Repola J., Lindblad J., Heikkinen J. (2021). Pulpwood green density prediction models and sampling-based calibration. Silva Fennica vol. 55 no. 4 article id 10539. https://doi.org/10.14214/sf.10539

Supplementary file S1: Prediction of random week effects

To explain the details of constructing the predicted week effect $\hat{w}_{ij}$ for nationally calibrated prediction $\widehat{GD}_{ijk} = \hat{\boldsymbol{\beta}}\boldsymbol{x}_{ijk} + \hat{w}_{ij}$ of the green density of pulpwood parcel k delivered in week number j of year i, we must introduce some new notation. First, we "reverse time" and call the delivery day of parcel k "day 1," the previous day "day 2," and so on. In this terminology, we used sample measurements of the same assortment as parcel k from "days" 2 to 28 to construct the prediction. Let $\boldsymbol{y}$ denote the n-vector of the corresponding measured green densities and re-formulate MODELS 1 as

$$y_i = \mu_i + w_{j(i)} + e_i,$$

where i now refers to the elements of $\boldsymbol{y}$ (rather than year), μ_i is the prediction from the fixed part of the model, and $j(i)$ is the "reverse week number" from "day 1" backwards so that $j(i) = 1$ for samples from "days" 2 to 7, $j(i) = 2$ for samples from "days" 8 to 14, and so on. Thus, the week effects are no longer tied to calendar weeks, as they were in model fitting, but we assumed that the fitted covariance structure of the random effects remained valid for these "running weeks." This formulation allows the most efficient use of earlier sample measurements in prediction.

Since $j(i) \in \{1,2,3,4\}$, the reformulated model can be expressed in matrix notation as

$$\boldsymbol{y} = \boldsymbol{\mu} + \boldsymbol{Z}\boldsymbol{w} + \boldsymbol{e},$$

where $\boldsymbol{Z}$ is an $n \times 4$ dummy matrix with $\boldsymbol{Z}_{ik} = 1$, if the $j(i) = k$, 0 otherwise, and $w = [w_1\ w_2\ w_3\ w_4]'$. The empirical best linear predictor of random effect vector $\boldsymbol{w}$ is then (Mehtätalo and Lappi 2020, sec. 5.4.1)

$$\hat{\boldsymbol{w}} = \boldsymbol{D}\boldsymbol{Z}'(\boldsymbol{Z}\boldsymbol{D}\boldsymbol{Z}' + \boldsymbol{R})^{-1}(\boldsymbol{y} - \boldsymbol{\mu}),$$

where $\boldsymbol{D}$ is a 4×4 matrix with elements

$$\boldsymbol{D}_{jl} = \begin{cases} \widehat{\mathrm{var}}(w_j) = \hat{\sigma}_w^2, & l = j, \\ \widehat{\mathrm{cov}}(w_j, w_l) = \hat{\sigma}_w^2 \hat{\rho}^{|l-j|}, & l \neq j, \end{cases}$$

$\boldsymbol{R}$ is a $n \times n$ diagonal matrix with non-zero elements

$$\boldsymbol{R}_{ii} = \widehat{\mathrm{var}}(e_i) = \hat{\sigma}_{e,g(i)}^2,$$

and $g(i) \in \{1,2,3,4\}$ indicates one of the four groups with separate estimates of residual variance (delineated according to the delivery month and storage time).

Our prediction $\hat{w}_{ij}$ (in the original notation) was therefore the first element of $\hat{\boldsymbol{w}}$.

Using utilities available in base R (R Core Team 2020) and package nlme (Pinheiro et al. 2020), the following code was used to compute the predictions:

```
d$j <- factor(d$j,levels=1:4)
form <- formula(y~j-1)
Z <- model.matrix(form, model.frame(form, d))
D <- corMatrix(corAR1(corr_week), covariate=1:4)*var_week
R <- diag(d$var_res, nrow=nrow(d))
V <- Z %*% D %*% t(Z) + R
hat_w <- as.numeric(D[1,] %*% t(Z) %*% solve(V)) * (d$y-d$mu)
```

Here `d` is a data frame containing the n sample measurements used to predict the week effect for parcel k with columns `j` (the "reverse week number" counting down from the delivery day of parcel k), `y` (measured green density), `mu` (prediction from the fixed part of the model), and `var_res` (model-based residual variance $\hat{\sigma}^2_{e,g(i)}$). Scalars `corr_week` and `var_week` are respectively the estimated lag-1 autocorrelation $\hat{\rho}$ and variance $\hat{\sigma}^2_w$ of the week effects, and `hat_w` the predicted week effect, the first element of $\widehat{\boldsymbol{w}}$.

References

Mehtätalo L, Lappi J (2020) Biometry for forestry and environmental data: With examples in R. CRC Press LLC.

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2020) nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-148. URL https://CRAN.Rproject.org/package=nlme.

R Core Team (2020) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.