Web-based Supplementary Materials:
Multilevel Cross-dependent
Binary Longitudinal Data

Nicoleta Serban
H. Milton Stewart School of Industrial Systems and Engineering
Georgia Institute of Technology
nserban@isye.gatech.edu

Ana-Maria Staicu
Department of Statistics
North Carolina State University
ana-maria_staicu@ncsu.edu

Raymond J. Carroll
Department of Statistics
Texas A&M University
carroll@stat.tamu.edu

1Correspondent Author
Web Appendix A: Additional Insights for Section 3

In this appendix, we complement the discussion on the limitations of the linear approximation under rare-events binary data with a simulation study following the simulation settings of Hall et al. (2008) but under the rare-evens case. We refer to this supplemental material in Section 2 of the main paper.

To investigate how close the probabilities \( g\{X(t)\} \) are to a linear fit in the simulation study of Hall et al. (2008), we generated a large number of data sets according to their simulation settings. The simulation model is \( \text{pr}\{Y_r(t) = 1|X_r(t)\} = g\{X_r(t)\} \). We formed the 1st and the 99th percentiles of the probabilities \( g\{X(t)\} \) according to their model. We inverted the logistic function to get the range of these probabilities, and plotted the logistic distribution function against its best linear fit on this range (Figure 1 a); the fit is very linear indeed. We also plot the logistic distribution function and its best linear fit only on the interval where the probabilities \( g\{X(t)\} \) are between 0.005 and 0.075 (Figure 1 b). For this interval, the curvature is noticeable. Therefore, the simulation of Hall et al. (2008) only validates the use of the linear approximation under the non-rare event case.
Figure 1: (a) The logistic distribution function and the linear fit on the interval that includes the 1st to the 99th percentiles of the probabilities $g\{X(t)\}$ (b) The logistic distribution function and the linear fit on the interval where the function is between 0.005 and 0.075 in the simulation of Hall et al. (2008).
Web Appendix B: Model Component Estimation

Under the Linear Approximation

We refer to this supplemental material in Section 4 of the main paper. This supplemental material provides additional details on estimation procedures for the model components under the linear approximation.

B.1 Estimation of $\mu(\cdot)$

The marginal mean function $\alpha(t)$ can be estimated using nonparametric smoothing of the data $\{t_{rij}, Y_{ri}(t_{rij})\}$, and a working independence assumption. By neglecting the higher order terms, we estimate the mean $\mu$ as $\hat{\mu}(t) = g^{-1}\{\hat{\alpha}(t)\}$, where $\hat{\alpha}(t)$ is the smooth mean estimate described earlier and $g$ is the link function. This approach is similar to Hall et al. (2008).

B.2 Estimation of $\nu(\Delta)$

The covariance function $\nu(\Delta)$ quantifies the dependence between functional observations within the same subject, that correspond to units located at lag $\Delta$ apart. Estimation of $\nu(\Delta)$ entails a preliminary step, namely estimation of the within-unit correlation function $S_W(t, t', \Delta)$ under the assumption that the correlation between two functional observations within the same subject corresponding to design points that are far apart is negligible. More specifically, consider $\Delta^*$ to be the threshold beyond which functional measurements within the same subject, that correspond to design points with distance larger than $\Delta^*$ are uncorrelated; such value can be decided using the best available scientific information. Then $\nu(\Delta) \approx \{S_W^Y(t, t', \Delta^*) - S_W^Y(t, t', \Delta)\}/g'(\mu(t))g'(\mu(t'))$

One approach to construct a smooth estimator $\hat{S}_W^Y(t, t', \Delta)$ uses the methods of Staicu et al. (2010), which provides a consistent estimator of the within covariance for continuous functional data. We revisit this approach in the setting of rare events.
B.3 Estimation of \( K_Z(t, t') \) and \( K_W(t, t') \)

The next step is to estimate the covariance operators \( K_Z(t, t') \) and \( K_W(t, t') \) of the latent model components \( Z \) and \( W \). For this, we use the threshold \( \Delta^* \) defined above. Using equation (7) in the main paper, we have that

\[
\tilde{K}_W(t, t') = \tilde{S}_W(t, t', \Delta^*)/g'\{\hat{\mu}(t)\}g'\{\hat{\mu}(t')\} - \hat{\sigma}^2_u. \tag{1}
\]

Finally, an estimator of \( K_Z(t, t') \) can be determined using equation (5) in the main paper and substituting the estimators for the unknown parameters described earlier, so that

\[
\tilde{K}_Z(t, t') = \tilde{S}_W(t, t')/[g'\{\hat{\mu}(t)\}g'\{\hat{\mu}(t')\}] - \tilde{K}_W(t, t') - \hat{\sigma}^2_u. \tag{2}
\]

These estimators are symmetric; however they may not be positive semi-definite. To correct for this issue, the final estimators are obtained from the spectral decomposition of \( \tilde{K}_W \) and \( \tilde{K}_Z \) respectively, by ignoring the terms that correspond to negative eigenvalues; let \( \hat{K}_W \) and \( \hat{K}_Z \) be the final covariance estimators. Such an approach is fairly standard in the literature of functional data; see for example Yao et al. (2005).
Web Appendix C: Model Component Estimation

Under the Exponential Approximation

C.1 Estimation of $\nu(\Delta)$

Estimation of $\nu(\Delta)$ involves two preliminary estimation steps: 1) estimation of $\alpha(t)$ and 2) estimation of $E_{YB}(t, t', \Delta)$. In addition, it uses the assumption that $\nu(\Delta)$ becomes negligible when $\Delta$ exceeds a certain threshold $\Delta^*$. In the rare event case, $\alpha(t)$ can be consistently estimated via a global smoothing method obtained by fitting a logistic regression model to the data $\{t_{rij}, Y_{ri}(t_{rij})\}$; denote by $\hat{\alpha}(t)$ the resulting estimator. For numerical stability, we replace $\hat{\alpha}(t)$ by $\max\{c_1, \hat{\alpha}(t)\}$, where $c_1 > 0$ is a user-defined lower bound on the probabilities.

We approximate the covariance function $\nu(\Delta)$ by $e^{\nu(\Delta)} = \log\left\{\sum_{t} \sum_{t'} E_{YB}(t, t', \Delta) \right\} - \log\left\{\sum_{t} \sum_{t'} E_{YB}(t, t', \Delta^*) \right\}$. (3) which reduces to estimation of $E_{YB}(t, t', \Delta)$. In the estimation of the covariance function we assume that $E_{YB}(t, t', \Delta) = E_{YB}(t, t', \Delta^*)$ for $\Delta \geq \Delta^*$. We refer to this supplemental material in Section 5 of the main paper.

We now turn to the estimation of $E_{YB}(t, t', \Delta)$. Let $Y_r(t_{rij}, t_{r\ell k}, \Delta_{r\ell}) = Y_r(d_{r\ell}, t_{rij})Y_r(s_{r\ell}, t_{rij})$ for $i < \ell$, where $\Delta_{r\ell} = \|d_{r\ell} - s_{r\ell}\|$. For general $t, t'$ and $\Delta$, we have that $E[Y_r(t, t', \Delta)] = E_{YB}(t, t', \Delta)$. For $\Delta \geq \Delta^*$, the estimate $\hat{E}_{YB}(t, t', \Delta)$ is obtained by fitting a bivariate smoother to the data $\{t_{rij}, t_{r\ell k}, Y_r(t_{rij}, t_{r\ell k}, \Delta_{r\ell})\}$, where $\Delta_{r\ell} \geq \Delta^*$. For $\Delta < \Delta^*$, we employ a $k$-nearest neighbors method to estimate $E_{YB}(t, t', \Delta)$, a similar estimation method as introduced by Staicu et al. (2011). This method is computationally friendly as compared to the alternative method of fitting a trivariate smoother to the data $(t_{rij}, t_{r\ell k}, Y_{r\ell jk})$, where $Y_{r\ell jk} = \{Y_r(d_{r\ell}, t_{rij}) - Y_r(s_{r\ell}, t_{rij})\}\{Y_r(d_{r\ell}, t_{r\ell k}) - Y_r(s_{r\ell}, t_{r\ell k})\}/2$ for $r, i, \ell, j, k$ such that $\|d_{r\ell} - s_{r\ell}\| \geq \Delta^*$. This estimator is denoted by $\hat{E}_{YB}(t, t', \Delta^*)$.5
Specifically, for a fix $k > 1$, define the weights $w_{riℓ}(∆) = w_{k,riℓ}(∆) = 1\{Δ_{riℓ} ∈ N_k(∆)\}$, where $N_k(∆)$ is the subset of $k$th closest values to $∆$ among all the pairwise unit distances, and $Δ_{riℓ} = ∥d_{ri} - s_{rℓ}∥$. In addition $N_k(∆^*)$ is defined as the set of all the pairwise unit distances $(d_{ri}, s_{rℓ})$ such that $Δ_{riℓ} ≥ Δ^*$. An estimator of $E_YB(t, t', ∆)$ is defined by

$$e_{E_YB}(t, t', ∆) = ∑_{r,i} ∑_{i̸=ℓ} w_{riℓ}(∆)1\{Y_r(d_{ri}, t) = 1, Y_r(s_{rℓ}, t') = 1\}/∑_{r,i} ∑_{i̸=ℓ} w_{riℓ}(∆).$$

The estimator $e_{E_YB}(t, t', ∆)$ can be viewed as a kernel estimator of $E_YB(t, t', ∆)$, with moving kernel bandwidth, and thus it is a consistent estimator of $E_YB(t, t', ∆)$ for fixed $(t, t')$. The advantage of this approach is that the proposed covariance estimator for $ν(∆)$ does not depend on the marginal mean function estimator $b_α(t)$. The resulting estimator is consistent.

An estimator for $ν(∆)$ can be obtained by substituting $E_YB(t, t', ∆)$ for $∆ < Δ^*$ by $E_YB(t, t', ∆)$ and $E_YB(t, t', Δ^*)$ by $E_YB(t, t', Δ^*)$ in expression (3). Observe that $∫ ∫ E_YB(t, t', ∆) dt dt' ∝ exp\{ν(∆)\}$, where the proportionality is with respect to a finite constant. Using the dominated convergence theorem, since $|E_YB(t, t', ∆)| ≤ 1$ and the domain of $t$ is closed and bounded, it follows that $∑_t ∑_t' E_YB(t, t', ∆)/|(t, t'): t, t'|$ converges in probability to $∫ ∫ E_YB(t, t', ∆) dt dt'$ for all $∆$.

The estimator of $ν(∆)$ is consistent which leads to a consistent estimator for $ν(∆)$ if the approximation for the logistic link is accurate. The estimator $ν(∆)$ may not be positive semidefinite, and can be adjusted to be so as in Li et al. (2007). It follows that if $ν(∆)$ is the final covariance estimator, the variance $σ^2_u$ is estimated by $σ^2_u = ν(0)$.

One has to bear in mind that we cannot guarantee (asymptotic) unbiasedness of the estimator $ν(∆)$ since it is based on the approximation of the logistic link function. For example, under the rare case events, the exponential function is a good approximation of the logistic link function, which further ensures lower bias of this estimator - the better the approximation of the link function, the more accurate the estimator is.
C.2 Estimation of $K_Z(t, t')$, $K_W(t, t')$

Consider now the estimation of the covariance functions of the latent random components $Z_r$ and $W_{ri}$. Using equation (10) in the main paper, along with the assumption that $\nu(\Delta) \approx 0$ for $\Delta > \Delta^*$ we obtain an estimator for $K_Z(t, t')$ based on $\hat{E}_B'(t, t', \Delta)$; specifically $\hat{K}_Z(t, t') = \log \left\{ \hat{E}_B'(t, t', \Delta^*)/\hat{\alpha}(t)\hat{\alpha}(t') \right\}$, where $\hat{\alpha}(t)$ is the estimate of $\alpha(t)$ described earlier. This estimator is symmetric and it can be adjusted to be positive semi-definite, by using the corresponding technique outlined in the context of non-rare events.

Next we turn to the estimation of $K_W(t, t')$, which is based on equation (9) in the main paper requiring estimation of $E_T^Y(t, t')$. Let $Y_r(t_{rij}, t_{rik}) = Y_{ri}(t_{rij})Y_{ri}(t_{rik})$, where we suppressed the cross-dependence between units. An obvious consistent estimator for $E_T^Y$ is obtained by using a bivariate smoother, such as the thin plate spline smoother, of the data $\{t_{rij}, t_{rik}, Y_r(t_{rij}, t_{rik}) : r, i, j \neq k\}$ with an exponential or logistic canonical link function. Denote by $\hat{E}_T^Y(t, t')$ the estimator of $E_T^Y(t, t')$.

This estimator is symmetric; however the number of pairs of the form $\{Y_{ri}(t) = 1, Y_{ri}(t') = 1\}$, on which it relies may be insufficient for good accuracy. To improve the accuracy of the estimator of $E_T^Y(t, t')$, we combine the estimator above with a second one which uses the discordant pairs. Specifically, for $t \neq t'$ we have $\Pr\{Y_{ri}(t) = 1, Y_{ri}(t') = 0\} = \hat{\alpha}(t) - E_T^Y(t, t')$.

Furthermore to ensure symmetry of the estimator of $E_T^Y(t, t')$ using this approach, we use both types of discordant pairs $\{Y_{ri}(t) = 1, Y_{ri}(t') = 0\}$ and $\{Y_{ri}(t) = 0, Y_{ri}(t') = 1\}$. Let $\hat{E}_{1,T}^Y(t, t')$ be a smooth estimator of $[1\{Y_{ri}(t) = 1\} 1\{Y_{ri}(t') = 0\} : r, i]$ for $t < t'$ and using the logistic canonical link, and $\hat{E}_{2,T}^Y(t, t')$ be defined similarly corresponding to the data $[1\{Y_{ri}(t) = 0\} 1\{Y_{ri}(t') = 1\} : r, i]$ for $t < t'$. Then a symmetric estimator of $E_T^Y(t, t')$ is defined as:

$$\hat{E}_T^Y(t, t') = \{\hat{\alpha}(t) + \hat{\alpha}(t')\}/2 - \{\hat{E}_{1,T}^Y(t, t') + \hat{E}_{2,T}^Y(t, t')\}/2$$

Irrespective of the approach, we estimate $K_W(t, t')$ by $\hat{K}_W(t, t') = \log[\hat{E}_T^Y(t, t')/\{\hat{\alpha}(t)\hat{\alpha}(t')\}]$.
\( \hat{K}_Z(t, t') + \hat{\sigma}_u^2 \), where \( \sigma_u^2 = \hat{\nu}(0) \) is the variance of the cross-dependence process.

### C.3 Estimation of \( \mu(t) \)

Finally we estimate the mean function \( \mu(t) \) by employing equation (9) in the main paper. Using the above estimates for the mean function \( \alpha(t) \) and for the covariance functions \( K_Z(t, t'), K_W(t, t') \) and \( \nu(\Delta) \) we obtain an estimator of \( \mu(t) \) as a solution of (9). Specifically the estimator \( \hat{\mu}(t) \) is

\[
\hat{\mu}(t) = \log\{\hat{\alpha}(t)\} - \{\hat{K}_Z(t, t) + \hat{K}_Z(t, t) + \hat{\sigma}_u^2\}/2
\]

\[
-\log[1 + \hat{\alpha}(t) \exp\{\hat{K}_Z(t, t) + \hat{K}_Z(t, t) + \hat{\sigma}_u^2\}].
\]

This estimator is consistent provided the exponential approximation is accurate.
Web Appendix C: Additional Simulation Results

We complement the results in the main text presented in Section 6 with additional simulations and tables. In our notation, we observe response functional observation $Y_r(d_{ri})$ for fixed design points $d_{ri}, i = 1, \ldots, m_r$ for the $r$th subject. The design points are assumed to lie into the domain $D$ of dimensionality $dim$. In the main document, we presented and discussed simulation results for $dim = 1$. In this supplemental material we compare estimation accuracy results for $dim = 0, 1, 2$. The estimation accuracy is measured by the mean square error. The MSE for a function $f(t)$ and its estimator $\hat{f}(t)$ is computed as follows:

$$MSE(f) = \frac{1 + \int_T (\hat{f}(t) - f(t))^2 dt}{1 + \int_T f^2(t) dt}$$

We added the value one to both the nominator and denominator since $\int_T f^2(t) dt$ could be approximately equal to zero for some of the model components.

We include two additional tables presenting the MSE and its variance for the primary components in the model: $\mu(t)$, $\phi_k^Z(t)$, $\lambda_k^Z$, $\phi_k^W(t)$, and $\lambda_k^W$ for $k = 1, 2$. One table displays the MSE over 100 simulations under setting 1 (non-rare case) and the second table under setting 2 (rare case). Both tables provide accuracy results for both approximations, linear and exponential, and for three different structures of the cross-dependence between within subject functional observations, the simpler case when there is no cross-dependence ($dim = 0$) and the more difficult cases when there is spatial interdependence ($dim = 1$ or 2). These results show the reduction in the accuracy estimation when there is cross-dependence between within-subject observations for both approximation methods.
Table 1: **Setting 1**: Mean square error (average and variance in parentheses over 100 simulations) for the estimation procedures under linear and exponential approximations and for two interdependence structures, no dependence ($dim = 0$) and cross-dependence ($dim = 1$ or 2).

<table>
<thead>
<tr>
<th></th>
<th>$d = 0$</th>
<th>$d = 1$</th>
<th>$d = 2$</th>
<th>$d = 0$</th>
<th>$d = 1$</th>
<th>$d = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu(t)$</td>
<td>0.136(0.039)</td>
<td>0.145(0.018)</td>
<td>0.140(0.020)</td>
<td>0.11(0.063)</td>
<td>0.170(0.021)</td>
<td>0.158(0.039)</td>
</tr>
<tr>
<td>$\phi_1^Z(t)$</td>
<td>0.044(0.011)</td>
<td>0.051(0.017)</td>
<td>0.048(0.011)</td>
<td>0.074(0.015)</td>
<td>0.066(0.018)</td>
<td>0.068(0.010)</td>
</tr>
<tr>
<td>$\phi_2^Z(t)$</td>
<td>1.568(0.205)</td>
<td>1.833(0.02)</td>
<td>1.835(0.02)</td>
<td>0.643(0.332)</td>
<td>0.881(0.490)</td>
<td>0.783(0.46)</td>
</tr>
<tr>
<td>$\lambda_1^Z$</td>
<td>0.164</td>
<td>0.106</td>
<td>0.105</td>
<td>0.154</td>
<td>0.136</td>
<td>0.137</td>
</tr>
<tr>
<td>$\lambda_2^Z$</td>
<td>0.177</td>
<td>0.644</td>
<td>0.645</td>
<td>0.192</td>
<td>0.186</td>
<td>0.192</td>
</tr>
<tr>
<td>$\phi_1^W(t)$</td>
<td>0.038(0.001)</td>
<td>0.046(0.004)</td>
<td>0.047(0.005)</td>
<td>0.079(0.002)</td>
<td>0.049(0.005)</td>
<td>0.049(0.006)</td>
</tr>
<tr>
<td>$\phi_2^W(t)$</td>
<td>0.075(0.023)</td>
<td>0.071(0.011)</td>
<td>0.072(0.010)</td>
<td>0.078(0.121)</td>
<td>0.079(0.017)</td>
<td>0.081(0.011)</td>
</tr>
<tr>
<td>$\lambda_1^W$</td>
<td>0.023</td>
<td>0.035</td>
<td>0.034</td>
<td>0.023</td>
<td>0.035</td>
<td>0.035</td>
</tr>
<tr>
<td>$\lambda_2^W$</td>
<td>0.01</td>
<td>0.009</td>
<td>0.010</td>
<td>0.008</td>
<td>0.011</td>
<td>0.012</td>
</tr>
</tbody>
</table>
Table 2: **Setting 2**: Mean square error (average and variance in parentheses over 100 simulations) for the estimation procedures under linear and exponential approximations and for two interdependence structures, no dependence \((dim = 0)\) and cross-dependence \((dim = 1 \text{ or } 2)\).

<table>
<thead>
<tr>
<th></th>
<th>Linear</th>
<th></th>
<th></th>
<th></th>
<th>Linear</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(d = 0)</td>
<td>(d = 1)</td>
<td>(d = 2)</td>
<td></td>
<td>(d = 0)</td>
<td>(d = 1)</td>
<td>(d = 2)</td>
</tr>
<tr>
<td>(\mu(t))</td>
<td>0.001(0.0003)</td>
<td>0.002(0.0002)</td>
<td>0.001(0.0001)</td>
<td></td>
<td>0.0002(0.0001)</td>
<td>0.0003(0.0000)</td>
<td>0.0003(0.0000)</td>
</tr>
<tr>
<td>(\phi_1^Z(t))</td>
<td>0.071(0.048)</td>
<td>0.067(0.411)</td>
<td>0.061(0.032)</td>
<td></td>
<td>0.077(0.032)</td>
<td>0.075(0.028)</td>
<td>0.076(0.026)</td>
</tr>
<tr>
<td>(\phi_2^Z(t))</td>
<td>1.268(0.301)</td>
<td>1.808(0.021)</td>
<td>1.810(0.108)</td>
<td></td>
<td>0.626(0.367)</td>
<td>0.745(0.402)</td>
<td>0.688(0.433)</td>
</tr>
<tr>
<td>(\lambda_1^Z)</td>
<td>0.037</td>
<td>0.040</td>
<td>0.044</td>
<td></td>
<td>0.072</td>
<td>0.107</td>
<td>0.133</td>
</tr>
<tr>
<td>(\lambda_2^Z)</td>
<td>0.185</td>
<td>0.223</td>
<td>0.324</td>
<td></td>
<td>0.110</td>
<td>0.111</td>
<td>0.100</td>
</tr>
<tr>
<td>(\phi_1^W(t))</td>
<td>0.266(0.152)</td>
<td>0.272(0.164)</td>
<td>0.240(0.148)</td>
<td></td>
<td>0.163(0.114)</td>
<td>0.183(0.138)</td>
<td>0.189(0.138)</td>
</tr>
<tr>
<td>(\phi_2^W(t))</td>
<td>0.336(0.181)</td>
<td>0.378(0.158)</td>
<td>0.361(0.154)</td>
<td></td>
<td>0.281(0.470)</td>
<td>0.380(0.467)</td>
<td>0.378(0.137)</td>
</tr>
<tr>
<td>(\lambda_1^W)</td>
<td>0.085</td>
<td>0.126</td>
<td>0.117</td>
<td></td>
<td>0.122</td>
<td>0.242</td>
<td>0.550</td>
</tr>
<tr>
<td>(\lambda_2^W)</td>
<td>0.268</td>
<td>0.214</td>
<td>0.126</td>
<td></td>
<td>0.145</td>
<td>0.351</td>
<td>0.355</td>
</tr>
</tbody>
</table>
Web Appendix D: Additional Figures - LIDAR Data

This Appendix supplements the Figures in Section 7 of the main manuscript with comparing plots from the group means and eigenfunctions as estimates using the linear approximation. We also refer to this supplemental material in Section 8 of the main paper.

Figure 2: Displayed are: (a) estimated group mean functions of $\mu_g(t)$ for the control group ($g = 0$) and treatment group ($g = 1$), using the linear approximation; (b) the estimated correlation function varying with the distance between locations/bursts; (c) the first three estimated eigenfunctions of the between-covariance function $K_Z(t, t')$; and (d) the first estimated eigenfunction of the within-covariance function $K_W(t, t')$. 

12
Web Appendix E: R Implementation

With this Web Appendix we include R software codes implementing the methodology in the paper for \( \text{dim} = 0, 1, 2 \) and using the two approximations. The appended .zip directory consists of three folders, a folder for each value of \( \text{dim} \), and each folder contains two sub-folders, one for each approximation. The denotation of the folders and sub-folders is self-explanatory. In this appendix, we consider one specific example of our implementation to illustrate how the R software codes can be used. Specifically, we consider a particular example, estimating the model using exponential approximation where the domain of the unit design points has dimensionality \( \text{dim} = 2 \).

**Step 1.** Prepare the data in a matrix format of \( R \times M \times N \) rows and eight columns, where each column represents

- Column 1: Group ID (if the data are not grouped then this column will consist of 1’s only)
- Column 2: Subject ID taking values in 1, \ldots, \( R \) where \( R \) is the number of subjects
- Column 3: Unit ID for each subject taking values in 1, \ldots, \( M \) where \( M \) is the number of units per subject
- Column 4 & 5: The design points which are two-dimensional, first dimension in column 4 and the second dimension in column 5. (This is the case \( \text{dim} = 2 \))
- Column 6: Sub-unit ID taking values in 1, \ldots, \( N \) where \( N \) is the number of time points per one spatial unit for each subject (In our implementation, we assume the same \( N \) time points across subjects.)
- Column 7: Time (or other functional form) points converted to a \( (0, 1) \) scale
- Column 8: The binary observation for each subject at the corresponding unit and sub-unit converted in 0 and 1 values

**Step 2.** Apply the estimation function. Source the following to files

source("exponential/utilities_MFD.R")
source("exponential/fit_MFD_binary.R")

In order to obtain data specifics that will be used as an input, you may use the following R commands

data.specifics = fn.response.matrix(DATA)

## the binary observations in a matrix of dimensions N X (RM)
response.matrix = data.specifics$response.matrix
Rd = data.specifics$Rd
D = data.specifics$D
R = data.specifics$R
M = data.specifics$M
N = data.specifics$N
timepoints = DATA[,7]
spacepoints = cbind(DATA[,4],DATA[,5])

Other input variables are as follows
– ‘spacedist’ which are the distances between the unit design points across all subjects
– ‘size.bin’ is the number of the closest unit design points to a given unit design point used in defining its closest neighbors in estimating the correlation function
– ‘delta.star’ which is the value corresponding to $\Delta^*$ in the paper and it is the distance value at which the cross-correlation is truncated
– ‘delta.trunc’ is a value larger (not with much) than $\Delta^*$ but smaller than the maximum distance between any two unit design points; it is used in the smoothing of the cross-correlation function
– ‘delta.output’ a sample of distances at which we output the estimate of the correlation function

A sample of R code used in our simulation study is provided below

14
### INPUTS as described above

\[
\text{spacedist} = \text{fn.unit.distsances}(\text{DATA}, \text{Rd}, \text{M})
\]

\[
\text{size.bin} = 150
\]

\[
\text{delta.max} = \max(\text{spacedist})
\]

\[
\text{delta.star} = \text{quantile}(\text{spacedist}, .5)
\]

\[
\text{delta.trunc} = \text{quantile}(\text{spacedist}, .6)
\]

\[
\text{delta.output} \leftarrow \text{sort}(\text{sample}(\text{spacedist}, 1000, \text{replace} = \text{FALSE}))
\]

##### Estimation function

\[
\text{Z.W.eigen.decom} = \text{fn.eigen.decom.Z.W}(\text{DATA}, \text{response.matrix}, \text{timepoints}, \text{spacepoints}, \text{Rd}, \\
\text{delta.star} = \text{delta.star}, \text{delta.trunc} = \text{delta.trunc}, \text{delta.output} = \text{delta.output}, \text{bound} = 0.0005^2, \text{size.bin})
\]

#### Step 3. Obtain the output consisting of

- Estimate of the mean function \( \mu \) evaluate at the observed time points: 'mu'
- Estimate of the correlation function evaluate at the distances specified by the 'delta.output': 'corUS'
- Estimate of the variance \( \sigma_s \): 'varU'
- Estimate of the level-1 eigenfunctions (for the covariance function of the process \( Z \)): 'phi1'
- Estimate of the level-1 eigenvalues (for the covariance function of the process \( Z \)): 'lambda1'
- Estimate of the level-2 eigenfunctions (for the covariance function of the process \( W \)): 'phi2'
- Estimate of the level-2 eigenvalues (for the covariance function of the process \( W \)): 'lambda2'
- Percent of the variance explained by the level-1/level-2 functional components: 'percent1'/'percent2'

## mean function

\[
\text{mu.est} = \text{Z.W.eigen.decom}$\text{mu}
\]

## spatial covariance
corU = Z.W.eigen.decom$corUS
varU = Z.W.eigen.decom$varU
print("Spatial Variance Estimate")
print(varU)

## eigenfunctions if only two of them are selected
fit.phi1 = Z.W.eigen.decom$phi1
fit.phi2 = Z.W.eigen.decom$phi2

## eigenvalues if only two of them are selected
fit.lambda1 = Z.W.eigen.decom$lambda1
print("Estimates for level-1 Eigenvalues")
print(fit.lambda1)
fit.lambda2 = Z.W.eigen.decom$lambda2
print("Estimates for level-2 Eigenvalues")
print(fit.lambda2)

Additional README files are available in each of the three folders.