1	0
43	0	1	0
44	0	1	0
45	1	1	0
46	1	1	0
47	1	1	0
48	1	1	0
49	0	0	1
50	0	0	1
51	0	0	1
52	0	0	1
53	1	0	1
54	1	0	1
55	1	0	1
56	1	0	1
57	0	1	1
58	0	1	1
59	0	1	1
60	0	1	1
61	1	1	1
62	1	1	1
63	1	1	1
64	1	1	1

6.2 Compute predictive quantities

• To avoid problems with memory (when the quantiles are asked, all predictive quantities during all the MCMC iterations must be at some moment stored in the memory which might often kill R due to memory problems), we compute predictive functions always only for two combinations of covariates at once and then put the results together.

• Here we define the combinations of covariates for which the predictive curves will be computed at once.

```
> side <- c("RIGHT", "LEFT")
> jaw <- c("MAXILLARY", "MANDIBULAR")
> start <- list(maxilla = seq(1, ncov, by = 2), mandible = seq(ncov +
+     1, 2 * ncov, by = 2))
> end <- list(maxilla = seq(2, ncov, by = 2), mandible = seq(ncov +
+     2, 2 * ncov, by = 2))
```

• Compute the predictive quantities using the function `predictive2`. The loop over `k` is the loop over the right and left pair of teeth, the loop over `jj` is the loop over the upper (maxillary) and lower (mandibular) tooth. The loop over `ii` is the loop over the sets of covariate combinations which are processed at once.

The list `pred.PA` will have four components (one per tooth) and each component will again be the list with each component corresponding to the set of covariate combinations that were processed at once.

```
> pred.PA <- list()
> for (k in 1:2) {
+     cat("Performing ", side[k], " side\n", sep = "")
+     for (jj in 1:2) {
+         cat("Performing ", jaw[jj], " jaw\n", sep = "")
+         pred.PA[[(k - 1) * 2 + jj]] <- list()
+         for (ii in 1:length(start[[jj]])) {
+             cat("Performing the covariate set number ", ii, "\n",
+                 sep = "")
+             pdata.now <- pred.data[start[[jj]][ii]:end[[jj]][ii],
+                 ]
+             nr <- nrow(pdata.now)
+             pred.PA[[(k - 1) * 2 + jj]][[ii]] <- predictive2(Surv(rep(1,
+                 nr), rep(1, nr)) ~ Girl + Brush + Plaque + Seal +
+                 Prim5 + cluster(Idnr), obs.dim = pdata.now$Dimension,
+                 grid = pred.grid, data = pdata.now, Gspline = list(dim = 2,
+                   K = c(15, 15)), quantile = quants, skip = skip,
+                 by = 1, nwrite = nwrite, only.aver = FALSE, predict = list(density = TRUE,
+                   Surv = TRUE, hazard = TRUE), dir = chaindir.PA[k],
+                 extens = "_2")
+         }
+     }
+ }
> tnames <- c("16", "46", "26", "36")
> names(pred.PA) <- tnames
```

• With the following code we create the list `predd.PA` with four components (one per tooth). Each of its components will join the results for all sets of covariates that were separately processed. At the end we assign `predd.PA` to `pred.PA`.

```
> predd.PA <- list()
> for (tt in 1:4) {
+     predd.PA[[tt]] <- list()
+     predd.PA[[tt]]$grid <- pred.PA[[tt]][[1]]$grid
+     predd.PA[[tt]]$Surv <- pred.PA[[tt]][[1]]$Surv
```

```
+     predd.PA[[tt]]$hazard <- pred.PA[[tt]][[1]]$hazard
+     predd.PA[[tt]]$density <- pred.PA[[tt]][[1]]$density
+     predd.PA[[tt]]$quant.Surv <- pred.PA[[tt]][[1]]$quant.Surv
+     predd.PA[[tt]]$quant.hazard <- pred.PA[[tt]][[1]]$quant.hazard
+     predd.PA[[tt]]$quant.density <- pred.PA[[tt]][[1]]$quant.density
+     for (ii in 2:length(start[[1]])) {
+         predd.PA[[tt]]$Surv <- rbind(predd.PA[[tt]]$Surv, pred.PA[[tt]][[ii]]$Surv)
+         predd.PA[[tt]]$hazard <- rbind(predd.PA[[tt]]$hazard,
+             pred.PA[[tt]][[ii]]$hazard)
+         predd.PA[[tt]]$density <- rbind(predd.PA[[tt]]$density,
+             pred.PA[[tt]][[ii]]$density)
+         templ <- length(predd.PA[[tt]]$quant.Surv)
+         for (ij in 1:length(pred.PA[[tt]][[ii]]$quant.Surv)) {
+             predd.PA[[tt]]$quant.Surv[[templ + ij]] <- pred.PA[[tt]][[ii]]$quant.Surv[[ij]]
+             predd.PA[[tt]]$quant.hazard[[templ + ij]] <- pred.PA[[tt]][[ii]]$quant.hazard[[ij]]
+             predd.PA[[tt]]$quant.density[[templ + ij]] <- pred.PA[[tt]][[ii]]$quant.density[[ij]]
+         }
+     }
+ }
> names(predd.PA) <- tnames
> pred.PA <- predd.PA
> rm(list = "predd.PA")
```

6.3 Store predictive quantities in files

We can also store predictive quantities in files for future use. We store them in the directory `predCurvesdir`.

• Posterior predictive means. In each file the first row is the grid in which the predictive functions are evaluated, the 2nd to 33rd rows are predictive functions for each combination of covariates specified by either `pred.data[1:32,]` or by `pred.data[33:64]`. We create one file per tooth.

```
> for (tt in 1:4) {
+     sink(paste(predCurvesdir, "predDensCaries.", tnames[tt],
+         ".PA.", "dat", sep = ""))
+     cat(pred.PA[[tt]]$grid, "\n", sep = "   ")
+     for (j in 1:ncov) cat(pred.PA[[tt]]$density[j, ], "\n", sep = "   ")
+     sink()
+     sink(paste(predCurvesdir, "predSurvCaries.", tnames[tt],
+         ".PA.", "dat", sep = ""))
+     cat(pred.PA[[tt]]$grid, "\n", sep = "   ")
+     for (j in 1:ncov) cat(pred.PA[[tt]]$Surv[j, ], "\n", sep = "   ")
+     sink()
+     sink(paste(predCurvesdir, "predhazardCaries.", tnames[tt],
+         ".PA.", "dat", sep = ""))
+     cat(pred.PA[[tt]]$grid, "\n", sep = "   ")
+     for (j in 1:ncov) cat(pred.PA[[tt]]$hazard[j, ], "\n", sep = "   ")
+     sink()
+ }
```

• Posterior predictive quantiles. In each file the first row is the grid in which the predictive functions are evaluated, the 2nd to 4th row are 2.5%, 50% and 97.5% quantile for the first combination of covariates, the 5th to 7th row are 2.5%, 50% and 97.5% quantile for the second combination of covariates etc. Again, one file per tooth is created.

```
> for (tt in 1:4) {
+     sink(paste(predCurvesdir, "quantDensCaries.", tnames[tt],
+         ".PA.", "dat", sep = ""))
+     cat(pred.PA[[tt]]$grid, "\n", sep = "   ")
```

```
+     for (j in 1:ncov) {
+         for (jj in 1:nquant) cat(pred.PA[[tt]]$quant.density[[j]][jj,
+             ], "\n", sep = "   ")
+     }
+     sink()
+     sink(paste(predCurvesdir, "quantSurvCaries.", tnames[tt],
+         ".PA.", "dat", sep = ""))
+     cat(pred.PA[[tt]]$grid, "\n", sep = "   ")
+     for (j in 1:ncov) {
+         for (jj in 1:nquant) cat(pred.PA[[tt]]$quant.Surv[[j]][jj,
+             ], "\n", sep = "   ")
+     }
+     sink()
+     sink(paste(predCurvesdir, "quanthazardCaries.", tnames[tt],
+         ".PA.", "dat", sep = ""))
+     cat(pred.PA[[tt]]$grid, "\n", sep = "   ")
+     for (j in 1:ncov) {
+         for (jj in 1:nquant) cat(pred.PA[[tt]]$quant.hazard[[j]][jj,
+             ], "\n", sep = "   ")
+     }
+     sink()
+ }
```

6.4 Read the predictive quantities and plot them

We read the predictive quantities from files and plot some of them.

• Some useful variables.

```
> tnames <- c("16", "26", "36", "46")
> qnames <- c("2.5%", "50%", "97.5%")
> nquant <- length(qnames)
> ncov <- 32
```

• Read posterior predictive means.

```
> pred.PA <- list()
> for (tt in 1:4) {
+     pred.PA[[tt]] <- list()
+     pred.PA[[tt]]$grid <- scan(paste(predCurvesdir, "predDensCaries.",
+         tnames[tt], ".PA.dat", sep = ""), nlines = 1)
+     ngrid <- length(pred.PA[[tt]]$grid)
+     pred.PA[[tt]]$density <- matrix(scan(paste(predCurvesdir,
+         "predDensCaries.", tnames[tt], ".PA.dat", sep = ""),
+         skip = 1), ncol = ngrid, byrow = TRUE)
+     pred.PA[[tt]]$Surv <- matrix(scan(paste(predCurvesdir, "predSurvCaries.",
+         tnames[tt], ".PA.dat", sep = ""), skip = 1), ncol = ngrid,
+         byrow = TRUE)
+     pred.PA[[tt]]$hazard <- matrix(scan(paste(predCurvesdir,
+         "predhazardCaries.", tnames[tt], ".PA.dat", sep = ""),
+         skip = 1), ncol = ngrid, byrow = TRUE)
+ }
> names(pred.PA) <- tnames
```

• Read posterior predictive quantiles.

```
> for (tt in 1:4) {
+     ngrid <- length(pred.PA[[tt]]$grid)
```

```
+     pred.PA[[tt]]$quant.density <- list()
+     pred.PA[[tt]]$quant.Surv <- list()
+     pred.PA[[tt]]$quant.hazard <- list()
+     for (j in 1:ncov) {
+         pred.PA[[tt]]$quant.density[[j]] <- matrix(scan(paste(predCurvesdir,
+             "quantDensCaries.", tnames[tt], ".PA.dat", sep = ""),
+             skip = nquant * (j - 1) + 1, nlines = nquant), ncol = ngrid,
+             byrow = TRUE)
+         pred.PA[[tt]]$quant.Surv[[j]] <- matrix(scan(paste(predCurvesdir,
+             "quantSurvCaries.", tnames[tt], ".PA.dat", sep = ""),
+             skip = nquant * (j - 1) + 1, nlines = nquant), ncol = ngrid,
+             byrow = TRUE)
+         pred.PA[[tt]]$quant.hazard[[j]] <- matrix(scan(paste(predCurvesdir,
+             "quanthazardCaries.", tnames[tt], ".PA.dat", sep = ""),
+             skip = nquant * (j - 1) + 1, nlines = nquant), ncol = ngrid,
+             byrow = TRUE)
+         rownames(pred.PA[[tt]]$quant.density[[j]]) <- qnames
+         rownames(pred.PA[[tt]]$quant.Surv[[j]]) <- qnames
+         rownames(pred.PA[[tt]]$quant.hazard[[j]]) <- qnames
+     }
+ }
```

• Some parameters for the plots.

```
> pdfwidth <- 6
> pdfheight <- 9
> teeth <- c(16, 26, 36, 46)
```

• Combinations of covariates we have.

```
> pGender <- factor(c("boy", "girl"))
> pBrush <- factor(c("not.daily", "daily"))
> pSeal <- factor(c("no", "yes"))
> pPlaque <- factor(c("none", "present"))
> pPrim5 <- factor(c("sound", "dmf"))
> pdata <- expand.grid(ffPrim5 = pPrim5, ffPlaque = pPlaque, fSeal = pSeal,
+     fBrush = pBrush, Gender = pGender)
> pdata$Prim5 <- 1 * (pdata$ffPrim5 == "dmf")
> pdata$Plaque <- 1 * (pdata$ffPlaque == "present")
> pdata$Seal <- 1 * (pdata$fSeal == "yes")
> pdata$Brush <- 1 * (pdata$fBrush == "daily")
> pdata$Girl <- 1 * (pdata$Gender == "girl")
> ncov <- nrow(pdata)
> print(pdata)
```

```
   ffPrim5 ffPlaque fSeal    fBrush Gender Prim5 Plaque Seal Brush Girl
1    sound     none    no not.daily    boy     0      0    0     0    0
2      dmf     none    no not.daily    boy     1      0    0     0    0
3    sound  present    no not.daily    boy     0      1    0     0    0
4      dmf  present    no not.daily    boy     1      1    0     0    0
5    sound     none   yes not.daily    boy     0      0    1     0    0
6      dmf     none   yes not.daily    boy     1      0    1     0    0
7    sound  present   yes not.daily    boy     0      1    1     0    0
8      dmf  present   yes not.daily    boy     1      1    1     0    0
9    sound     none    no     daily    boy     0      0    0     1    0
10     dmf     none    no     daily    boy     1      0    0     1    0
11   sound  present    no     daily    boy     0      1    0     1    0
12     dmf  present    no     daily    boy     1      1    0     1    0
```

```
13   sound     none   yes     daily    boy     0      0    1     1    0
14     dmf     none   yes     daily    boy     1      0    1     1    0
15   sound  present   yes     daily    boy     0      1    1     1    0
16     dmf  present   yes     daily    boy     1      1    1     1    0
17   sound     none    no not.daily   girl     0      0    0     0    1
18     dmf     none    no not.daily   girl     1      0    0     0    1
19   sound  present    no not.daily   girl     0      1    0     0    1
20     dmf  present    no not.daily   girl     1      1    0     0    1
21   sound     none   yes not.daily   girl     0      0    1     0    1
22     dmf     none   yes not.daily   girl     1      0    1     0    1
23   sound  present   yes not.daily   girl     0      1    1     0    1
24     dmf  present   yes not.daily   girl     1      1    1     0    1
25   sound     none    no     daily   girl     0      0    0     1    1
26     dmf     none    no     daily   girl     1      0    0     1    1
27   sound  present    no     daily   girl     0      1    0     1    1
28     dmf  present    no     daily   girl     1      1    0     1    1
29   sound     none   yes     daily   girl     0      0    1     1    1
30     dmf     none   yes     daily   girl     1      0    1     1    1
31   sound  present   yes     daily   girl     0      1    1     1    1
32     dmf  present   yes     daily   girl     1      1    1     1    1
```

• Predictive survivor functions, one covariate combination per plot and 95% confidence region. See Figure 1 for the plot of the tooth 16, boys and first 8 covariate combinations.

```
> sets <- list(set1 = 1:8, set2 = 9:16, set3 = 17:24, set4 = 25:32)
> label <- paste(pdata$Gender, ", brush: ", pdata$fBrush, ", seal: ",
+     pdata$fSeal, ", plaq: ", pdata$ffPlaque, ", ", pdata$ffPrim5,
+     sep = "")
> pdf(paste(figuredir, "predSurvCI_PA.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> for (tt in 1:4) {
+     for (ss in 1:length(sets)) {
+         par(mfrow = c(4, 2), bty = "n", lwd = 1.2)
+         for (ii in sets[[ss]]) {
+             plot(pred.PA[[tt]]$grid, pred.PA[[tt]]$Surv[ii, ],
+                 type = "l", lty = 1, xlab = "Age (years)", ylab = "Caries free",
+                 ylim = c(0, 1))
+             lines(pred.PA[[tt]]$grid, pred.PA[[tt]]$quant.Surv[[ii]]["2.5%",
+                 ], lty = 2)
+             lines(pred.PA[[tt]]$grid, pred.PA[[tt]]$quant.Surv[[ii]]["97.5%",
+                 ], lty = 2)
+             title(main = paste("Tooth ", teeth[tt], sep = ""),
+                 sub = label[ii])
+         }
+     }
+ }
> dev.off()
```

• Predictive hazard functions, one covariate combination per plot and 95% confidence region. See Figure 2 for the plot of the tooth 16, boys and first 8 covariate combinations.

```
> pdf(paste(figuredir, "predhazardCI_PA.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> ylim <- NULL
> for (tt in 1:4) {
+     for (ss in 1:length(sets)) {
+         par(mfrow = c(4, 2), bty = "n", lwd = 1.2)
+         for (ii in sets[[ss]]) {
+             plot(pred.PA[[tt]]$grid, pred.PA[[tt]]$quant.hazard[[ii]]["97.5%",
```

```
+                 ], type = "l", lty = 2, xlab = "Age (years)",
+                 ylab = "Hazard for caries", ylim = ylim)
+             lines(pred.PA[[tt]]$grid, pred.PA[[tt]]$hazard[ii,
+                 ], lty = 1)
+             lines(pred.PA[[tt]]$grid, pred.PA[[tt]]$quant.hazard[[ii]]["2.5%",
+                 ], lty = 2)
+             title(main = paste("Tooth ", teeth[tt], sep = ""),
+                 sub = label[ii])
+         }
+     }
+ }
> dev.off()
```

• Posterior predictive means of the survivor functions. 16 covariate combinations per plot, 1 plot per tooth and gender. See Figure 3 for the plot of the tooth 16 and boys.

red = *not.daily brushers* and *dmf*;
orange = *not.daily brushers* and *sound*;
blue = *daily brushers* and *dmf*;
darkblue = *daily brushers* and *sound*;

Line type: 2 = *no plaque, no seal*; 4 = *present plaque, no seal*; 1 = *no plaque, sealed*; 3 = *present plaque, sealed.*

```
> col <- c(rep(c("orange", "red"), 4), rep(c("darkblue", "blue"),
+     4))
> lty <- rep(c(2, 2, 4, 4, 1, 1, 3, 3), 2)
> glabel <- c("Boy", "Girl")
> ncov.gender <- ncov/2
> pdf(paste(figuredir, "predSurv_PA.pdf", sep = ""), width = pdfheight,
+     height = pdfwidth)
> for (tt in 1:4) {
+     for (gg in 1:2) {
+         par(mfrow = c(1, 1), bty = "n", lwd = 1.2)
+         plot(pred.PA[[tt]]$grid, pred.PA[[tt]]$Surv[(gg - 1) *
+             ncov.gender + 1, ], type = "l", lty = lty[1], col = col[1],
+             xlab = "Time since emergence (years)", ylab = "Caries free",
+             ylim = c(0, 1))
+         for (i in 2:ncov.gender) {
+             lines(pred.PA[[tt]]$grid, pred.PA[[tt]]$Surv[(gg -
+                 1) * ncov.gender + i, ], lty = lty[i], col = col[i])
+         }
+         title(main = paste("Tooth ", teeth[tt], ", ", glabel[gg],
+             sep = ""))
+         legend(0, 0.7, legend = c("No plaque, sealed", "No plaque, not sealed",
+             "Plaque, sealed", "Plaque, not sealed"), lty = 1:4,
+             bty = "n", y.intersp = 1.2, col = "black")
+         legend(0, 0.3, legend = c("Daily brush, sound primary",
+             "Daily brush, dmf primary", "Not daily brush, sound primary",
+             "Not daily brush, dmf primary"), lty = 1, bty = "n",
+             y.intersp = 1.2, col = c("darkblue", "blue", "orange",
+                 "red"))
+     }
+ }
> dev.off()
```

• Posterior predictive means of the hazard functions. 16 covariate combinations per plot, 1 plot per tooth and gender. See Figure 4 for the plot of the tooth 16 and boys. Format is the same as for the survivor functions.

```
> pdf(paste(figuredir, "predhazard_PA.pdf", sep = ""), width = pdfheight,
+     height = pdfwidth)
> ylim <- c(0, 0.25)
> for (tt in 1:4) {
+     for (gg in 1:2) {
+         par(mfrow = c(1, 1), bty = "n", lwd = 1.2)
+         plot(pred.PA[[tt]]$grid, pred.PA[[tt]]$hazard[(gg - 1) *
+             ncov.gender + 1, ], type = "l", lty = lty[1], col = col[1],
+             xlab = "Time since emergence (years)", ylab = "Hazard for caries",
+             ylim = ylim)
+         for (i in 2:ncov.gender) {
+             lines(pred.PA[[tt]]$grid, pred.PA[[tt]]$hazard[(gg -
+                 1) * ncov.gender + i, ], lty = lty[i], col = col[i])
+         }
+         title(main = paste("Tooth ", teeth[tt], ", ", glabel[gg],
+             sep = ""))
+         legend(2.5, 0.27, legend = c("No plaque, sealed", "No plaque, not sealed",
+             "Plaque, sealed", "Plaque, not sealed"), lty = 1:4,
+             bty = "n", y.intersp = 1.2, col = "black")
+         legend(2.5, 0.2, legend = c("Daily brush, sound primary",
+             "Daily brush, dmf primary", "Not daily brush, sound primary",
+             "Not daily brush, dmf primary"), lty = 1, bty = "n",
+             y.intersp = 1.2, col = c("darkblue", "blue", "orange",
+                 "red"))
+     }
+ }
> dev.off()
```

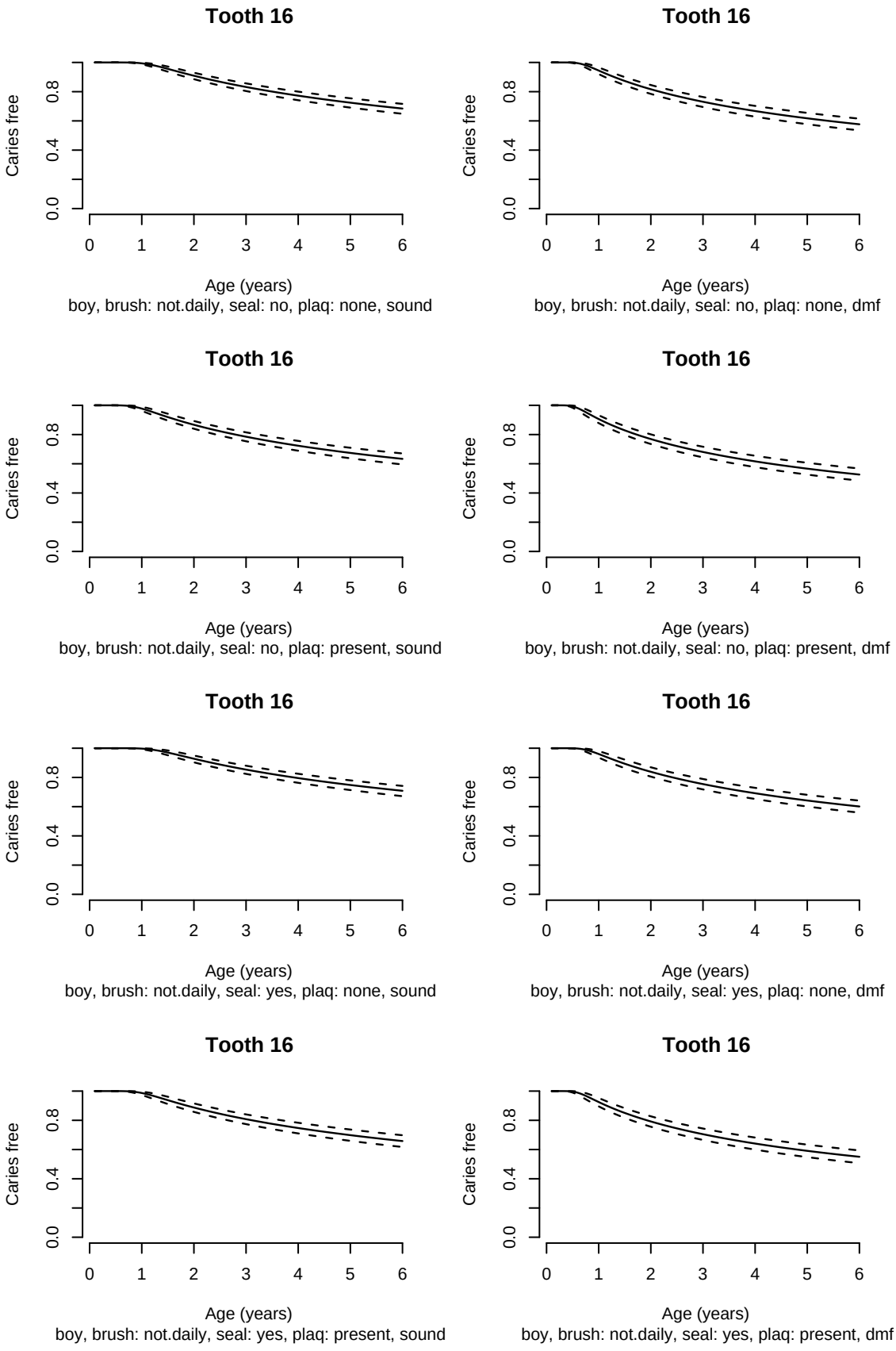


Figure 1: Predictive survivor curves (posterior predictive means and 95% pointwise credible regions) for caries for tooth 16 of boys.

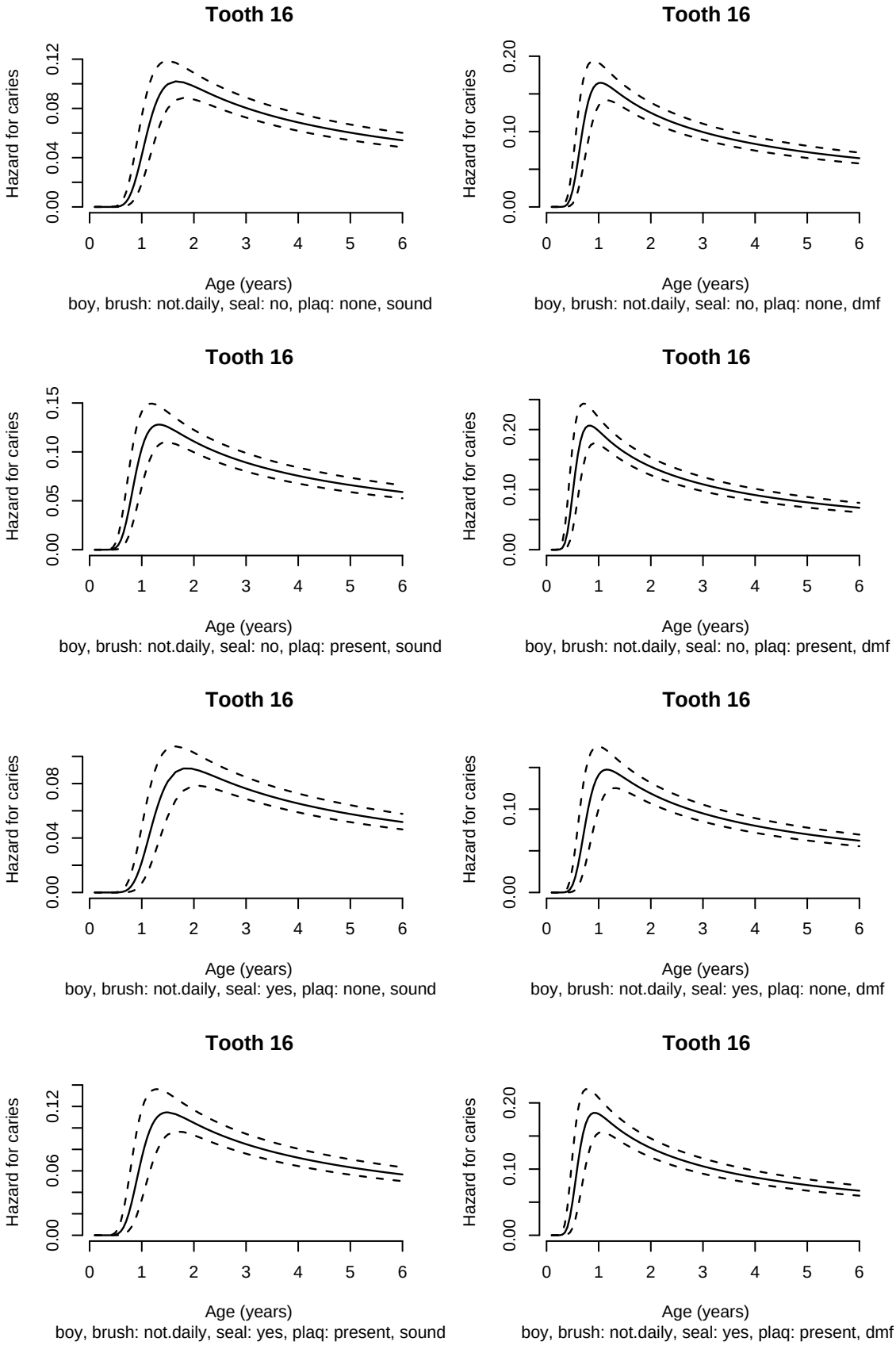


Figure 2: Predictive hazard curves (posterior predictive means and 95% pointwise credible regions) for caries for tooth 16 of boys.

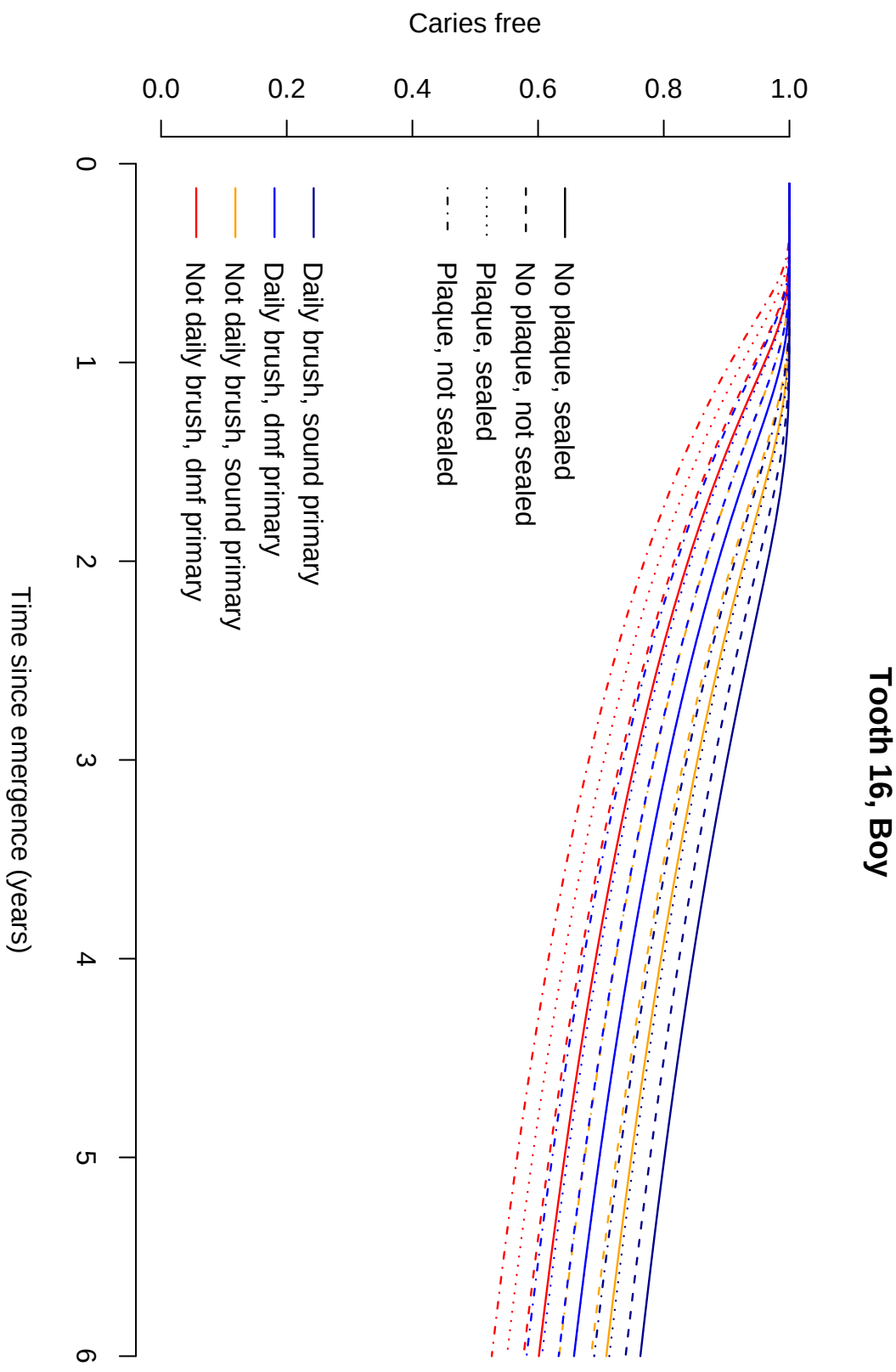


Figure 3: Predictive survivor curves (posterior predictive means) for caries for tooth 16 of boys.

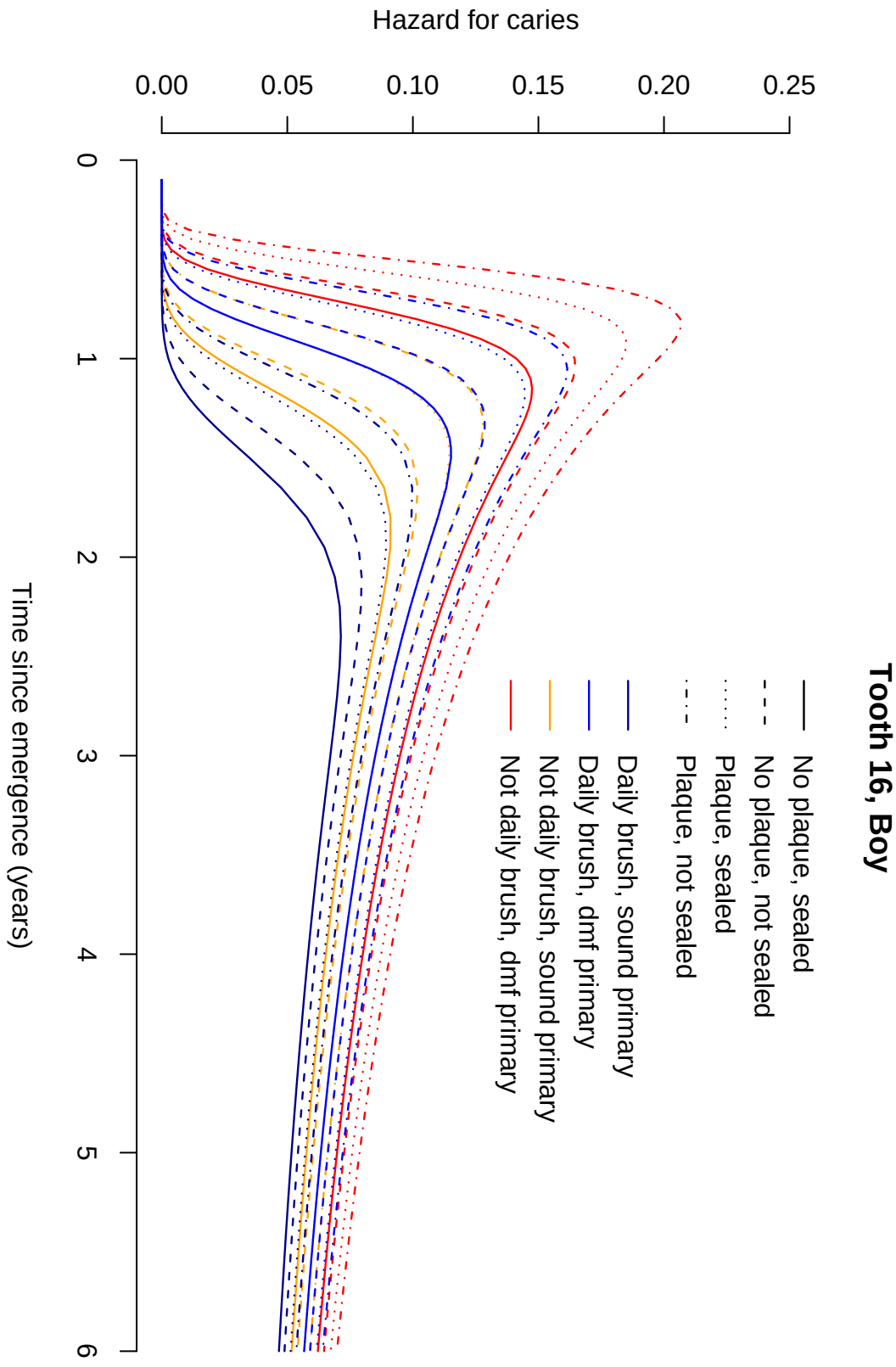


Figure 4: Predictive hazard curves (posterior predictive means) for caries for tooth 16 of boys.

7 Posterior predictive error densities – function `bayesGspline`

In this section we compute the estimates of the bivariate densities, store them in files and plot them. It is done separately for each pair of teeth (16-46 and 26-36). Computed estimates are stored in directories `chaindir.PA[1]` and `chaindir.PA[2]`.

7.1 Compute and store the results

• Specify whether some sampled values are to be skipped and how often the progress information will be print on the screen.

```
> skip <- 0
> nwrite <- 5000
```

• Predictive error density for the emergence.

```
> grid1.emerg <- seq(0, 0.8, length = 50)
> grid2.emerg <- seq(0, 0.8, length = 50)
> dens.emerg <- list()
> for (k in 1:2) {
+     dens.emerg[[k]] <- bayesGspline(chaindir.PA[k], grid1 = grid1.emerg,
+         grid2 = grid2.emerg, skip = skip, by = 1, nwrite = nwrite,
+         only.aver = TRUE)
+     sink(paste(chaindir.PA[k], "/densEmerg.dat", sep = ""), append = FALSE)
+     cat(dens.emerg[[k]]$grid1, "\n", sep = "  ")
+     cat(dens.emerg[[k]]$grid2, "\n", sep = "  ")
+     cat(dens.emerg[[k]]$average, "\n", sep = "  ")
+     sink()
+ }
```

• Predictive error density for the caries.

```
> grid1.caries <- seq(-0.5, 8, length = 50)
> grid2.caries <- seq(-0.5, 8, length = 50)
> dens.caries <- list()
> for (k in 1:2) {
+     dens.caries[[k]] <- bayesGspline(chaindir.PA[k], extens = "_2",
+         grid1 = grid1.caries, grid2 = grid2.caries, skip = 0,
+         by = 1, nwrite = 100, only.aver = TRUE)
+     sink(paste(chaindir.PA[k], "/densCaries.dat", sep = ""),
+         append = FALSE)
+     cat(dens.caries[[k]]$grid1, "\n", sep = "  ")
+     cat(dens.caries[[k]]$grid2, "\n", sep = "  ")
+     cat(dens.caries[[k]]$average, "\n", sep = "  ")
+     sink()
+ }
```

7.2 Read the results and plot them

The plots are stored in the directory `figuredir`.

• Some parameters for the plots.

```
> pdfwidth <- 9
> pdfheight <- 6
> tname <- c(1646, 2636)
> tname2 <- c("16, 46", "26, 36")
> col <- "lightblue"
```

• Read computed error densities for emergence from files.

```
> dens.emerg <- list()
> for (k in 1:2) {
+     dens.emerg[[k]] <- list()
+     dens.emerg[[k]]$grid1 <- scan(paste(chaindir.PA[k], "/densEmerg.dat",
+         sep = ""), nlines = 1)
+     dens.emerg[[k]]$grid2 <- scan(paste(chaindir.PA[k], "/densEmerg.dat",
+         sep = ""), skip = 1, nlines = 1)
+     ngrid1 <- length(dens.emerg[[k]]$grid1)
+     ngrid2 <- length(dens.emerg[[k]]$grid2)
+     if (ngrid1 != ngrid2)
+         stop("Different grid sizes read???")
+     ngrid <- ngrid1
+     dens.emerg[[k]]$average <- matrix(scan(paste(chaindir.PA[k],
+         "/densEmerg.dat", sep = ""), skip = 2, nlines = 1), ncol = ngrid)
+ }
> names(dens.emerg) <- c("right", "left")
```

• Read computed error densities for caries from files.

```
> dens.caries <- list()
> for (k in 1:2) {
+     dens.caries[[k]] <- list()
+     dens.caries[[k]]$grid1 <- scan(paste(chaindir.PA[k], "/densCaries.dat",
+         sep = ""), nlines = 1)
+     dens.caries[[k]]$grid2 <- scan(paste(chaindir.PA[k], "/densCaries.dat",
+         sep = ""), skip = 1, nlines = 1)
+     ngrid1 <- length(dens.caries[[k]]$grid1)
+     ngrid2 <- length(dens.caries[[k]]$grid2)
+     if (ngrid1 != ngrid2)
+         stop("Different grid sizes read???")
+     ngrid <- ngrid1
+     dens.caries[[k]]$average <- matrix(scan(paste(chaindir.PA[k],
+         "/densCaries.dat", sep = ""), skip = 2, nlines = 1),
+         ncol = ngrid)
+ }
> names(dens.caries) <- c("right", "left")
```

• Contour plot and three-dimensional plots from two different directions for the error density of the emergence. See Figures 5 and 6.

```
> theta <- c(30, 300)
> phi <- c(30, 30)
> for (k in 1:2) {
+     pdf(paste(figuredir, "errDensEmerg", tname[k], "_PA.pdf",
+         sep = ""), width = pdfwidth, height = pdfheight)
+     par(bty = "n", mfrow = c(2, 2), mar = c(4, 4, 4, 2) + 0.1)
+     contour(dens.emerg[[k]]$grid1, dens.emerg[[k]]$grid2, dens.emerg[[k]]$average,
+         lty = 1, main = paste("Emergence, teeth ", tname2[k],
+             sep = ""), xlim = c(0, 1), ylim = c(0, 1))
+     plot.new()
+     par(mar = c(0, 0, 0, 0) + 0.1)
+     for (i in 1:2) {
+         persp(dens.emerg[[k]]$grid1, dens.emerg[[k]]$grid2, dens.emerg[[k]]$average,
+             theta = theta[i], phi = phi[i], box = FALSE, xlab = "y1",
+             ylab = "y2", zlab = "g(y1, y2)", col = col)
+     }
```

```
+     dev.off()
+ }
```

• Contour plot and three-dimensional plots from two different directions for the error density of the caries See Figures 7 and 8.

```
> theta <- c(30, 300)
> phi <- c(30, 30)
> for (k in 1:2) {
+     pdf(paste(figuredir, "errDensCaries", tname[k], "_PA.pdf",
+         sep = ""), width = pdfwidth, height = pdfheight)
+     par(bty = "n", mfrow = c(2, 2), mar = c(4, 4, 4, 2) + 0.1)
+     contour(dens.caries[[k]]$grid1, dens.caries[[k]]$grid2, dens.caries[[k]]$average,
+         lty = 1, main = paste("Caries, teeth ", tname2[k], sep = ""),
+         xlim = c(-1, 9), ylim = c(-1, 9))
+     plot.new()
+     par(mar = c(0, 0, 0, 0) + 0.1)
+     for (i in 1:2) {
+         persp(dens.caries[[k]]$grid1, dens.caries[[k]]$grid2,
+             dens.caries[[k]]$average, theta = theta[i], phi = phi[i],
+             box = FALSE, xlab = "y1", ylab = "y2", zlab = "g(y1, y2)",
+             col = col)
+     }
+     dev.off()
+ }
```

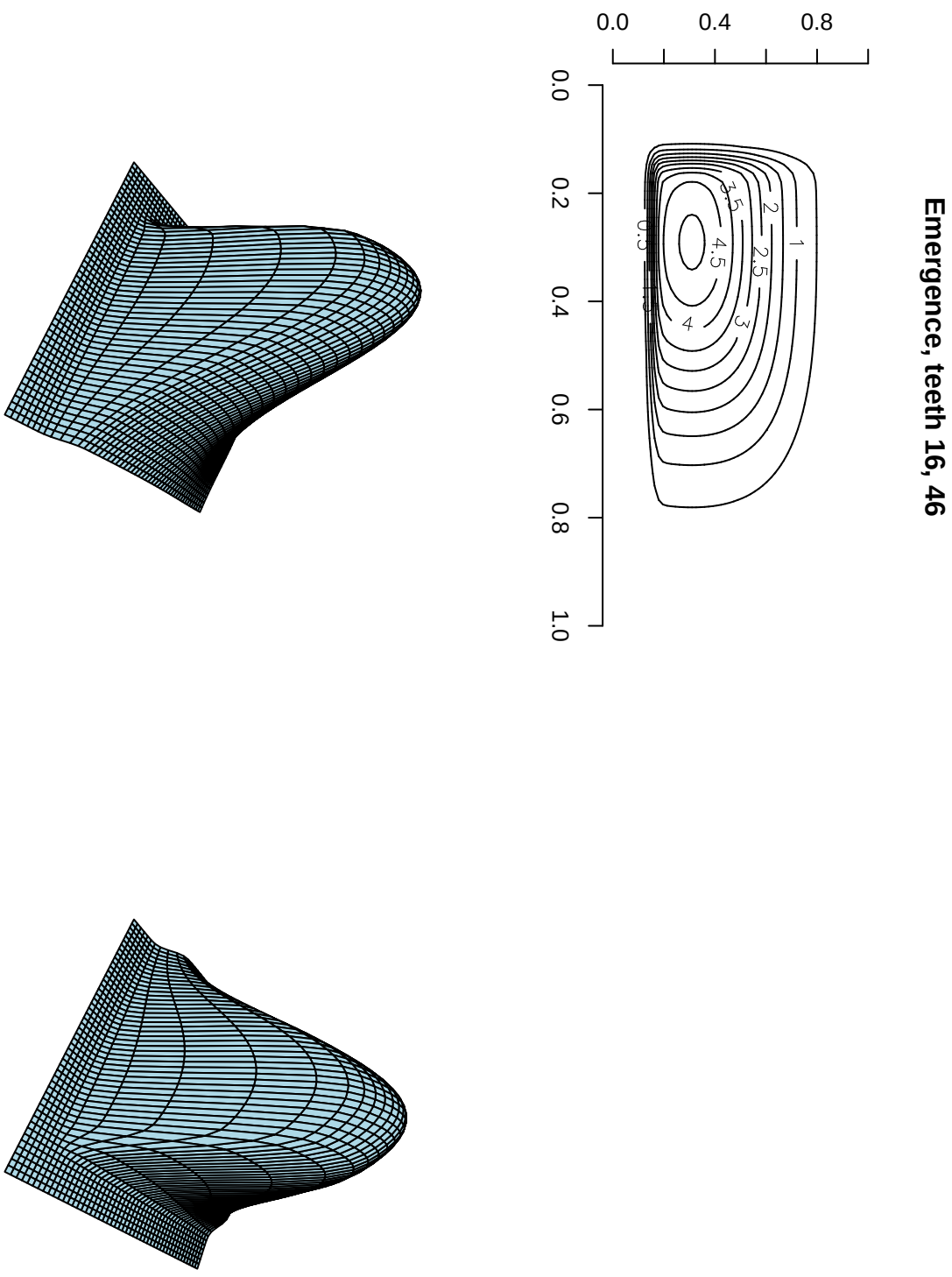


Figure 5: Estimate of the error density. Emergence of the right teeth 16 and 46.

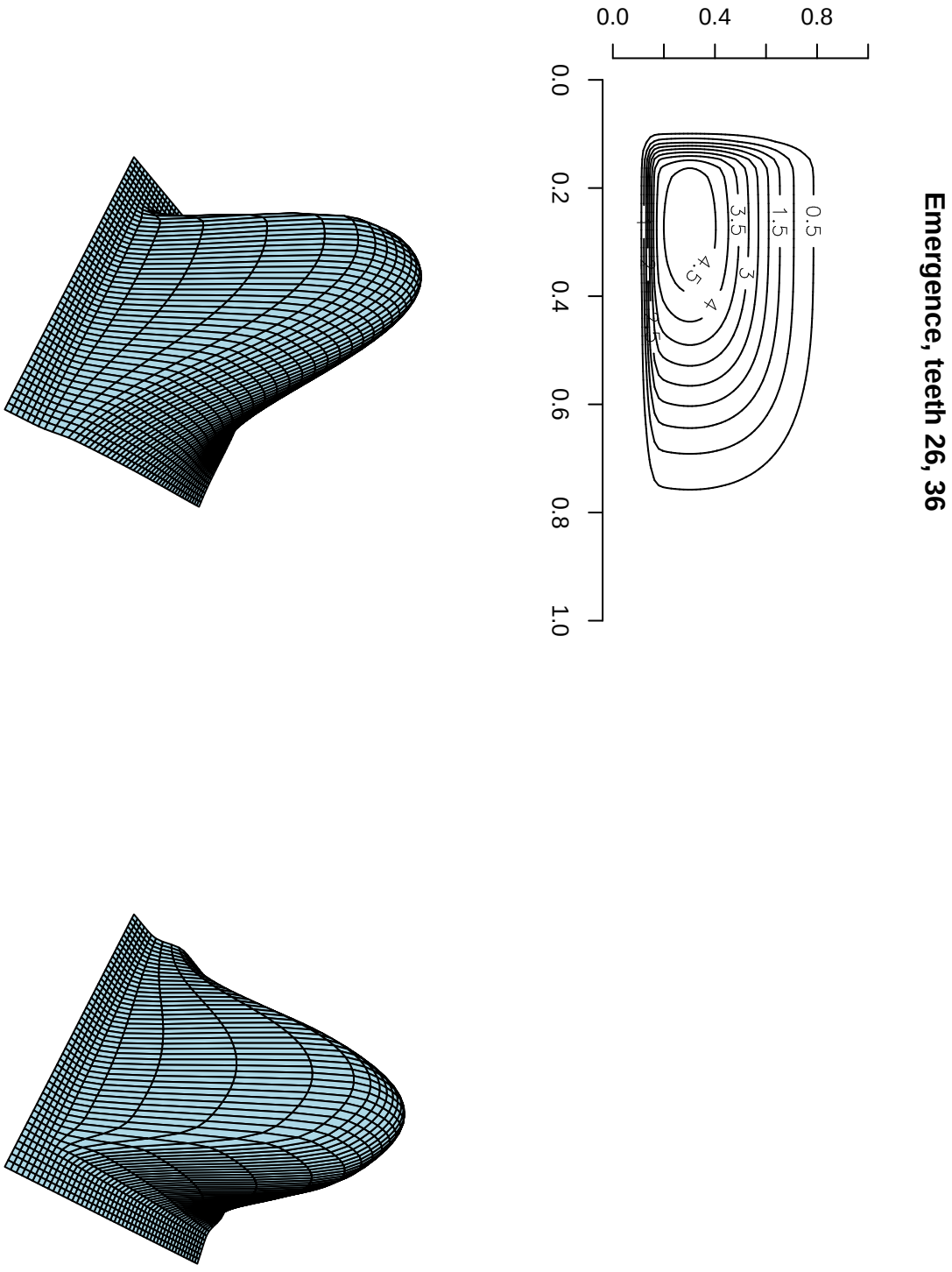


Figure 6: Estimate of the error density. Emergence of the left teeth 26 and 36.

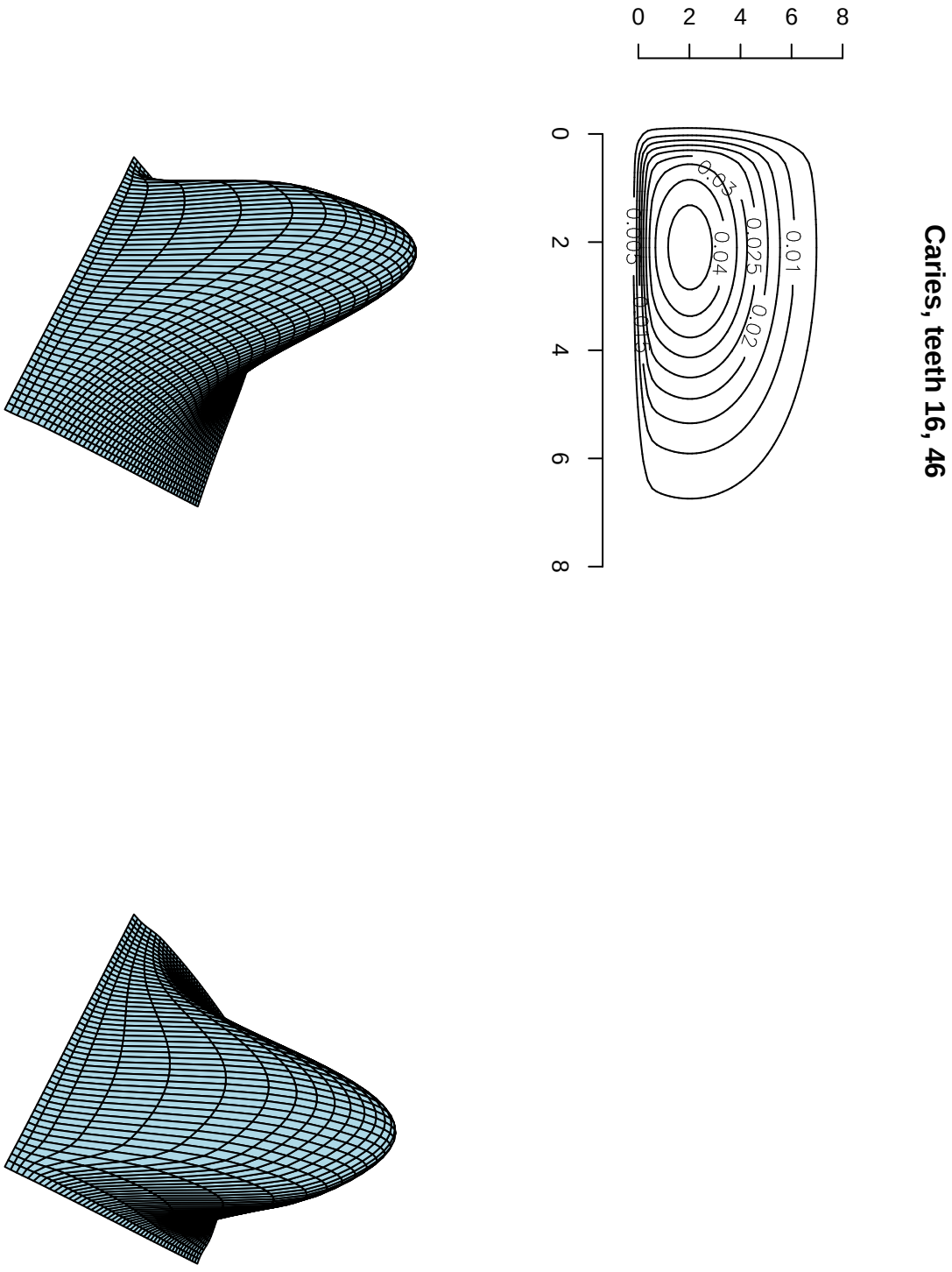


Figure 7: Estimate of the error density. Caries of the right teeth 16 and 46.

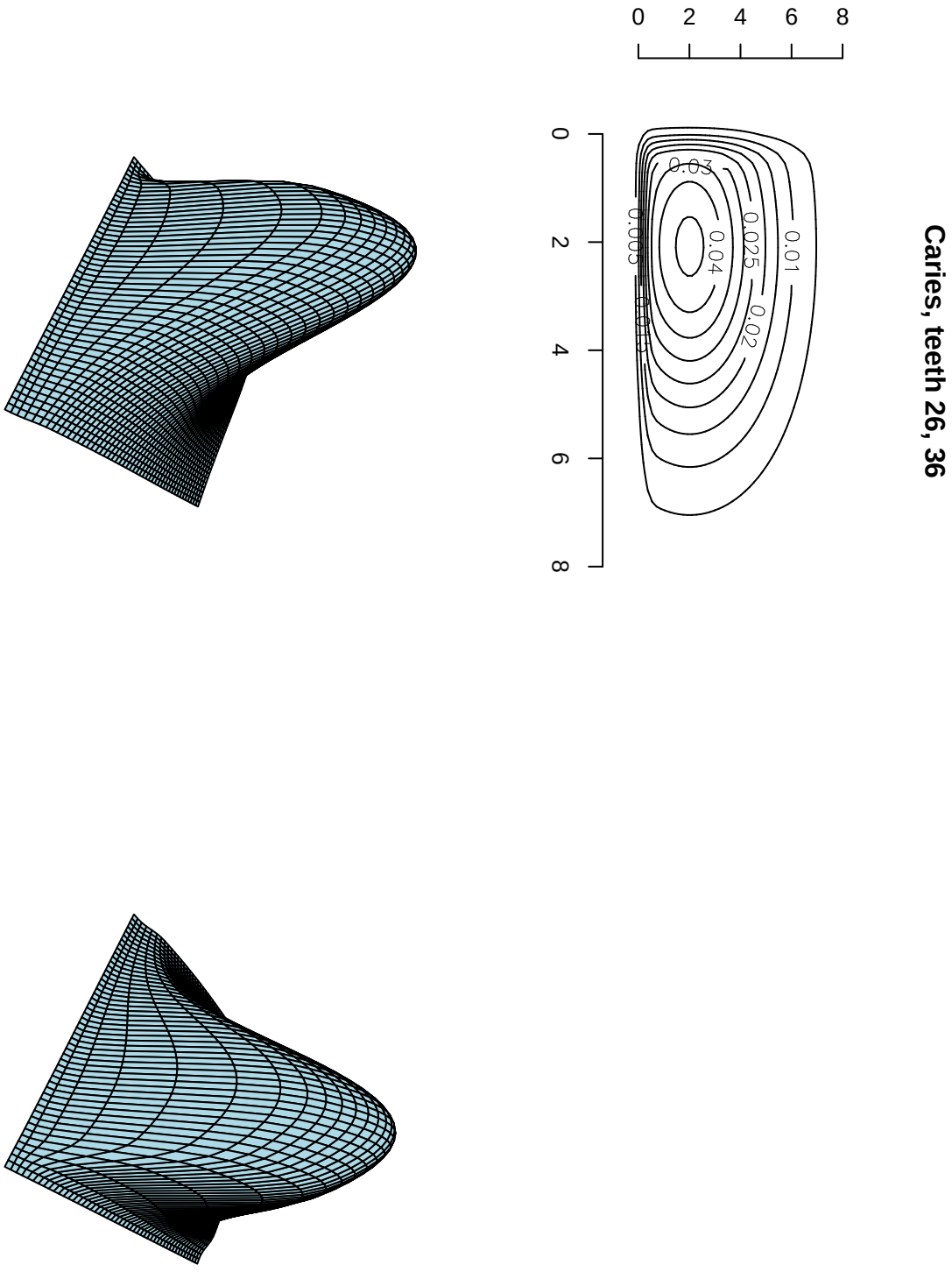


Figure 8: Estimate of the error density. Caries of the left teeth 26 and 36.

8 Read sampled chains

In this section we read the sampled chains. It is assumed that the chains for the model for the right teeth (16 and 46) are stored in the directory `chaindir.PA[1]` and the chains for the model for the left teeth (26 and 36) in the directory `chaindir.PA[2]`.

• Numbers of regression parameters for the emergence and caries parts of the model.

```
> nbeta.emerg.PA <- 1
> nbeta.caries.PA <- 5
```

• Read the indeces of stored iterations and show the beginning using the standard function `str`.

```
> itersInd.1 <- read.table(paste(chaindir.PA[1], "/iteration.sim",
+     sep = ""), header = TRUE)[, 1]
> itersInd.2 <- read.table(paste(chaindir.PA[2], "/iteration.sim",
+     sep = ""), header = TRUE)[, 1]
> if (sum(itersInd.1 - itersInd.2) > 0) stop("Different simulation lengths for the left and right ch
> itersInd <- itersInd.1
> rm(list = c("itersInd.1", "itersInd.2"))
> nlines <- length(itersInd)
> str(itersInd)
```

```
 int [1:25000] 225001 225002 225003 225004 225005 225006 225007 225008 225009 225010 ...
```

8.1 Chains for the emergence part of the model

We read the sampled chains from the files `mixmoment.sim`, `lambda.sim`, `logposter.sim` and `beta.sim` and store them (or their transformations) in the lists error.emerg.PA, beta.emerg.PA, logposter.emerg.PA which have each two components, one component for the model with the right teeth 16 and 46 and one component for the model with the left teeth 26 and 36. Additionally we create the variable lambda.emerg.PA which will be a data frame with 4 columns.

Note how the main effect of the upper (maxillary) tooth is defined as the difference of the two intercept terms.

```
> mixmoment.emerg.PA <- list()
> error.emerg.PA <- list()
> lambda.emerg.PA <- list()
> logposter.emerg.PA <- list()
> beta.emerg.PA <- list()
> for (k in 1:2) {
+     mixmoment.emerg.PA[[k]] <- matrix(scan(paste(chaindir.PA[k],
+         "/mixmoment.sim", sep = ""), skip = 1, nlines = nlines),
+         ncol = 6, byrow = TRUE)
+     lambda.emerg.PA[[k]] <- matrix(scan(paste(chaindir.PA[k],
+         "/lambda.sim", sep = ""), skip = 1, nlines = nlines),
+         ncol = 2, byrow = TRUE)
+     logposter.emerg.PA[[k]] <- matrix(scan(paste(chaindir.PA[k],
+         "/logposter.sim", sep = ""), skip = 1, nlines = nlines),
+         ncol = 4, byrow = TRUE)
+     beta.emerg.PA[[k]] <- matrix(scan(paste(chaindir.PA[k], "/beta.sim",
+         sep = ""), skip = 1, nlines = nlines), ncol = nbeta.emerg.PA,
+         byrow = TRUE)
+     colnames(mixmoment.emerg.PA[[k]]) <- scan(paste(chaindir.PA[k],
+         "/mixmoment.sim", sep = ""), what = "character", nlines = 1)
+     colnames(lambda.emerg.PA[[k]]) <- scan(paste(chaindir.PA[k],
+         "/lambda.sim", sep = ""), what = "character", nlines = 1)
```

```
+     colnames(logposter.emerg.PA[[k]]) <- scan(paste(chaindir.PA[k],
+         "/logposter.sim", sep = ""), what = "character", nlines = 1)
+     colnames(beta.emerg.PA[[k]]) <- scan(paste(chaindir.PA[k],
+         "/beta.sim", sep = ""), what = "character", nlines = 1)
+     scale1 <- sqrt(mixmoment.emerg.PA[[k]][, "D.1.1"])
+     scale2 <- sqrt(mixmoment.emerg.PA[[k]][, "D.2.2"])
+     correl <- mixmoment.emerg.PA[[k]][, "D.2.1"]/(scale1 * scale2)
+     error.emerg.PA[[k]] <- data.frame(Mean1 = mixmoment.emerg.PA[[k]][,
+         "Mean.1"], Mean2 = mixmoment.emerg.PA[[k]][, "Mean.2"],
+         Scale1 = scale1, Scale2 = scale2, Covar = mixmoment.emerg.PA[[k]][,
+             "D.2.1"], Correl = correl)
+     beta.emerg.PA[[k]] <- data.frame(Maxilla = error.emerg.PA[[k]]$Mean1 -
+         error.emerg.PA[[k]]$Mean2, Girl = beta.emerg.PA[[k]][,
+         "Girl"])
+     logposter.emerg.PA[[k]] <- as.data.frame(logposter.emerg.PA[[k]])
+     rm(list = c("scale1", "scale2", "correl"))
+ }
> rm(list = c("mixmoment.emerg.PA"))
> names(error.emerg.PA) <- c("right", "left")
> names(beta.emerg.PA) <- c("right", "left")
> names(logposter.emerg.PA) <- c("right", "left")
> lambda.emerg.PA <- data.frame(lambda16 = lambda.emerg.PA[[1]][,
+     1], lambda26 = lambda.emerg.PA[[2]][, 1], lambda36 = lambda.emerg.PA[[2]][,
+     2], lambda46 = lambda.emerg.PA[[1]][, 2])
```

• Show the beginning of each read and kept quantity for the right teeth using the standard function `str`.

```
> str(error.emerg.PA$right)

'data.frame':        25000 obs. of  6 variables:
 $ Mean1 : num  0.384 0.383 0.383 0.383 0.383 ...
 $ Mean2 : num  0.407 0.407 0.408 0.405 0.404 ...
 $ Scale1: num  0.166 0.168 0.167 0.167 0.167 ...
 $ Scale2: num  0.167 0.168 0.167 0.169 0.168 ...
 $ Covar : num  0.00144 0.00146 0.00145 0.00146 0.00145 ...
 $ Correl: num  0.0519 0.0518 0.0517 0.0517 0.0516 ...

> str(beta.emerg.PA$right)

'data.frame':        25000 obs. of  2 variables:
 $ Maxilla: num  -0.0224 -0.0242 -0.0253 -0.0216 -0.0209 ...
 $ Girl   : num  -0.00871 -0.00748 -0.00794 -0.00776 -0.00790 ...

> str(logposter.emerg.PA$right)

'data.frame':        25000 obs. of  4 variables:
 $ loglik  : num  1029 1043 1083 1076 1068 ...
 $ penalty1: num  -0.00558 -0.00560 -0.00611 -0.00592 -0.00480 ...
 $ penalty2: num  -0.00598 -0.00521 -0.00473 -0.00512 -0.00541 ...
 $ logprw  : num  -16979 -16968 -16973 -16969 -16954 ...

> str(lambda.emerg.PA)

'data.frame':        25000 obs. of  4 variables:
 $ lambda16: num  46454 41607 47311 46224 48807 ...
 $ lambda26: num  37408 41494 47822 42681 37662 ...
 $ lambda36: num  47569 47585 44614 47933 45743 ...
 $ lambda46: num  45224 50443 50059 46925 49661 ...
```

8.2 Chains for the caries part of the model

We read the sampled chains from the files `mixmoment_2.sim`, `lambda_2.sim`, `logposter_2.sim` and `beta_2.sim` and store them (or their transformations) in the lists error.caries.PA, beta.caries.PA, logposter.caries.PA which have each two components, one component for the model with the right teeth 16 and 46 and one component for the model with the left teeth 26 and 36. Additionally we create the variable lambda.caries.PA which will be a data frame with 4 columns.

Note how the main effect of the upper (maxillary) tooth is defined as the difference of the two intercept terms.

```
> mixmoment.caries.PA <- list()
> error.caries.PA <- list()
> lambda.caries.PA <- list()
> logposter.caries.PA <- list()
> beta.caries.PA <- list()
> for (k in 1:2) {
+     mixmoment.caries.PA[[k]] <- matrix(scan(paste(chaindir.PA[k],
+         "/mixmoment_2.sim", sep = ""), skip = 1, nlines = nlines),
+         ncol = 6, byrow = TRUE)
+     lambda.caries.PA[[k]] <- matrix(scan(paste(chaindir.PA[k],
+         "/lambda_2.sim", sep = ""), skip = 1, nlines = nlines),
+         ncol = 2, byrow = TRUE)
+     logposter.caries.PA[[k]] <- matrix(scan(paste(chaindir.PA[k],
+         "/logposter_2.sim", sep = ""), skip = 1, nlines = nlines),
+         ncol = 4, byrow = TRUE)
+     beta.caries.PA[[k]] <- matrix(scan(paste(chaindir.PA[k],
+         "/beta_2.sim", sep = ""), skip = 1, nlines = nlines),
+         ncol = nbeta.caries.PA, byrow = TRUE)
+     colnames(mixmoment.caries.PA[[k]]) <- scan(paste(chaindir.PA[k],
+         "/mixmoment_2.sim", sep = ""), what = "character", nlines = 1)
+     colnames(lambda.caries.PA[[k]]) <- scan(paste(chaindir.PA[k],
+         "/lambda_2.sim", sep = ""), what = "character", nlines = 1)
+     colnames(logposter.caries.PA[[k]]) <- scan(paste(chaindir.PA[k],
+         "/logposter_2.sim", sep = ""), what = "character", nlines = 1)
+     colnames(beta.caries.PA[[k]]) <- scan(paste(chaindir.PA[k],
+         "/beta_2.sim", sep = ""), what = "character", nlines = 1)
+     scale1 <- sqrt(mixmoment.caries.PA[[k]][, "D.1.1"])
+     scale2 <- sqrt(mixmoment.caries.PA[[k]][, "D.2.2"])
+     correl <- mixmoment.caries.PA[[k]][, "D.2.1"]/(scale1 * scale2)
+     error.caries.PA[[k]] <- data.frame(Mean1 = mixmoment.caries.PA[[k]][,
+         "Mean.1"], Mean2 = mixmoment.caries.PA[[k]][, "Mean.2"],
+         Scale1 = scale1, Scale2 = scale2, Covar = mixmoment.caries.PA[[k]][,
+             "D.2.1"], Correl = correl)
+     beta.caries.PA[[k]] <- data.frame(Maxilla = error.caries.PA[[k]]$Mean1 -
+         error.caries.PA[[k]]$Mean2, Girl = beta.caries.PA[[k]][,
+         "Girl"], Brush = beta.caries.PA[[k]][, "Brush"], Plaque = beta.caries.PA[[k]][,
+         "Plaque"], Seal = beta.caries.PA[[k]][, "Seal"], Prim5 = beta.caries.PA[[k]][,
+         "Prim5"])
+     logposter.caries.PA[[k]] <- as.data.frame(logposter.caries.PA[[k]])
+     rm(list = c("scale1", "scale2", "correl"))
+ }
> rm(list = c("mixmoment.caries.PA"))
> names(error.caries.PA) <- c("right", "left")
> names(beta.caries.PA) <- c("right", "left")
> names(logposter.caries.PA) <- c("right", "left")
> lambda.caries.PA <- data.frame(lambda16 = lambda.caries.PA[[1]][,
+     1], lambda26 = lambda.caries.PA[[2]][, 1], lambda36 = lambda.caries.PA[[2]][,
+     2], lambda46 = lambda.caries.PA[[1]][, 2])
```

• Show the beginning of each read and kept quantity for the right teeth using the standard function `str`.

```
> str(error.caries.PA$right)

'data.frame':        25000 obs. of  6 variables:
 $ Mean1 : num  2.96 2.95 2.96 2.96 2.97 ...
 $ Mean2 : num  2.96 2.96 2.96 2.96 2.96 ...
 $ Scale1: num  1.80 1.79 1.79 1.79 1.78 ...
 $ Scale2: num  1.87 1.88 1.87 1.85 1.84 ...
 $ Covar : num  0.0960 0.0957 0.0957 0.0939 0.0931 ...
 $ Correl: num  0.0286 0.0285 0.0287 0.0283 0.0283 ...

> str(beta.caries.PA$right)

'data.frame':        25000 obs. of  6 variables:
 $ Maxilla: num   0.00400 -0.00898  0.00453 -0.00346  0.01568 ...
 $ Girl   : num  -0.00526  0.00735  0.01291 -0.02082 -0.04078 ...
 $ Brush  : num  0.178 0.175 0.167 0.182 0.179 ...
 $ Plaque : num  -0.215 -0.209 -0.237 -0.199 -0.175 ...
 $ Seal   : num  0.136 0.105 0.138 0.100 0.125 ...
 $ Prim5  : num  -0.503 -0.507 -0.508 -0.534 -0.529 ...

> str(logposter.caries.PA$right)

'data.frame':        25000 obs. of  4 variables:
 $ loglik  : num  1004 1121 1021 1044  972 ...
 $ penalty1: num  -0.00570 -0.00602 -0.00531 -0.00589 -0.00523 ...
 $ penalty2: num  -0.00591 -0.00499 -0.00525 -0.00472 -0.00500 ...
 $ logprw  : num  -16046 -16055 -16048 -16058 -16062 ...

> str(lambda.caries.PA)

'data.frame':        25000 obs. of  4 variables:
 $ lambda16: num  42103 45179 44100 42830 47974 ...
 $ lambda26: num  49103 48016 47328 54919 51495 ...
 $ lambda36: num  48598 44743 47487 44980 47512 ...
 $ lambda46: num  42728 47288 46121 49685 51423 ...
```

9 Summary for some parameters – function `give.summary`

In this section we show how to compute some summary for the sampled chains.

• Compute the summary.

```
> summ.error.emerg.PA <- lapply(error.emerg.PA, give.summary)
> summ.error.caries.PA <- lapply(error.caries.PA, give.summary)
> summ.beta.emerg.PA <- lapply(beta.emerg.PA, give.summary)
> summ.beta.caries.PA <- lapply(beta.caries.PA, give.summary)
> summ.lambda.emerg.PA <- give.summary(lambda.emerg.PA)
> summ.lambda.caries.PA <- give.summary(lambda.caries.PA)
```

• Round the summary.

```
> rsumm.error.emerg.PA <- lapply(summ.error.emerg.PA, round, dig = 4)
> rsumm.error.caries.PA <- lapply(summ.error.caries.PA, round,
+     dig = 4)
> rsumm.beta.emerg.PA <- lapply(summ.beta.emerg.PA, round, dig = 4)
> rsumm.beta.caries.PA <- lapply(summ.beta.caries.PA, round, dig = 4)
> rsumm.lambda.emerg.PA <- round(summ.lambda.emerg.PA, dig = 4)
> rsumm.lambda.caries.PA <- round(summ.lambda.caries.PA, dig = 4)
```

• Print the summary for the emergence part of the model.

```
> print(rsumm.beta.emerg.PA)

$right
        Maxilla    Girl
means   -0.0171 -0.0172
50%     -0.0170 -0.0172
2.5%    -0.0315 -0.0327
97.5%   -0.0027 -0.0028
p.value  0.0206  0.0178

$left
        Maxilla    Girl
means   -0.0188 -0.0039
50%     -0.0188 -0.0039
2.5%    -0.0348 -0.0187
97.5%   -0.0040  0.0116
p.value  0.0127  0.5837

> print(rsumm.error.emerg.PA)

$right
         Mean1  Mean2 Scale1 Scale2  Covar Correl
means   0.3919 0.4089 0.1700 0.1705 0.0011 0.0385
50%     0.3920 0.4091 0.1700 0.1704 0.0011 0.0371
2.5%    0.3794 0.3972 0.1626 0.1642 0.0009 0.0297
97.5%   0.4043 0.4206 0.1781 0.1772 0.0015 0.0505
p.value 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

$left
         Mean1  Mean2 Scale1 Scale2  Covar Correl
means   0.3784 0.3973 0.1650 0.1708 0.0010 0.0345
50%     0.3783 0.3976 0.1649 0.1707 0.0010 0.0341
2.5%    0.3658 0.3839 0.1579 0.1637 0.0008 0.0269
97.5%   0.3919 0.4098 0.1725 0.1784 0.0012 0.0431
p.value 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

```
> print(rsumm.lambda.emerg.PA)

        lambda16 lambda26 lambda36 lambda46
means   44936.22 41257.73 45390.13 45164.09
50%     44892.49 41201.10 45301.68 45117.77
2.5%    39508.62 36341.83 40110.77 39848.70
97.5%   50545.64 46581.81 51070.36 50770.34
p.value     0.00     0.00     0.00     0.00
```

• Print the summary for the caries part of the model.

```
> print(rsumm.beta.caries.PA)

$right
        Maxilla    Girl  Brush  Plaque   Seal   Prim5
means   -0.0262 -0.0440 0.2499 -0.2281 0.1092 -0.4819
50%     -0.0243 -0.0443 0.2493 -0.2281 0.1097 -0.4820
2.5%    -0.2183 -0.1198 0.1387 -0.3129 0.0188 -0.5755
97.5%    0.1579  0.0329 0.3692 -0.1412 0.1946 -0.3884
p.value  0.8155  0.2674 0.0000  0.0000 0.0218  0.0000

$left
        Maxilla    Girl  Brush  Plaque   Seal   Prim5
means    0.0399 -0.0504 0.2809 -0.2754 0.1076 -0.5344
50%      0.0416 -0.0501 0.2798 -0.2762 0.1086 -0.5345
2.5%    -0.2098 -0.1252 0.1672 -0.3646 0.0187 -0.6299
97.5%    0.2615  0.0238 0.3998 -0.1842 0.1921 -0.4397
p.value  0.7381  0.1868 0.0000  0.0000 0.0190  0.0000

> print(rsumm.error.caries.PA)

$right
         Mean1  Mean2 Scale1 Scale2  Covar Correl
means   2.8433 2.8695 1.7385 1.8140 0.0734 0.0233
50%     2.8465 2.8696 1.7371 1.8119 0.0731 0.0234
2.5%    2.6455 2.7063 1.6309 1.7216 0.0575 0.0182
97.5%   3.0428 3.0401 1.8551 1.9177 0.0901 0.0278
p.value 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

$left
         Mean1  Mean2 Scale1 Scale2  Covar Correl
means   2.9487 2.9088 1.8318 1.8237 0.0718 0.0215
50%     2.9486 2.9126 1.8329 1.8245 0.0719 0.0215
2.5%    2.7463 2.7232 1.7182 1.7126 0.0535 0.0160
97.5%   3.1483 3.0779 1.9455 1.9348 0.0892 0.0264
p.value 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

> print(rsumm.lambda.caries.PA)

        lambda16 lambda26 lambda36 lambda46
means   47643.48 48306.45 48226.48 48087.44
50%     47570.77 48250.21 48153.44 48041.87
2.5%    42164.11 42752.70 42661.15 42523.08
97.5%   53424.80 54187.00 54191.68 53965.57
p.value     0.00     0.00     0.00     0.00
```

10 Posterior densities and traceplots for some parameters – functions `densplot2` and `traceplot2`

In this section we show how to draw traceplots and estimates of the posterior densities for some parameters.

10.1 Posterior densities

These can be drawn using the following code.

```
> plfun <- "densplot2"
> plname <- "dens"
> pdfwidth <- 9
> pdfheight <- 6
> tt <- c(16, 26, 36, 46)
```

• Effect of Maxilla and Girl on the emergence (see Figure 9).

```
> pdf(paste(figuredir, plname, "_emergBeta.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(beta.emerg.PA[[2]][, "Maxilla"]), main = "Maxilla (emerg), left"))
> title(sub = paste("Maxilla = ", rsumm.beta.emerg.PA[[2]][1, "Maxilla"],
+     " (", rsumm.beta.emerg.PA[[2]][3, "Maxilla"], ", ", rsumm.beta.emerg.PA[[2]][4,
+         "Maxilla"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.emerg.PA[[1]][, "Maxilla"]), main = "Maxilla (emerg), right"))
> title(sub = paste("Maxilla = ", rsumm.beta.emerg.PA[[1]][1, "Maxilla"],
+     " (", rsumm.beta.emerg.PA[[1]][3, "Maxilla"], ", ", rsumm.beta.emerg.PA[[1]][4,
+         "Maxilla"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.emerg.PA[[2]][, "Girl"]), main = "Girl (emerg), left"))
> title(sub = paste("Girl = ", rsumm.beta.emerg.PA[[2]][1, "Girl"],
+     " (", rsumm.beta.emerg.PA[[2]][3, "Girl"], ", ", rsumm.beta.emerg.PA[[2]][4,
+         "Girl"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.emerg.PA[[1]][, "Girl"]), main = "Girl (emerg), right"))
> title(sub = paste("Girl = ", rsumm.beta.emerg.PA[[1]][1, "Girl"],
+     " (", rsumm.beta.emerg.PA[[1]][3, "Girl"], ", ", rsumm.beta.emerg.PA[[1]][4,
+         "Girl"], ")", sep = ""))
> dev.off()
```

• Intercept for the emergence process (see Figure 10).

```
> pdf(paste(figuredir, plname, "_emergIntcpt.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(error.emerg.PA[[1]][, "Mean1"]), main = "Intcpt (emerg), tooth 16"))
> title(sub = paste("Intercept = ", rsumm.error.emerg.PA[[1]][1,
+     "Mean1"], " (", rsumm.error.emerg.PA[[1]][3, "Mean1"], ", ",
+     rsumm.error.emerg.PA[[1]][4, "Mean1"], ")", sep = ""))
> eval(call(plfun, mcmc(error.emerg.PA[[2]][, "Mean1"]), main = "Intcpt (emerg), tooth 26"))
> title(sub = paste("Intercept = ", rsumm.error.emerg.PA[[2]][1,
+     "Mean1"], " (", rsumm.error.emerg.PA[[2]][3, "Mean1"], ", ",
+     rsumm.error.emerg.PA[[2]][4, "Mean1"], ")", sep = ""))
> eval(call(plfun, mcmc(error.emerg.PA[[2]][, "Mean2"]), main = "Intcpt (emerg), tooth 36"))
> title(sub = paste("Intercept = ", rsumm.error.emerg.PA[[2]][1,
+     "Mean2"], " (", rsumm.error.emerg.PA[[2]][3, "Mean2"], ", ",
+     rsumm.error.emerg.PA[[2]][4, "Mean2"], ")", sep = ""))
> eval(call(plfun, mcmc(error.emerg.PA[[1]][, "Mean2"]), main = "Intcpt (emerg), tooth 46"))
```

```
> title(sub = paste("Intercept = ", rsumm.error.emerg.PA[[1]][1,
+     "Mean2"], " (", rsumm.error.emerg.PA[[1]][3, "Mean2"], ", ",
+     rsumm.error.emerg.PA[[1]][4, "Mean2"], ")", sep = ""))
> dev.off()
```

• Scale for the emergence (see Figure 11).

```
> pdf(paste(figuredir, plname, "_emergScale.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(error.emerg.PA[[1]][, "Scale1"]), main = "Scale (emerg), tooth 16"))
> title(sub = paste("Scale = ", rsumm.error.emerg.PA[[1]][1, "Scale1"],
+     " (", rsumm.error.emerg.PA[[1]][3, "Scale1"], ", ", rsumm.error.emerg.PA[[1]][4,
+         "Scale1"], ")", sep = ""))
> eval(call(plfun, mcmc(error.emerg.PA[[2]][, "Scale1"]), main = "Scale (emerg), tooth 26"))
> title(sub = paste("Scale = ", rsumm.error.emerg.PA[[2]][1, "Scale1"],
+     " (", rsumm.error.emerg.PA[[2]][3, "Scale1"], ", ", rsumm.error.emerg.PA[[2]][4,
+         "Scale1"], ")", sep = ""))
> eval(call(plfun, mcmc(error.emerg.PA[[2]][, "Scale2"]), main = "Scale (emerg), tooth 36"))
> title(sub = paste("Scale = ", rsumm.error.emerg.PA[[2]][1, "Scale2"],
+     " (", rsumm.error.emerg.PA[[2]][3, "Scale2"], ", ", rsumm.error.emerg.PA[[2]][4,
+         "Scale2"], ")", sep = ""))
> eval(call(plfun, mcmc(error.emerg.PA[[1]][, "Scale2"]), main = "Scale (emerg), tooth 46"))
> title(sub = paste("Scale = ", rsumm.error.emerg.PA[[1]][1, "Scale2"],
+     " (", rsumm.error.emerg.PA[[1]][3, "Scale2"], ", ", rsumm.error.emerg.PA[[1]][4,
+         "Scale2"], ")", sep = ""))
> dev.off()
```

• Covariance and correlation for the emergence (see Figure 12).

```
> pdf(paste(figuredir, plname, "_emergCovCor.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(error.emerg.PA[[2]][, "Covar"]), main = "Covar (emerg), tooth 26 vs. 36"))
> title(sub = paste("Covar = ", rsumm.error.emerg.PA[[2]][1, "Covar"],
+     " (", rsumm.error.emerg.PA[[2]][3, "Covar"], ", ", rsumm.error.emerg.PA[[2]][4,
+         "Covar"], ")", sep = ""))
> eval(call(plfun, mcmc(error.emerg.PA[[1]][, "Covar"]), main = "Covar (emerg), tooth 16 vs. 46"))
> title(sub = paste("Covar = ", rsumm.error.emerg.PA[[1]][1, "Covar"],
+     " (", rsumm.error.emerg.PA[[1]][3, "Covar"], ", ", rsumm.error.emerg.PA[[1]][4,
+         "Covar"], ")", sep = ""))
> eval(call(plfun, mcmc(error.emerg.PA[[2]][, "Correl"]), main = "Correl (emerg), tooth 26 vs. 36"))
> title(sub = paste("Correl = ", rsumm.error.emerg.PA[[2]][1, "Correl"],
+     " (", rsumm.error.emerg.PA[[2]][3, "Correl"], ", ", rsumm.error.emerg.PA[[2]][4,
+         "Correl"], ")", sep = ""))
> eval(call(plfun, mcmc(error.emerg.PA[[1]][, "Correl"]), main = "Correl (emerg), tooth 16 vs. 46"))
> title(sub = paste("Correl = ", rsumm.error.emerg.PA[[1]][1, "Correl"],
+     " (", rsumm.error.emerg.PA[[1]][3, "Correl"], ", ", rsumm.error.emerg.PA[[1]][4,
+         "Correl"], ")", sep = ""))
> dev.off()
```

• Smoothing parameters λ_1^ζ and λ_2^ζ for the emergence (see Figure 13).

```
> pdf(paste(figuredir, plname, "_emergLambda.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> for (i in 1:4) {
+     eval(call(plfun, mcmc(lambda.emerg.PA[, i]), main = paste("Lambda (emergence), Tooth ",
+         tt[i], sep = "")))
```

```
+     title(sub = paste("lambda = ", rsumm.lambda.emerg.PA[1, i],
+         " (", rsumm.lambda.emerg.PA[3, i], ", ", rsumm.lambda.emerg.PA[4,
+             i], ")", sep = ""))
+ }
> dev.off()
```

• Effect of Maxilla and Girl on the caries (see Figure 14).

```
> pdf(paste(figuredir, plname, "_cariesBeta1.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(beta.caries.PA[[2]][, "Maxilla"]), main = "Maxilla (caries), left"))
> title(sub = paste("Maxilla = ", rsumm.beta.caries.PA[[2]][1,
+     "Maxilla"], " (", rsumm.beta.caries.PA[[2]][3, "Maxilla"],
+     ", ", rsumm.beta.caries.PA[[2]][4, "Maxilla"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.caries.PA[[1]][, "Maxilla"]), main = "Maxilla (caries), right"))
> title(sub = paste("Maxilla = ", rsumm.beta.caries.PA[[1]][1,
+     "Maxilla"], " (", rsumm.beta.caries.PA[[1]][3, "Maxilla"],
+     ", ", rsumm.beta.caries.PA[[1]][4, "Maxilla"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.caries.PA[[2]][, "Girl"]), main = "Girl (caries), left"))
> title(sub = paste("Girl = ", rsumm.beta.caries.PA[[2]][1, "Girl"],
+     " (", rsumm.beta.caries.PA[[2]][3, "Girl"], ", ", rsumm.beta.caries.PA[[2]][4,
+         "Girl"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.caries.PA[[1]][, "Girl"]), main = "Girl (caries), right"))
> title(sub = paste("Girl = ", rsumm.beta.caries.PA[[1]][1, "Girl"],
+     " (", rsumm.beta.caries.PA[[1]][3, "Girl"], ", ", rsumm.beta.caries.PA[[1]][4,
+         "Girl"], ")", sep = ""))
> dev.off()
```

• Effect of Brush and Seal on the caries (see Figure 15).

```
> pdf(paste(figuredir, plname, "_cariesBeta2.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(beta.caries.PA[[2]][, "Brush"]), main = "Brush (caries), left"))
> title(sub = paste("Brush = ", rsumm.beta.caries.PA[[2]][1, "Brush"],
+     " (", rsumm.beta.caries.PA[[2]][3, "Brush"], ", ", rsumm.beta.caries.PA[[2]][4,
+         "Brush"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.caries.PA[[1]][, "Brush"]), main = "Brush (caries), right"))
> title(sub = paste("Brush = ", rsumm.beta.caries.PA[[1]][1, "Brush"],
+     " (", rsumm.beta.caries.PA[[1]][3, "Brush"], ", ", rsumm.beta.caries.PA[[1]][4,
+         "Brush"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.caries.PA[[2]][, "Seal"]), main = "Seal (caries), left"))
> title(sub = paste("Seal = ", rsumm.beta.caries.PA[[2]][1, "Seal"],
+     " (", rsumm.beta.caries.PA[[2]][3, "Seal"], ", ", rsumm.beta.caries.PA[[2]][4,
+         "Seal"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.caries.PA[[1]][, "Seal"]), main = "Seal (caries), right"))
> title(sub = paste("Seal = ", rsumm.beta.caries.PA[[1]][1, "Seal"],
+     " (", rsumm.beta.caries.PA[[1]][3, "Seal"], ", ", rsumm.beta.caries.PA[[1]][4,
+         "Seal"], ")", sep = ""))
> dev.off()
```

• Effect of Plaque and Prim5 on the caries (see Figure 16).

```
> pdf(paste(figuredir, plname, "_cariesBeta3.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(beta.caries.PA[[2]][, "Plaque"]), main = "Plaque (caries), left"))
> title(sub = paste("Plaque = ", rsumm.beta.caries.PA[[2]][1, "Plaque"],
```

```
+     " (", rsumm.beta.caries.PA[[2]][3, "Plaque"], ", ", rsumm.beta.caries.PA[[2]][4,
+         "Plaque"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.caries.PA[[1]][, "Plaque"]), main = "Plaque (caries), right"))
> title(sub = paste("Plaque = ", rsumm.beta.caries.PA[[1]][1, "Plaque"],
+     " (", rsumm.beta.caries.PA[[1]][3, "Plaque"], ", ", rsumm.beta.caries.PA[[1]][4,
+         "Plaque"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.caries.PA[[2]][, "Prim5"]), main = "Prim5 (caries), left"))
> title(sub = paste("Prim5 = ", rsumm.beta.caries.PA[[2]][1, "Prim5"],
+     " (", rsumm.beta.caries.PA[[2]][3, "Prim5"], ", ", rsumm.beta.caries.PA[[2]][4,
+         "Prim5"], ")", sep = ""))
> eval(call(plfun, mcmc(beta.caries.PA[[1]][, "Prim5"]), main = "Prim5 (caries), right"))
> title(sub = paste("Prim5 = ", rsumm.beta.caries.PA[[1]][1, "Prim5"],
+     " (", rsumm.beta.caries.PA[[1]][3, "Prim5"], ", ", rsumm.beta.caries.PA[[1]][4,
+         "Prim5"], ")", sep = ""))
> dev.off()
```

• Intercept for the caries process (see Figure 17).

```
> pdf(paste(figuredir, plname, "_cariesIntcpt.pdf", sep = ""),
+     width = pdfwidth, height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(error.caries.PA[[1]][, "Mean1"]), main = "Intcpt (caries), tooth 16"))
> title(sub = paste("Intercept = ", rsumm.error.caries.PA[[1]][1,
+     "Mean1"], " (", rsumm.error.caries.PA[[1]][3, "Mean1"], ", ",
+     rsumm.error.caries.PA[[1]][4, "Mean1"], ")", sep = ""))
> eval(call(plfun, mcmc(error.caries.PA[[2]][, "Mean1"]), main = "Intcpt (caries), tooth 26"))
> title(sub = paste("Intercept = ", rsumm.error.caries.PA[[2]][1,
+     "Mean1"], " (", rsumm.error.caries.PA[[2]][3, "Mean1"], ", ",
+     rsumm.error.caries.PA[[2]][4, "Mean1"], ")", sep = ""))
> eval(call(plfun, mcmc(error.caries.PA[[2]][, "Mean2"]), main = "Intcpt (caries), tooth 36"))
> title(sub = paste("Intercept = ", rsumm.error.caries.PA[[2]][1,
+     "Mean2"], " (", rsumm.error.caries.PA[[2]][3, "Mean2"], ", ",
+     rsumm.error.caries.PA[[2]][4, "Mean2"], ")", sep = ""))
> eval(call(plfun, mcmc(error.caries.PA[[1]][, "Mean2"]), main = "Intcpt (caries), tooth 46"))
> title(sub = paste("Intercept = ", rsumm.error.caries.PA[[1]][1,
+     "Mean2"], " (", rsumm.error.caries.PA[[1]][3, "Mean2"], ", ",
+     rsumm.error.caries.PA[[1]][4, "Mean2"], ")", sep = ""))
> dev.off()
```

• Scale for caries (see Figure 18).

```
> pdf(paste(figuredir, plname, "_cariesScale.pdf", sep = ""), width = pdfwidth,
+     height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(error.caries.PA[[1]][, "Scale1"]), main = "Scale (caries), tooth 16"))
> title(sub = paste("Scale = ", rsumm.error.caries.PA[[1]][1, "Scale1"],
+     " (", rsumm.error.caries.PA[[1]][3, "Scale1"], ", ", rsumm.error.caries.PA[[1]][4,
+         "Scale1"], ")", sep = ""))
> eval(call(plfun, mcmc(error.caries.PA[[2]][, "Scale1"]), main = "Scale (caries), tooth 26"))
> title(sub = paste("Scale = ", rsumm.error.caries.PA[[2]][1, "Scale1"],
+     " (", rsumm.error.caries.PA[[2]][3, "Scale1"], ", ", rsumm.error.caries.PA[[2]][4,
+         "Scale1"], ")", sep = ""))
> eval(call(plfun, mcmc(error.caries.PA[[2]][, "Scale2"]), main = "Scale (caries), tooth 36"))
> title(sub = paste("Scale = ", rsumm.error.caries.PA[[2]][1, "Scale2"],
+     " (", rsumm.error.caries.PA[[2]][3, "Scale2"], ", ", rsumm.error.caries.PA[[2]][4,
+         "Scale2"], ")", sep = ""))
> eval(call(plfun, mcmc(error.caries.PA[[1]][, "Scale2"]), main = "Scale (caries), tooth 46"))
> title(sub = paste("Scale = ", rsumm.error.caries.PA[[1]][1, "Scale2"],
+     " (", rsumm.error.caries.PA[[1]][3, "Scale2"], ", ", rsumm.error.caries.PA[[1]][4,
```

```
+         "Scale2"], ")", sep = ""))
> dev.off()
```

• Covariance and correlation for the caries (see Figure 19).

```
> pdf(paste(figuredir, plname, "_cariesCovCor.pdf", sep = ""),
+     width = pdfwidth, height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> eval(call(plfun, mcmc(error.caries.PA[[2]][, "Covar"]), main = "Covar (caries), tooth 26 vs. 36"))
> title(sub = paste("Covar = ", rsumm.error.caries.PA[[2]][1, "Covar"],
+     " (", rsumm.error.caries.PA[[2]][3, "Covar"], ", ", rsumm.error.caries.PA[[2]][4,
+         "Covar"], ")", sep = ""))
> eval(call(plfun, mcmc(error.caries.PA[[1]][, "Covar"]), main = "Covar (caries), tooth 16 vs. 46"))
> title(sub = paste("Covar = ", rsumm.error.caries.PA[[1]][1, "Covar"],
+     " (", rsumm.error.caries.PA[[1]][3, "Covar"], ", ", rsumm.error.caries.PA[[1]][4,
+         "Covar"], ")", sep = ""))
> eval(call(plfun, mcmc(error.caries.PA[[2]][, "Correl"]), main = "Correl (caries), tooth 26 vs. 36"
> title(sub = paste("Correl = ", rsumm.error.caries.PA[[2]][1,
+     "Correl"], " (", rsumm.error.caries.PA[[2]][3, "Correl"],
+     ", ", rsumm.error.caries.PA[[2]][4, "Correl"], ")", sep = ""))
> eval(call(plfun, mcmc(error.caries.PA[[1]][, "Correl"]), main = "Correl (caries), tooth 16 vs. 46"
> title(sub = paste("Correl = ", rsumm.error.caries.PA[[1]][1,
+     "Correl"], " (", rsumm.error.caries.PA[[1]][3, "Correl"],
+     ", ", rsumm.error.caries.PA[[1]][4, "Correl"], ")", sep = ""))
> dev.off()
```

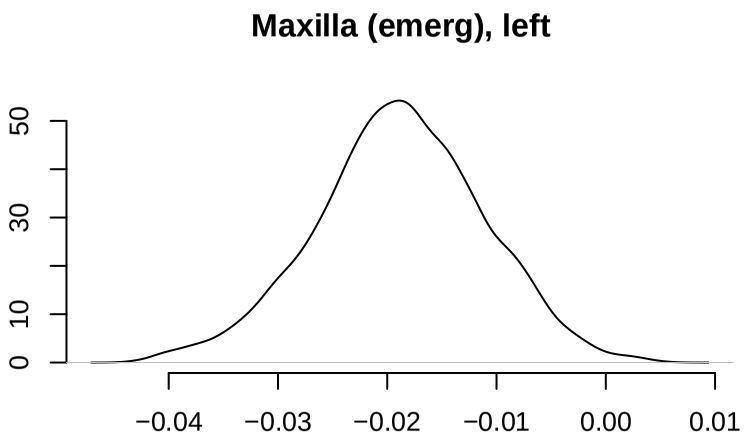
• Smoothing parameters λ_1^ε and λ_2^ε for the caries (see Figure 20).

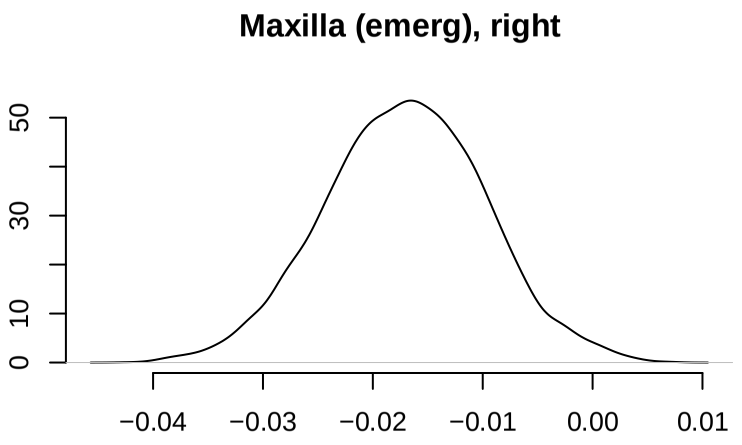
```
> pdf(paste(figuredir, plname, "_cariesLambda.pdf", sep = ""),
+     width = pdfwidth, height = pdfheight)
> par(mfrow = c(2, 2), bty = "n")
> for (i in 1:4) {
+     eval(call(plfun, mcmc(lambda.caries.PA[, i]), main = paste("Lambda (caries), Tooth ",
+         tt[i], sep = "")))
+     title(sub = paste("lambda = ", rsumm.lambda.caries.PA[1,
+         i], " (", rsumm.lambda.caries.PA[3, i], ", ", rsumm.lambda.caries.PA[4,
+         i], ")", sep = ""))
+ }
> dev.off()
```

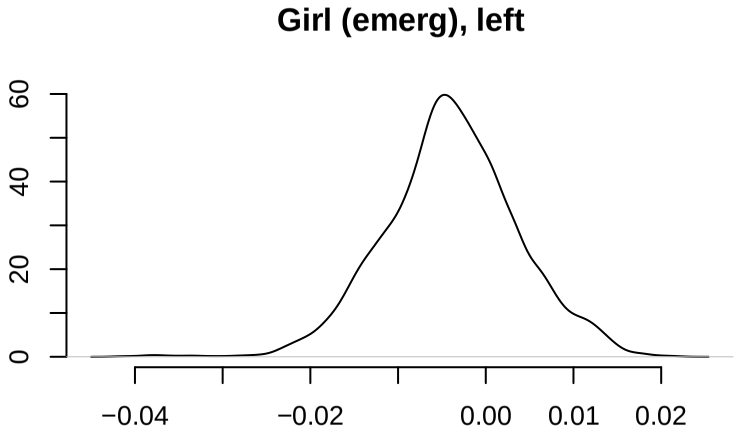
10.2 Traceplots

These can be drawn using the same code with the only change at the beginning.

```
> plfun <- "traceplot2"
> plname <- "trace"
```







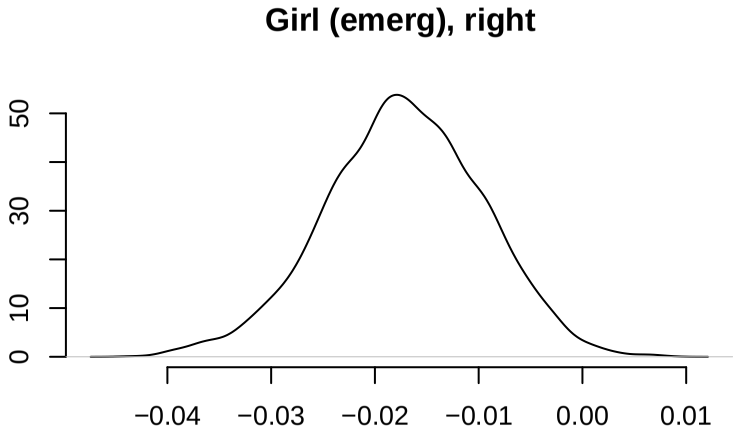


Figure 9: Estimate of the posterior distribution.

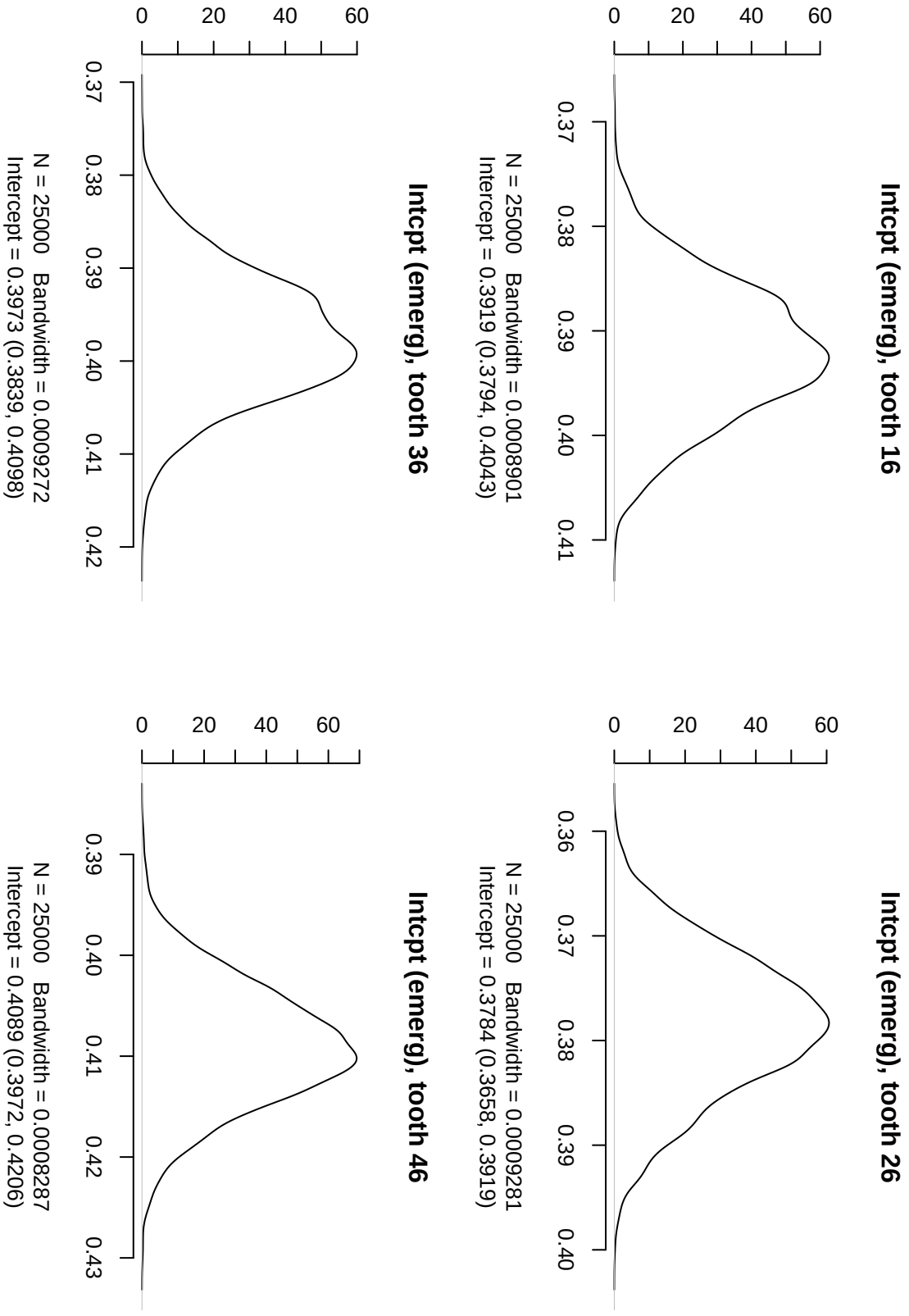


Figure 10: Estimate of the posterior distribution.

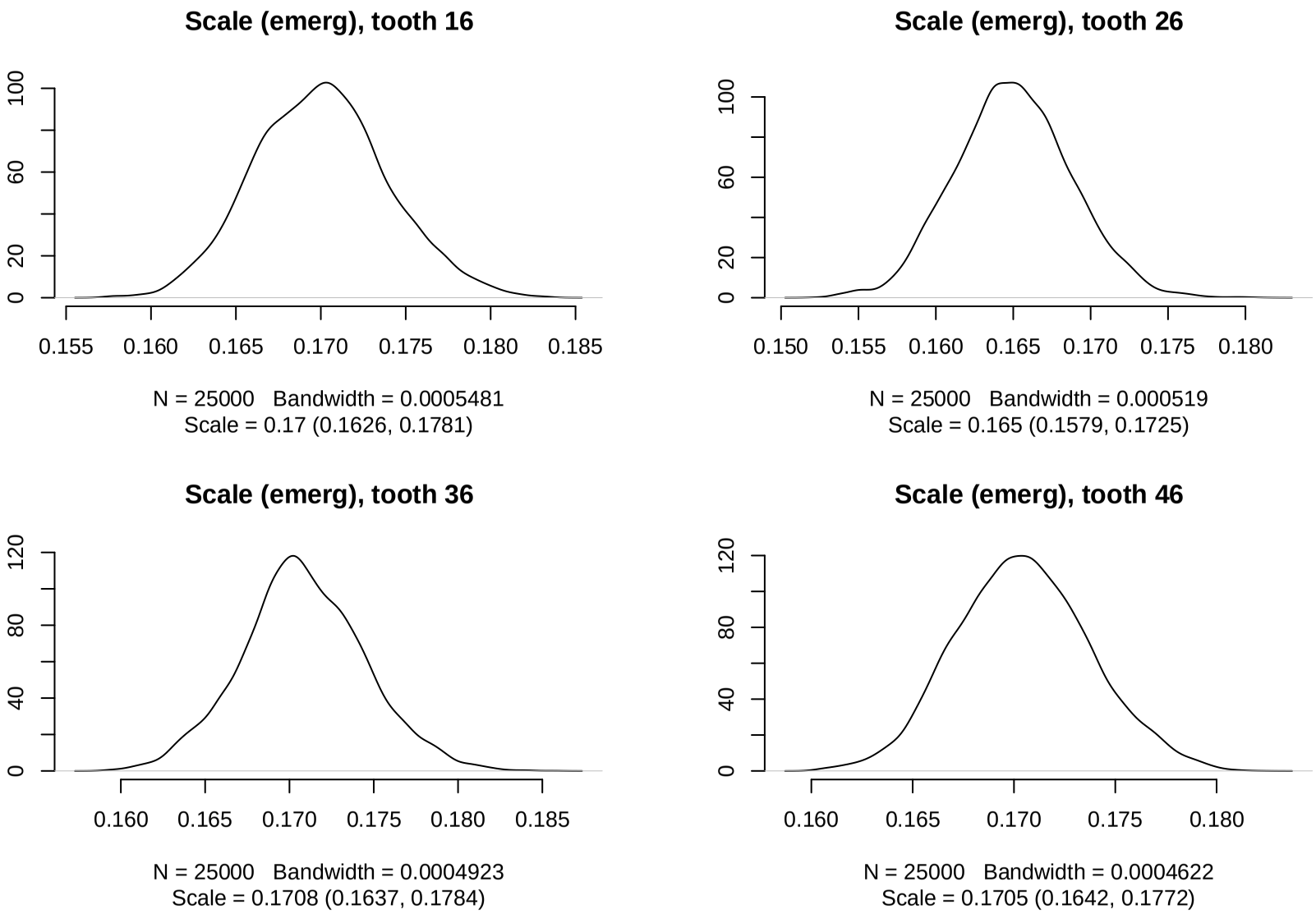


Figure 11: Estimate of the posterior distribution.

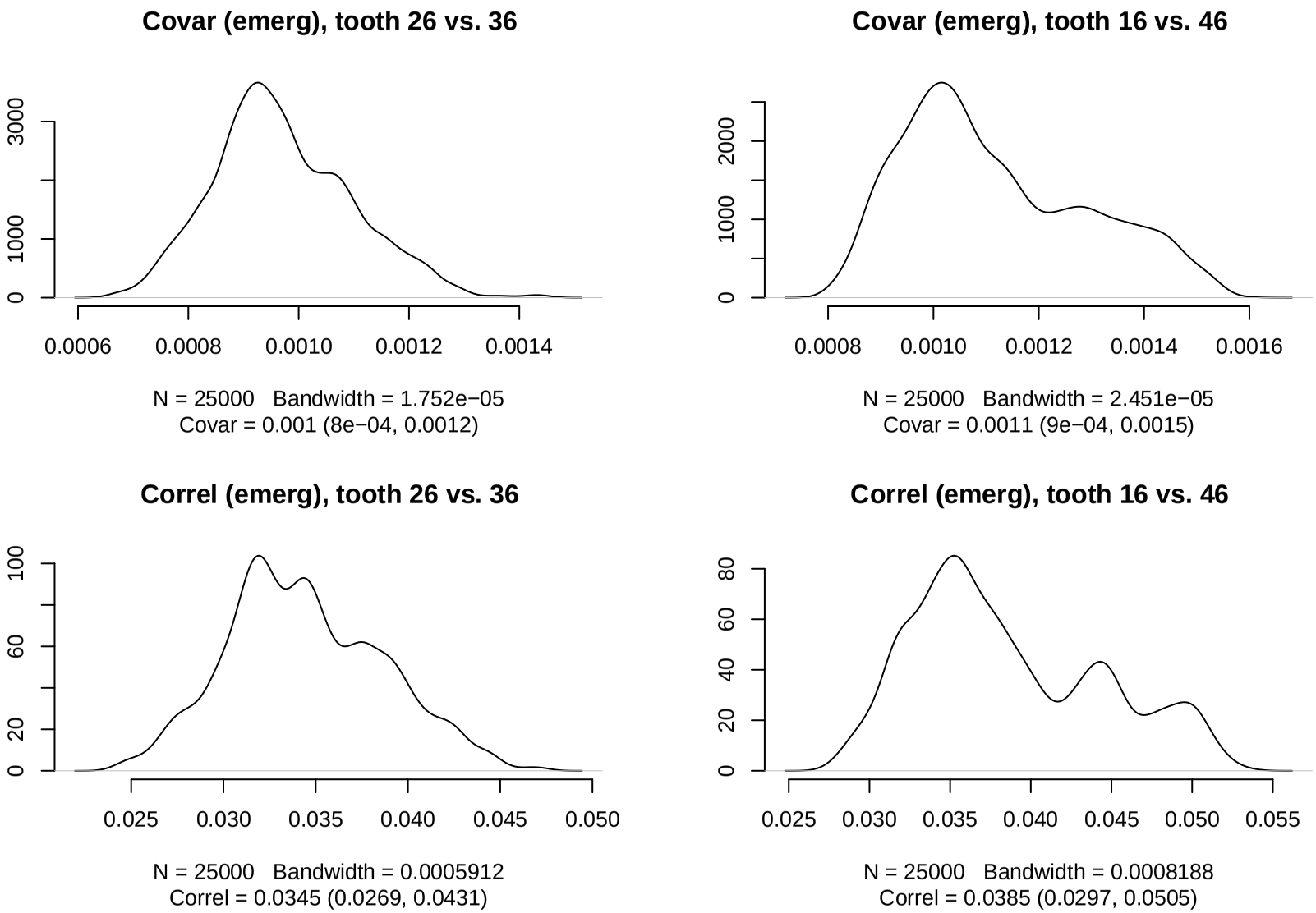


Figure 12: Estimate of the posterior distribution.

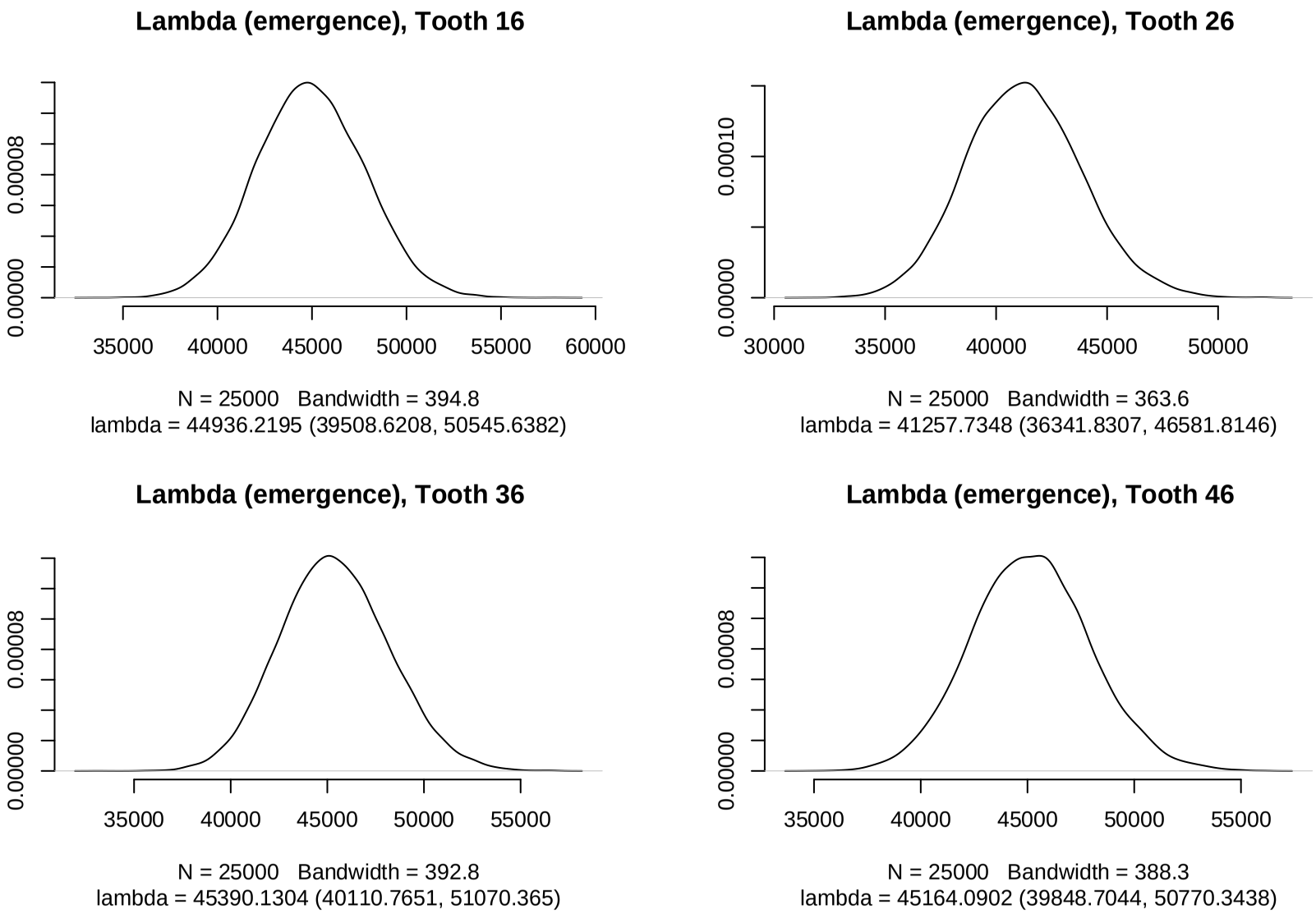
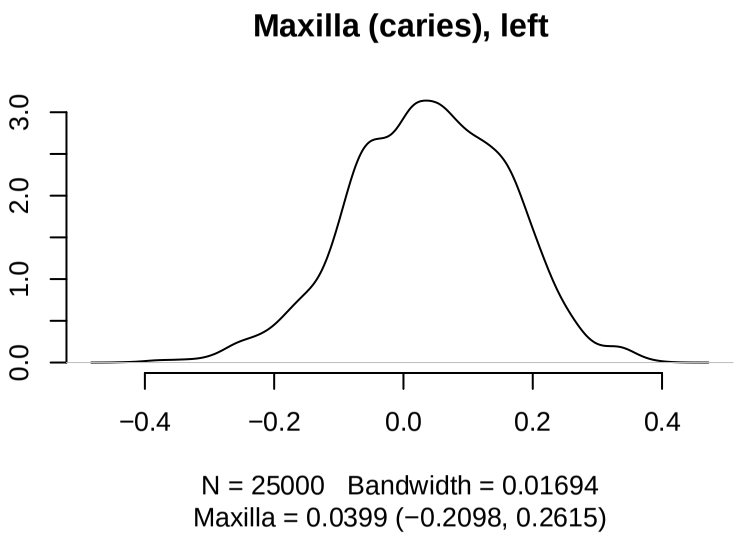
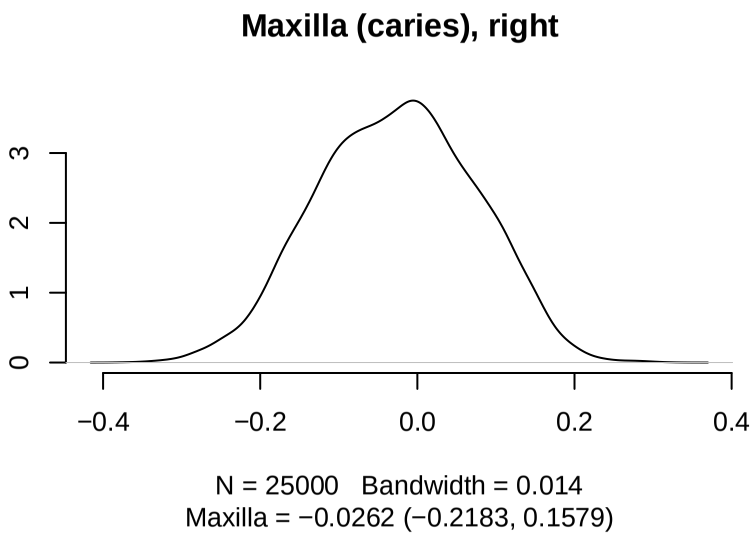
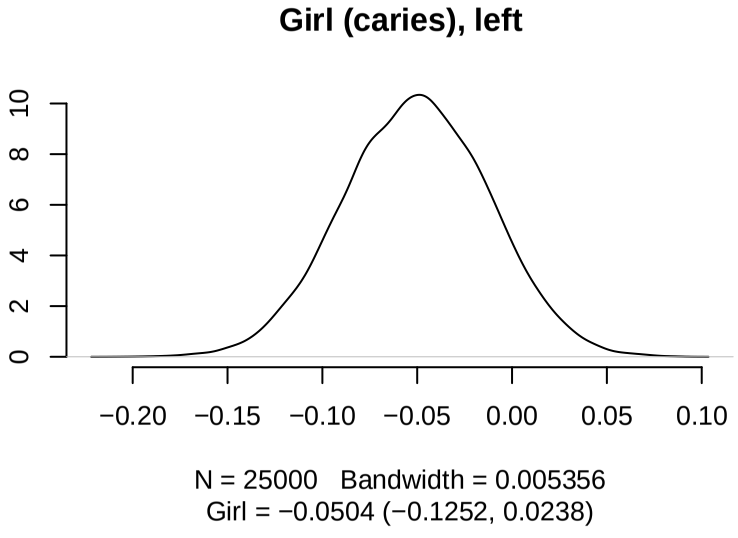


Figure 13: Estimate of the posterior distribution.







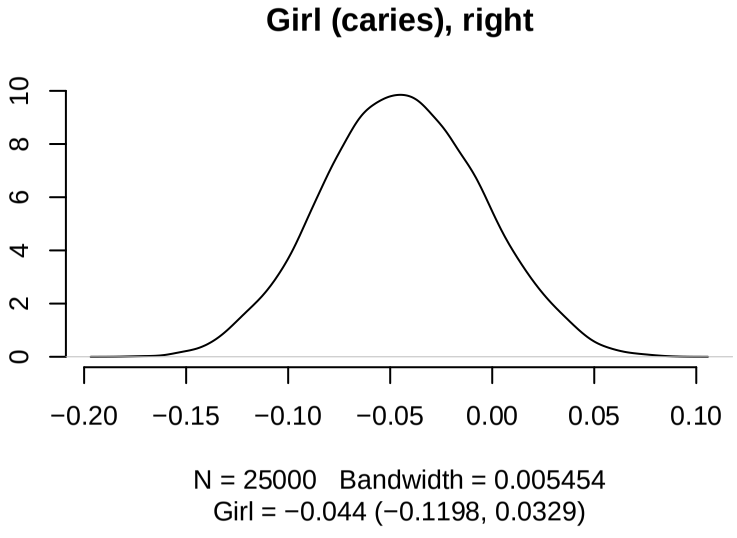


Figure 14: Estimate of the posterior distribution.

Figure 15: Estimate of the posterior distribution.

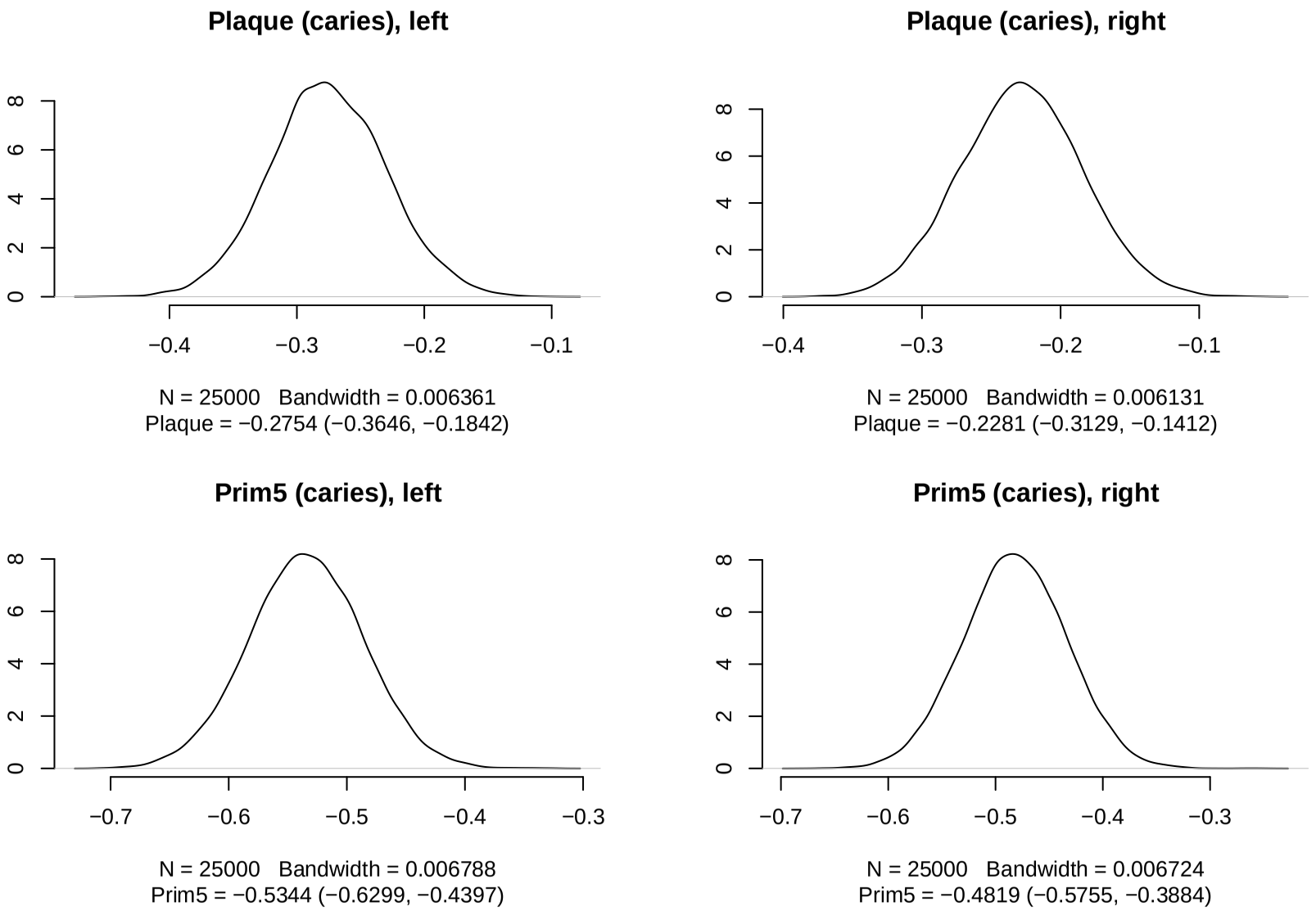
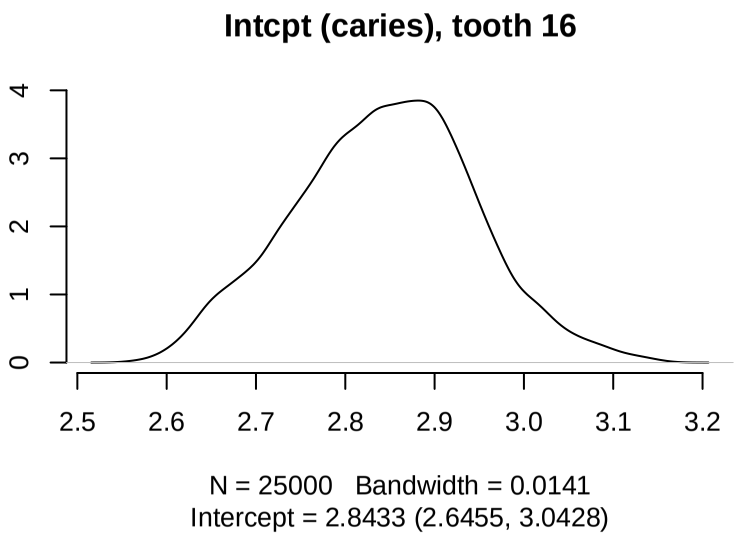
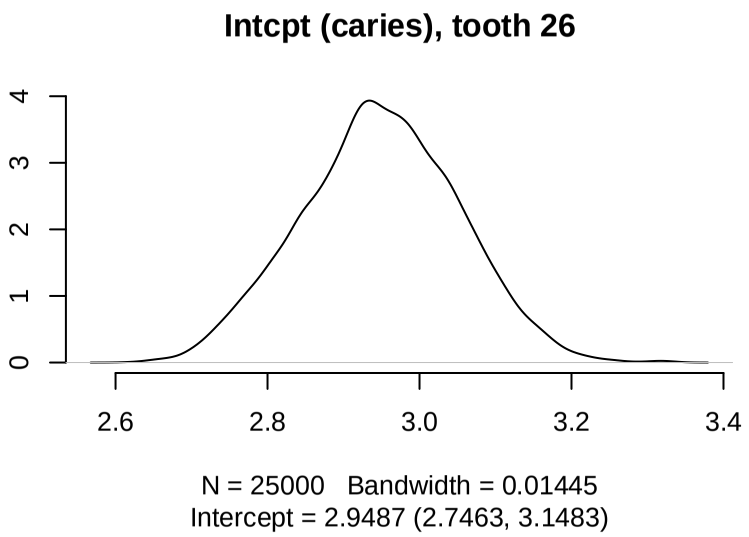
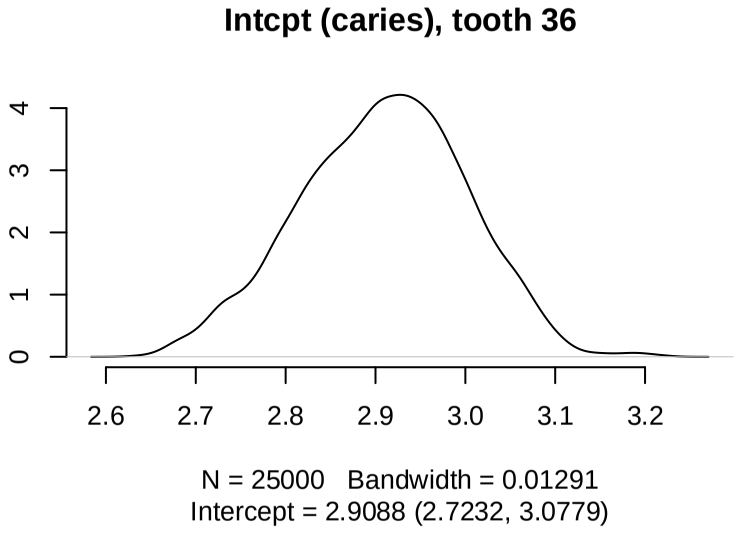


Figure 16: Estimate of the posterior distribution.







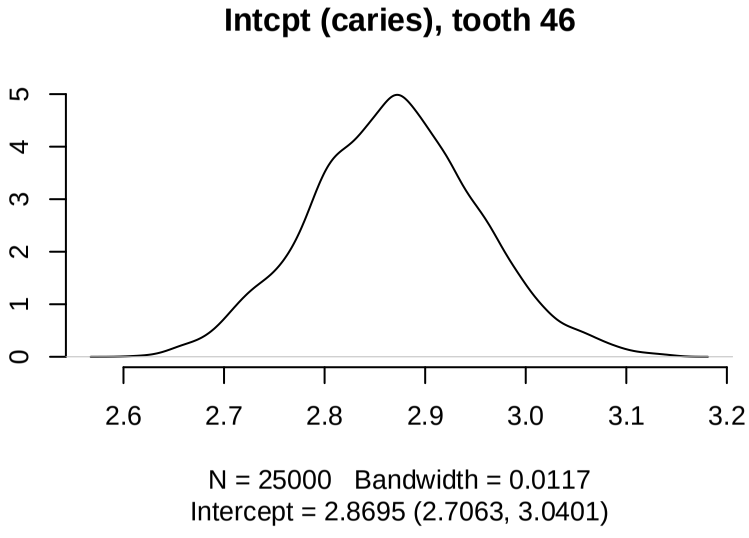
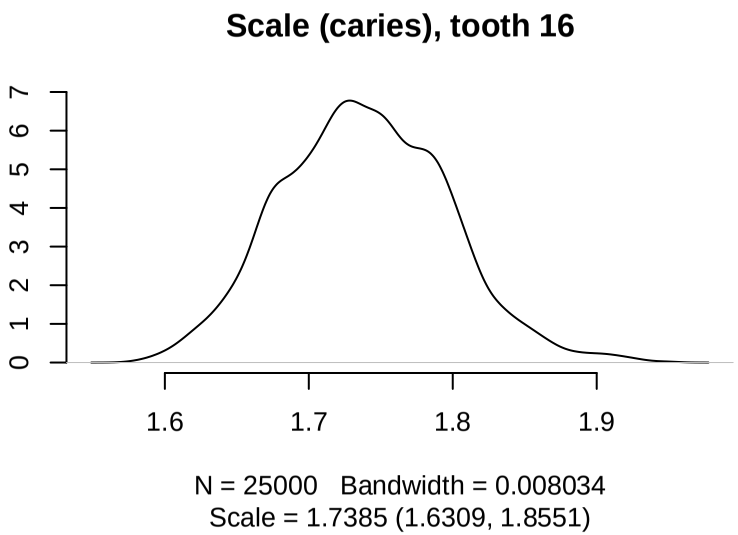
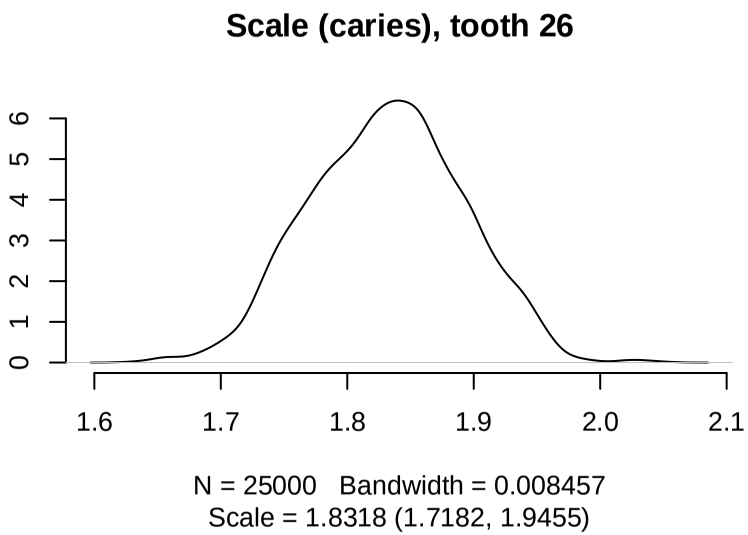
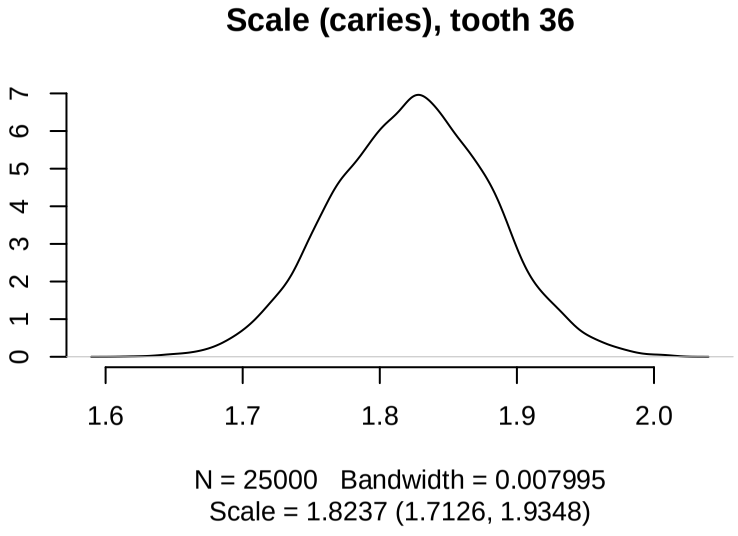


Figure 17: Estimate of the posterior distribution.







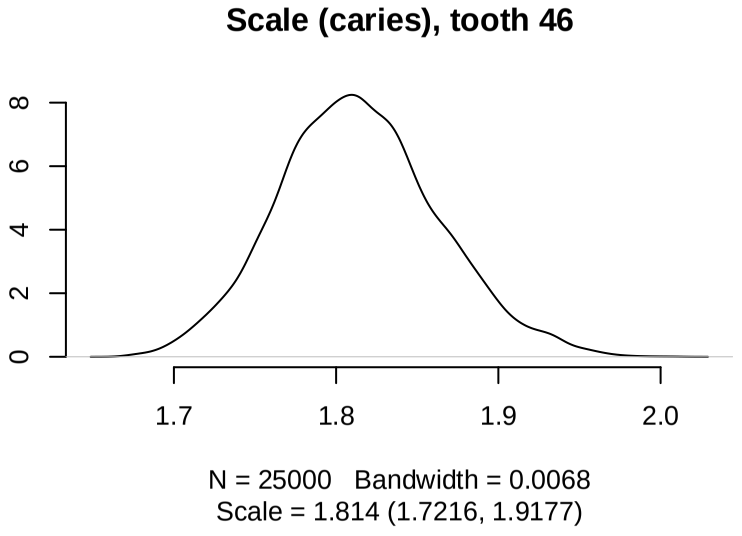
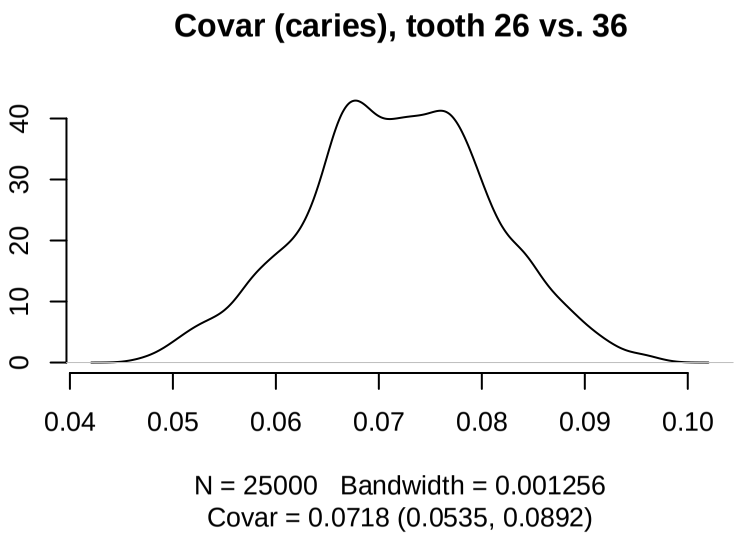
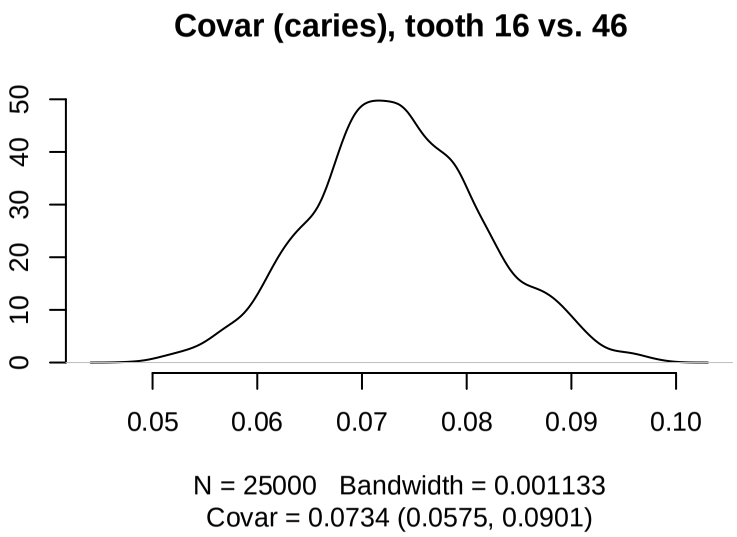
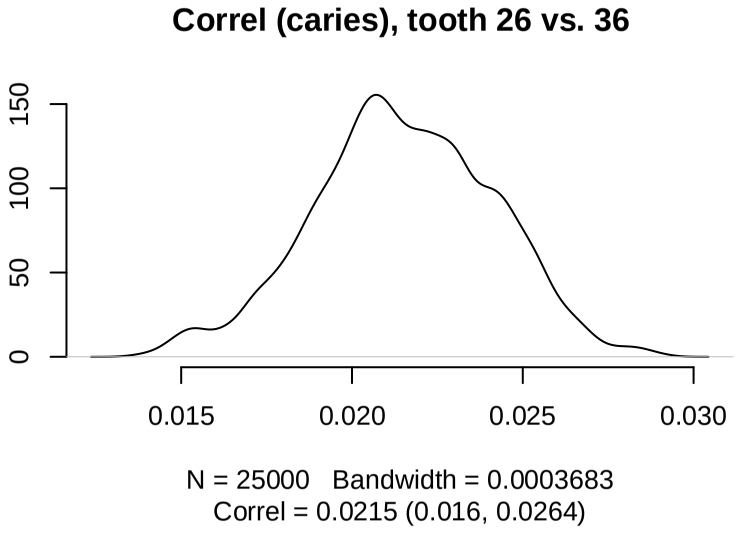


Figure 18: Estimate of the posterior distribution.







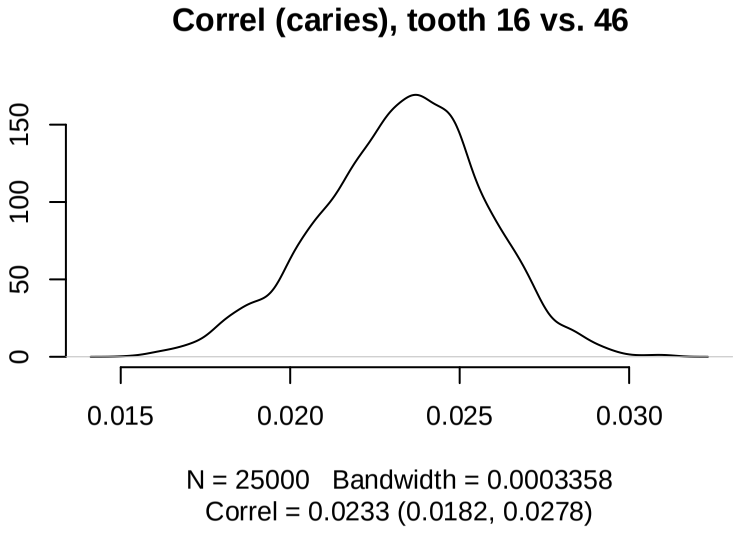
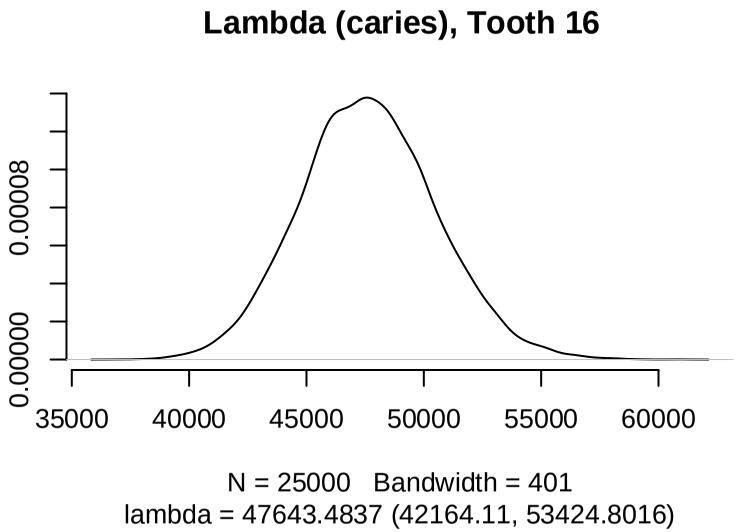
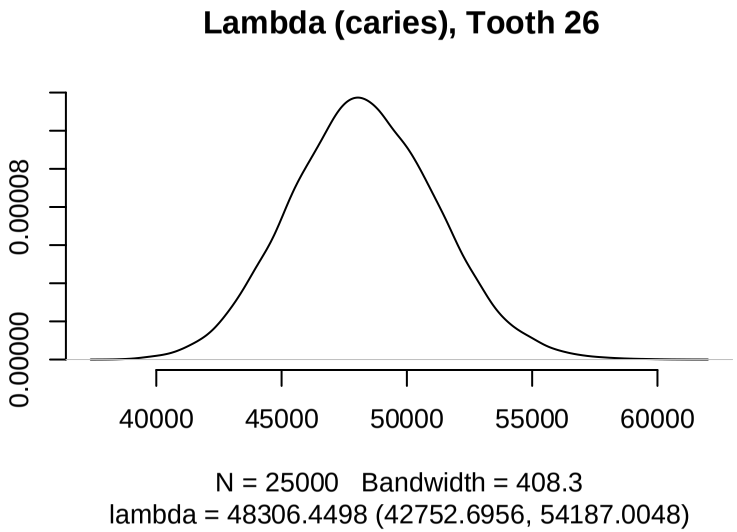
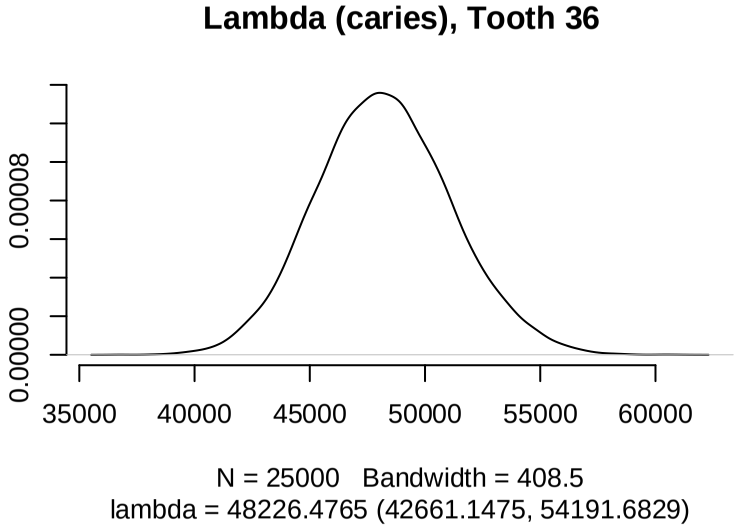


Figure 19: Estimate of the posterior distribution.







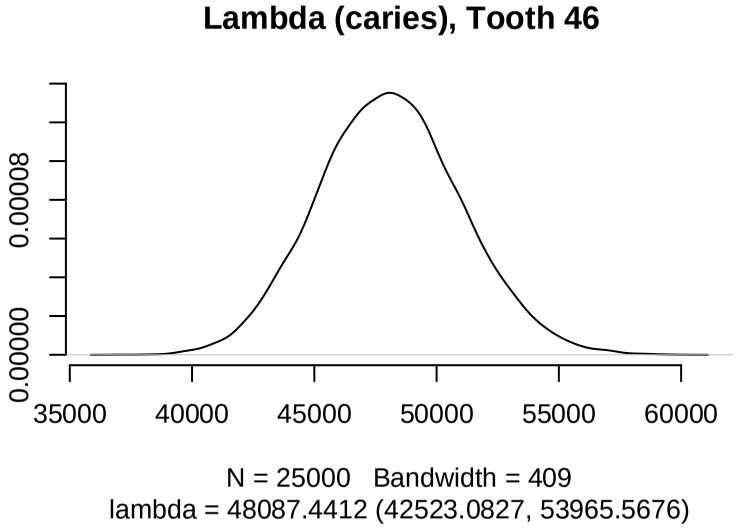


Figure 20: Estimate of the posterior distribution.