BIOS 7720 Final Project: Explorations and Experiments of NHANES data Functional Data Analysis with INLA

Randy

April 20, 2021

# Introduction

Functional Data Analysis (FDA) deals with the analysis and theory of data that are in the form of functions. On the other side, FPCA is also a powerful reduction tool and dealing with imputation with functional data that are sparsely observed. However most often the functional data are regarded as intrinsically infinite dimensional. It is both a challenge and a chance for data analysis. By imposing a smoothness assumption, we can “borrow” information from neighboring spatial or temporal measurement, which can be pooled to overcome the curse of dimensionality. Thus, smooth function is also a tool for data regularization. For this project, the NHANES dataset was analyzed. NHANES is one of the largest and most important studies in terms of size, scope, diversity, and accessibility of the data.

# Method

INLA is based on the Laplace approximation. This method approximates the integrand with a second order Taylor expansion around the mode for a Gaussian approximation. Currently, multiple researching groups are developing a nested version of this Laplace approximation, combined with modern numerical techniques to get *Integrated Nested Laplace Approximations* (INLA). INLA is a powerful tool for approximate Bayesian inference for *latent Gaussian models* (LGMs). In this project, the author mainly explored the application of R-INLA and brinla package for functional data analysis.

# Result

## 1. A Simple Simulation Study

In functional data analysis, the optimization of penalized likelihood estimate can be set using cross-validation. In Bayesian framework including the penalty term is equivalent to setting a specific prior on the covariate coefficients. For equidistant knots {} the prior is on the differences, which is equivalent to setting a RW1 prior on :

Alternatively, when the prior is on the second order differences, which is equivalent to setting a RW2 prior on the coefficients:

Hence, latent effects RW1 and RW2 can be used to include smooth terms on the covariates. The elements of the latent effect represent the values of the smooth term (details in Appendix Section 5). As we can see in Figure1, in general Overall, the resulting posterior distributions do not differ much from those given by the default prior (Figure2).

## 2. A practice modeling TLAC with INLA

Average daily total activity count (TAC) and the average daily total log activity count (TLAC) were analyzed with both mgcv and INLA. The purpose of this section is to confirm the INLA package and related functions can be applied. Overall this section proved that with proper modeling setting, INLA can provide

## 3. A Failure on the 5-year-mortality prediction

After several failed attempts on using high dimensional smoothing in Bayesian framework. Two Scalar on function regressions (SoFR) are applied. The results are showed in both Figure3 and the Appendix Code section. The FPCA were applied for both original scale and truncated binary scale, while adjusting for Age and Gender. Further model specification and model selection need to be applied, and more theoretical and practical knowledge is required for future model fitting.

## 4. An exploration on mgcv::ginla()

This is still an experimental function in mgcv package. Currently not all steps are optimally efficient and written for relatively expert users. The overall goal for this function is to apply INLA (Rue et al. 2009) to models estimable by gam() or bam(), using the INLA variant (Wood 2019). As show in the code section, the marginal posterior densities for each coefficient, selected coefficients or linear transformations of the coefficient vector are provided. The package author commented as “for many models the INLA estimates are very close to the usual Gaussian approximation to the posterior” (Wood 2019). No further analysis will be devoted into ginla(), until a better version of this function being published.

# Discussion

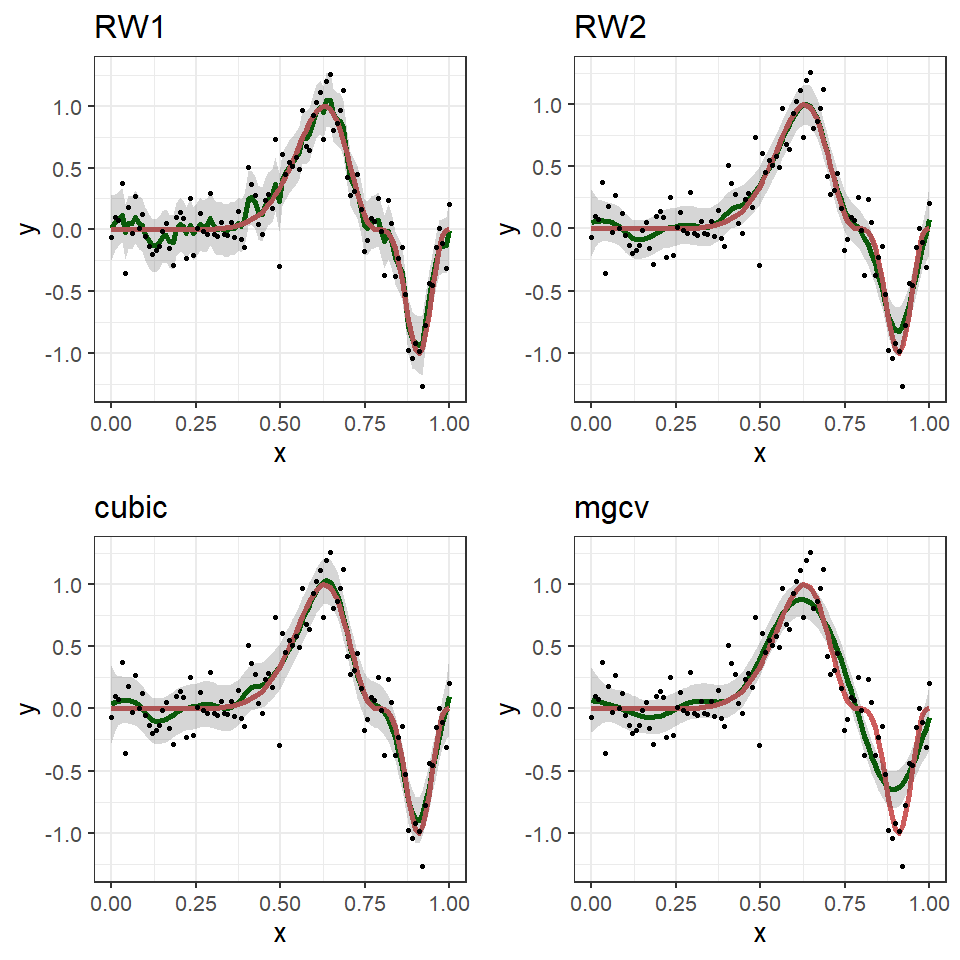
In this project the author explored a new methodology with INLA and Bayesian smoothing methods. More work needs to be done on both theoretical and practical levels. The major issues are in the areas of NHANES data structure manipulation. Even though the NHANES data is publicly available, the actual analyzing requires much more background information and data process, which beyond the ability in such limited time window. Another issue for this project is the limited knowledge on the Bayesian smoothing, more sophisticated higher dimensional smoothing need to be applied for the SoFR model at the end, especially the thin plate smoothing Bayesian methods, such as SPDE, or Besag. Several attempts have been tried in this project on fitting the model with RW2D and SPDE smoothing. However due to the unfamiliarity of the data structure and overlay grid structures, there are multiple fatal errors in INLA model.

Here are a few perspectives worthy exploring in the future. Of course, first the RW2 and RW2D smoothing should be remodeled with after better examination. For regular lattices data, the RW2D model can be functional and appropriate, however for irregularly space location, a more flexible model which applies thin-plate spline smoothing is required. To obtain a thin plate spline estimator, studies have provided the solution following stochastic partial differential equation (SPDE). The SPDE is solved by a finite element method on a triangular mesh, and the resulting thin-plate spline (TPS) prior has a multivariate normal density with mean zero and precision matrix , a highly sparse matrix due to the local nature of basis functions. This TPS prior will be a generalized extension for RW2D prior. However, this requires deeper knowledge about the spatial modeling and data manipulation, like point pattern overlay, which is beyond the purpose of this project in short terms. Hence both RW2D and SPDE will be further explored in the future.

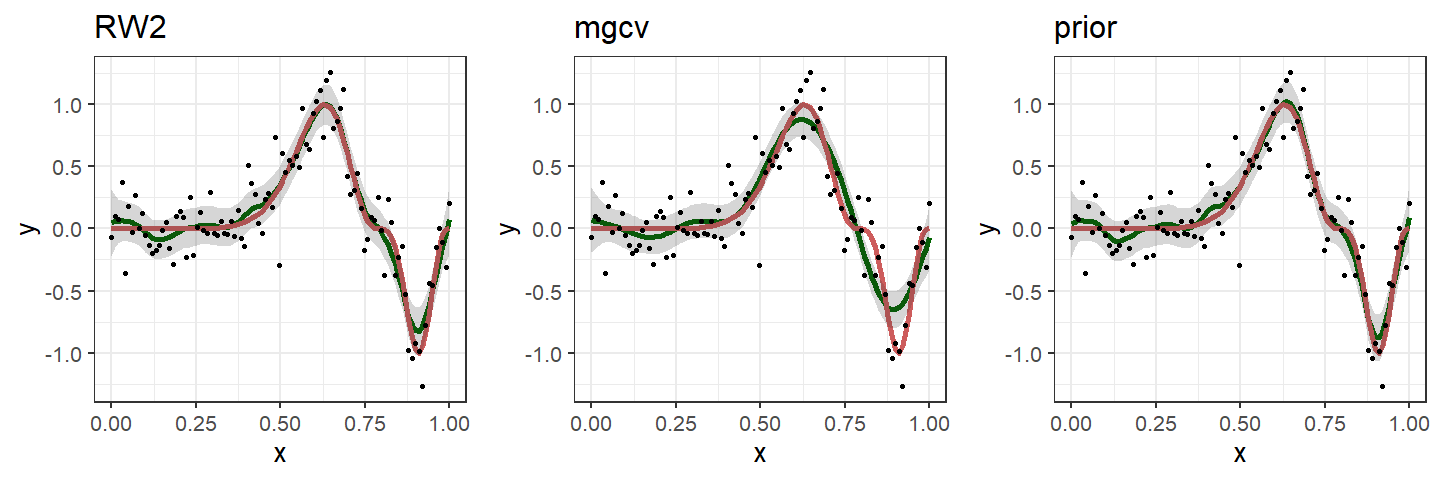
Previous research indicated that a directly application of FPCA on binary data accomplished similar results as LOESS. LOESS is one of the early and popular smoothing functions. The idea of LOESS is based on local polynomial ideas, which applies the least-squares. However, if the direct FPCA application on binary data is indeed equivalent to LOESS, the fit would also be potentially affected by influential samples or outliers (based on least squares). More questions need to be answered on whether a heavy tailed t-distribution kernel fitting is also sensitive in this case, compared with the Tukey’s distribution. Whether more robust functional analytic tools or quantile smoothing spline, such as the R functions qss() and rqss() from the package quantreg (Koenker 2017), or rgam (Salibian-Barrera 2014) and robustgam.

Overall, this project is a failure on every level. The fatal outcomes of this project resulted from multiple reasons. More theoretical and practical knowledge is needed before further model fitting. The final five-year-mortality models were built without INLA or Bayesian smoothing and attached in the appendix. Moreover, as mentioned earlier, more and more researching groups are trying to integrate INLA methods into functional data analysis, such as the ginla() function in mgcv package (Wood 2021).

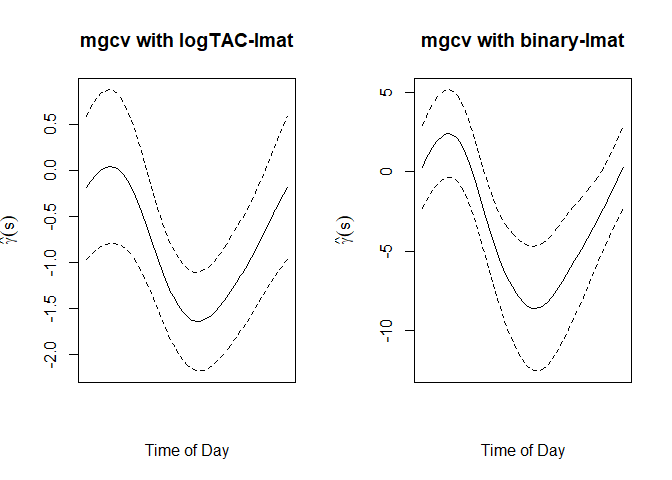
# Figures



**Figure1: A simple simulation study on the random walk smoothing (RW1 and RW2), cubic spline, and mgcv b-spline.** According to the plots, the RW1 and RW2 can provide efficient smoothing (green lines) over the simulated data (red lines), compared with mgcv function.



**Figure2: A simple simulation study on the RW2, mgcv, and informative prior.** Overall the resulting posterior distributions do not differ much from those given by the default priors.

****

**Figure3: Five-year-mortality model fitting results.** There is no corresponding INLA model, due to the complexity of NHANES data structure. Future study will be focused on the modeling fitting with counterpart smoothing like RW2D, or SPDE.

# Appendix

## 1. INLA Integrated Nested Laplace Approximation

Integrated Laplace Approximation  
Introduced by Rue, Martino and Chopin (2009).  
Posteriors are estimated using numerical approximations.

It is a deterministic approach to approximate  
Bayesian inference for latent Gaussian models (LGMs)  
INLA is both faster and more accurate than MCMC

Three key components required by INLA:

the LGM framework

the Gaussian Markov random field (GMRF)

the Laplace approximation

## 2. Latent Gaussian Models (LGM) framework

LGMs have a wide-ranging list of applications and most structured Bayesian models

Regression models, the most extensively used subset of LGMs.

Dynamic models, Spatial models and Spatial-temporal models

Although the likelihood function does not have to be Gaussian,  
each latent parameter must be a Gaussian given its hyperparameter in LGM.

the assumption must be held: for example, if we have two parameters

then, we have the latent effect follows Gaussian:

extended for additive models

relax the assumption of linear relationship

introduce random effects

instead of using , we apply the precision matrix

## 3. Gaussian Markov Random Fields (GMRFs)

the latent field should not only be Gaussian but also Gaussian Markov Random Field

We say is a GMRF if it has a multivariate normal density  
with additional conditional independence (also called the “Markov property”).

One common thing between different GMRFs:  
they all have a sparse precision matrix.

Sparse matrix provides a huge computational benefit when making Bayesian inference.

“Magic” in INLA: The joint distribution of of GMRF is also a GMRF

Precision matrix consists of sums of the precision matrices of the covariates and model components.

Example of band matrix example AR(1)

conditional independent for

conditional independent for

In summary all AR processes are a Gaussian process with a conditional independence property, making the corresponding precision matrix sparse.

## 4. Laplace Approximation and INLA

Laplace approximate integral let be the mode in function f(x), Taylor expansion on f(x) at :

is the Gaussian integral and since is the mode

Our goal is to accurately approximate the posterior marginals and . “The strategy used in INLA is to reformulate the problem as a series of sub-problems and only apply the Laplace approximation to the densities that are almost Gaussian.”

For Bayesian model, take a prior on each function which depends on hyperparameter

## 5. Random Walk and Smoothing functions

is the samll and equal space for ;  
 derivative of is continuous;  
 is the order backward difference operator:

# Code

set.seed(555)  
data <- mgcv::gamSim(1, n = 500, dist = "normal", scale = 2)

## Gu & Wahba 4 term additive model

formula1 <- y ~ f(x0, model = "rw1", scale.model = TRUE) +   
 f(x1, model = "rw1", scale.model = TRUE) +   
 f(x2, model = "rw1", scale.model = TRUE) +   
 f(x3, model = "rw1", scale.model = TRUE)  
  
formula2 <- y ~ f(x0, model = "rw2", scale.model = TRUE) +   
 f(x1, model = "rw2", scale.model = TRUE) +   
 f(x2, model = "rw2", scale.model = TRUE) +   
 f(x3, model = "rw2", scale.model = TRUE)  
  
n.group <- 50  
## ?inla.group  
## inla.group group or cluster covariates   
## so to reduce the number of unique values  
x0.new <- INLA::inla.group(data$x0, n = n.group, method = "quantile")  
x1.new <- INLA::inla.group(data$x1, n = n.group, method = "quantile")  
x2.new <- INLA::inla.group(data$x2, n = n.group, method = "quantile")  
x3.new <- INLA::inla.group(data$x3, n = n.group, method = "quantile")

names(inla.models()$latent)

## [1] "linear" "iid" "mec" "meb" "rgeneric"   
## [6] "rw1" "rw2" "crw2" "seasonal" "besag"   
## [11] "besag2" "bym" "bym2" "besagproper" "besagproper2"  
## [16] "fgn" "fgn2" "ar1" "ar1c" "ar"   
## [21] "ou" "intslope" "generic" "generic0" "generic1"   
## [26] "generic2" "generic3" "spde" "spde2" "spde3"   
## [31] "iid1d" "iid2d" "iid3d" "iid4d" "iid5d"   
## [36] "2diid" "z" "rw2d" "rw2diid" "slm"   
## [41] "matern2d" "dmatern" "copy" "clinear" "sigm"   
## [46] "revsigm" "log1exp" "logdist"

data.inla <- data.frame(y = data$y, x0 = x0.new,   
 x1 = x1.new, x2 = x2.new, x3 = x3.new)  
# str(data.inla)  
  
result1 <- INLA::inla(formula1, data = data.inla)  
result2 <- INLA::inla(formula2, data = data.inla)  
result3 <- INLA::inla(y ~ 1 + ns(x0, df = 10),   
 data = data.inla,  
 # verbose = TRUE,  
 # control.inla = list(strategy = "laplace", npoints = 20,  
 control.predictor = list(compute = TRUE))  
## Error in inla.inlaprogram.has.crashed() :   
## The inla-program exited with an error.   
## Unless you interupted it yourself,   
## please rerun with verbose=TRUE and check carefully.

summary(result1)

##   
## Call:  
## "INLA::inla(formula = formula1, data = data.inla)"   
## Time used:  
## Pre = 1.54, Running = 5.21, Post = 0.237, Total = 6.99   
## Fixed effects:  
## mean sd 0.025quant 0.5quant 0.975quant mode kld  
## (Intercept) 7.823 0.088 7.649 7.823 7.996 7.823 0  
##   
## Random effects:  
## Name Model  
## x0 RW1 model  
## x1 RW1 model  
## x2 RW1 model  
## x3 RW1 model  
##   
## Model hyperparameters:  
## mean sd 0.025quant 0.5quant  
## Precision for the Gaussian observations 2.61e-01 1.80e-02 0.228 2.60e-01  
## Precision for x0 2.72e+00 1.72e+00 0.752 2.29e+00  
## Precision for x1 1.58e+00 7.08e-01 0.614 1.45e+00  
## Precision for x2 2.09e-01 5.90e-02 0.114 2.02e-01  
## Precision for x3 1.88e+04 2.06e+04 1363.516 1.26e+04  
## 0.975quant mode  
## Precision for the Gaussian observations 2.97e-01 0.259  
## Precision for x0 7.20e+00 1.648  
## Precision for x1 3.34e+00 1.208  
## Precision for x2 3.44e-01 0.189  
## Precision for x3 7.31e+04 3734.570  
##   
## Expected number of effective parameters(stdev): 46.38(4.47)  
## Number of equivalent replicates : 10.78   
##   
## Marginal log-Likelihood: -1348.99

op <- par(mfrow = c(1, 3))  
bri.band.plot(result1, name = "x0", type = "random", xlab = "", ylab = "")  
lines(sort(data$x0), (data$f0 - mean(data$f0))[order(data$x0)], col = "red", lty = 2)  
  
bri.band.plot(result2, name = "x0", type = "random", xlab = "", ylab = "")  
lines(sort(data$x0), (data$f0 - mean(data$f0))[order(data$x0)], col = "red", lty = 2)  
  
# bri.band.plot(result3, name = "x0", type = "random", xlab = "", ylab = "")  
# bri.band.ggplot(result3, name = "x0", type = "random", xlab = "", ylab = "")  
# lines(sort(data$x0), (data$f0 - mean(data$f0))[order(data$x0)], col = "red", lty = 2)

data\_path <- here::here("NHANES\_AC\_processed.rds")  
df <- readr::read\_rds(data\_path)  
## extract the PA data  
lX <- log(1 + as.matrix(df[, paste0("MIN", 1:1440)]))  
lX[is.na(lX)] <- 0  
N <- nrow(lX)  
  
## bin the data into 60 minute intervals  
tlen <- 60  
nt <- ceiling(1440 / tlen)  
inx\_cols <- split(1:1440, rep(1:nt, each = tlen)[1:1440])  
lX\_bin <- vapply(inx\_cols,  
 function(x) rowMeans(lX[, x], na.rm = TRUE),  
 numeric(N))  
  
## get subject average curves  
inx\_rows <- split(1:N,  
 factor(df$SEQN,  
 levels = unique(df$SEQN)))  
lX\_bin\_ind <- t(vapply(inx\_rows,  
 function(x) colMeans(lX\_bin[x, ],  
 na.rm = TRUE),  
 numeric(nt)))  
nid <- nrow(lX\_bin\_ind)  
  
# get a data frame for model fitting  
sind <- seq(0, 1, len = nt)  
udf <- df %>%  
 dplyr::select(SEQN, Age) %>%  
 group\_by(SEQN) %>%  
 slice(1) %>%  
 ungroup()  
  
df\_fit <- data.frame(lAC = as.vector(t(lX\_bin\_ind)),  
 sind = rep(sind, nid),  
 SEQN = rep(unique(df$SEQN), each = nt)) %>%  
 left\_join(udf, by = "SEQN") %>%  
 mutate(id = factor(SEQN)) %>%  
 filter(!is.na(Age))  
View(df\_fit)  
set.seed(555)  
nid\_samp <- 500  
id\_samp <- sample(unique(df\_fit$id),  
 size = nid\_samp,  
 replace = FALSE)  
df\_fit\_sub <- subset(df\_fit, id %in% id\_samp)

fit\_naive <- bam(lAC ~ s(sind, bs = "cc", k = 20) +  
 s(sind, by = Age, bs = "cc", k = 20),  
 method = "fREML",  
 data = df\_fit\_sub,  
 discrete = TRUE)  
  
## extract the resiudals  
resid\_mat <- matrix(fit\_naive$residuals,  
 nid\_samp, nt,  
 byrow = TRUE)  
## fit fpca  
fpca\_fit <- refund::fpca.face(resid\_mat, knots = 15)  
## add in eigen-function  
for (k in 1:length(fpca\_fit$evalues)) {  
 df\_fit\_sub[[paste0("Phi", k)]] <- rep(fpca\_fit$efunctions[, k], nid\_samp)  
}

fit\_fri <- bam(lAC ~ s(sind, bs = "cc", k = 20) +  
 s(sind, by = Age, bs = "cc", k = 20) +  
 s(id, by = Phi1, bs = "re") +  
 s(id, by = Phi2, bs = "re") +  
 s(id, by = Phi3, bs = "re") +  
 s(id, by = Phi4, bs = "re"),  
 method = "fREML",  
 data = df\_fit\_sub,  
 discrete = TRUE)  
  
save(fit\_fri, file = "bios7720\_nhanes\_fri.Rdata")

formula\_nh1 <- lAC ~ -1 + f(Age, model = "rw1", scale.model = TRUE, constr = FALSE)   
fit\_inla <- INLA::inla(formula\_nh1, data = df\_fit\_sub)  
  
## Error in inla.check.location(location[[r]], term = gp$random.spec[[r]]$term, :   
## Locations are too close for f(Phi1, model="rw2", ...):   
## min(diff(sort(x)))/diff(range(x)) = 1.442e-04 < 1e-03   
## You can fix this by some kind of binning,   
## see ?inla.group If you want/need to bypass this check at your own risk, do   
## > m = get("inla.models", inla.get.inlaEnv())   
## > m$latent$rw2$min.diff = NULL   
## > assign("inla.models", m, inla.get.inlaEnv())

df\_fit\_sub$Phi1 <- INLA::inla.group(df\_fit\_sub$Phi1, n = n.group, method = "quantile")  
df\_fit\_sub$Phi2 <- INLA::inla.group(df\_fit\_sub$Phi2, n = n.group, method = "quantile")  
df\_fit\_sub$Phi3 <- INLA::inla.group(df\_fit\_sub$Phi3, n = n.group, method = "quantile")  
df\_fit\_sub$Phi4 <- INLA::inla.group(df\_fit\_sub$Phi4, n = n.group, method = "quantile")  
  
formula\_nh2 <- lAC ~ -1 + f(Age, model = "rw1", scale.model = TRUE, constr = FALSE) +  
 f(Phi1, model = "rw1", scale.model = TRUE) +   
 f(Phi2, model = "rw1", scale.model = TRUE) +   
 f(Phi3, model = "rw1", scale.model = TRUE) +   
 f(Phi4, model = "rw1", scale.model = TRUE)  
fit\_inla\_fri <- INLA::inla(formula\_nh2, data = df\_fit\_sub)  
  
# save(fit\_inla, file = "bios7720\_inla\_nhanes\_naive.Rdata")  
# save(fit\_inla\_fri, file = "bios7720\_inla\_nhanes\_fri.Rdata")  
# load("bios7720\_inla\_nhanes\_naive.Rdata")  
# load("bios7720\_inla\_nhanes\_fri.Rdata")

data\_mort <- here::here("data\_mort.rds") %>%  
 read\_rds()  
  
data <- here::here("NHANES\_AC\_processed.rds") %>%  
 read\_rds() %>%  
 ## subset the data  
 ## only consider good days of data  
 ## and individuals age 50 or over  
 filter(good\_day %in% c(1),  
 Age > 50,  
 n\_good\_days >= 3) %>%  
 ## get mortality data from the rnhanesdata package  
 ## merge and derive 5-year mortality indicator  
 left\_join(data\_mort, by = "SEQN") %>%  
 mutate(mort\_5yr = as.numeric(permth\_exm / 12 <= 5 &  
 mortstat %in% 1),  
 ## replace accidental deaths within 5 years as NA  
 mort\_5yr = ifelse(mort\_5yr == 1 & ucod\_leading %in% "004",  
 NA,  
 mort\_5yr)) %>%  
 ## drop anyone missing mortality data  
 ## or who had accidental deaths within 5 years  
 filter(!is.na(mort\_5yr))  
  
## extract just the activity count data  
Z <- as.matrix(data[, paste0("MIN", 1:1440)])  
## replace the (very few) missing values with 0  
Z[is.na(Z)] <- 0  
## get the binarized data  
Zb <- (Z >= 100) \* 1

## fit fpca on the log count data and binarized data----------------------------  
fit\_fpca <- fpca.face(log(1 + Z))  
fit\_fpca\_binary <- fpca.face(Zb)  
  
  
## extract the smoothed log count data  
## and estimate Pr(Active) data from the binary fit  
Zhat <- fit\_fpca$Yhat  
  
  
## truncate the estimate Pr(Active) values below at 0 and above at 1  
Zbhat <- fit\_fpca$Yhat  
Zbhat <- apply(Zbhat, 2,  
 function(x) ifelse(x < 0, 0, ifelse(x > 1, 1, x)))  
  
  
# Wed May 19 21:59:56 2021 -----------------------------------------------------  
## average across days within participants (SEQN)  
# unique subject identifiers  
uid <- unique(data$SEQN)  
# number of participants  
nid <- length(uid)  
# empty container to store average profiles  
Zsm <- matrix(NA, nid, 1440)  
Bsm <- matrix(NA, nid, 1440)

## loop over participants  
## get average "probability profiles"  
inx\_ls <- lapply(uid, function(x) which(data$SEQN %in% x))  
for (i in seq\_along(uid)) {  
 Zsm[i, ] <- colMeans(Zhat[inx\_ls[[i]], , drop = FALSE])  
 Bsm[i, ] <- colMeans(Zbhat[inx\_ls[[i]], , drop = FALSE])  
}  
  
  
## Get a data frame for analysis  
## which contains one row per participant  
df <- data[!duplicated(data$SEQN), ] %>%  
 dplyr::select(-one\_of(paste0("MIN", 1:1440)))  
## add in the activity count matrix  
## using the AsIs class via I()  
## note!! be careful when working  
## with dataframes which contain matrixes  
  
df$Zsm <- I(Zsm)  
df$Bsm <- I(Bsm)  
  
## fit SoFR using using average log(1+AC) profile versus  
## probability profile  
## set up the functional domain matrix  
## mgcv will use this to construct the basis \phi\_k^\gamma(s)  
sind <- seq(0, 1, len = 1440)  
smat <- matrix(sind, nrow(df), 1440, byrow = TRUE)  
df$smat <- I(smat)  
  
## set up the matrix of integration weights  
df$lmat <- I(matrix(1 / 1440, nrow(df), 1440))  
  
## multiply integration weights by the functional predictor  
df$zlmat <- I(df$lmat \* df$Zsm)  
df$blmat <- I(df$lmat \* df$Bsm)  
  
Age <- t(matrix(rep(df$Age, each = 1440), 1440, byrow = FALSE))  
Gender <- t(matrix(rep(df$Gender, each = 1440), 1440, byrow = FALSE))  
df$Age\_m <- I(Age)  
df$Gender\_m <- I(Gender)

## Age and Gender as linear coefficient.  
fglm\_ps0 <- gam(mort\_5yr ~   
 # s(smat, bs = "cc", k = 30) +  
 Age + Gender +  
 s(smat, by = zlmat, bs = "cc", k = 30),  
 data = df,  
 method = "REML",  
 family = binomial)  
  
fglm\_ps\_b0 <- gam(mort\_5yr ~   
 # s(smat, bs = "cc", k = 30) +  
 Age + Gender +  
 s(smat, by = blmat, bs = "cc", k = 30),  
 data = df,  
 method = "REML",  
 family = binomial)

summary(fglm\_ps0)

##   
## Family: binomial   
## Link function: logit   
##   
## Formula:  
## mort\_5yr ~ Age + Gender + s(smat, by = zlmat, bs = "cc", k = 30)  
##   
## Parametric coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.921816 0.623278 -7.897 2.86e-15 \*\*\*  
## Age 0.074815 0.007268 10.293 < 2e-16 \*\*\*  
## GenderFemale -0.570731 0.126214 -4.522 6.13e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df Chi.sq p-value   
## s(smat):zlmat 3.003 3.443 86.35 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.131 Deviance explained = 15.6%  
## -REML = 949.98 Scale est. = 1 n = 3243

summary(fglm\_ps\_b0)

##   
## Family: binomial   
## Link function: logit   
##   
## Formula:  
## mort\_5yr ~ Age + Gender + s(smat, by = blmat, bs = "cc", k = 30)  
##   
## Parametric coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.006195 0.776688 -6.446 1.15e-10 \*\*\*  
## Age 0.092677 0.006908 13.416 < 2e-16 \*\*\*  
## GenderFemale -0.594830 0.124675 -4.771 1.83e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df Chi.sq p-value   
## s(smat):blmat 3.481 4.007 44.5 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.108 Deviance explained = 13.4%  
## -REML = 973.96 Scale est. = 1 n = 3243

par(mfrow = c(1, 2))  
## plot the fit from mgcv::gam  
plot(fglm\_ps0,  
 xlab = "Time of Day", xaxt = "n",  
 ylab = expression(hat(gamma)(s)),  
 main = "mgcv with logTAC-lmat")  
  
plot(fglm\_ps\_b0,  
 xlab = "Time of Day", xaxt = "n",  
 ylab = expression(hat(gamma)(s)),  
 main = "mgcv with binary-lmat")

## Age and Gender interaction  
fglm\_ps1 <- gam(mort\_5yr ~ -1 +  
 # s(smat, bs = "cc", k = 30) +  
 s(smat, by = Age\_m, bs = "ts", k = 30) +  
 s(smat, by = Gender\_m, bs = "ts", k = 30) +  
 s(smat, by = zlmat, bs = "cc", k = 30),  
 data = df,  
 method = "REML",  
 family = binomial)

## Warning in newton(lsp = lsp, X = G$X, y = G$y, Eb = G$Eb, UrS = G$UrS, L =  
## G$L, : Fitting terminated with step failure - check results carefully

fglm\_ps\_b1 <- gam(mort\_5yr ~ -1 +  
 # s(smat, bs = "cc", k = 30) +  
 s(smat, by = Age\_m, bs = "ts", k = 30) +  
 s(smat, by = Gender\_m, bs = "ts", k = 30) +  
 s(smat, by = blmat, bs = "cc", k = 30),  
 data = df,  
 method = "REML",  
 family = binomial)  
  
summary(fglm\_ps1)

##   
## Family: binomial   
## Link function: logit   
##   
## Formula:  
## mort\_5yr ~ -1 + s(smat, by = Age\_m, bs = "ts", k = 30) + s(smat,   
## by = Gender\_m, bs = "ts", k = 30) + s(smat, by = zlmat, bs = "cc",   
## k = 30)  
##   
## Approximate significance of smooth terms:  
## edf Ref.df Chi.sq p-value   
## s(smat):Age\_m 1.0000 11.000 119.69 <2e-16 \*\*\*  
## s(smat):Gender\_m 0.9613 27.000 66.32 <2e-16 \*\*\*  
## s(smat):zlmat 2.0136 2.179 101.08 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.121 Deviance explained = 14.6%  
## -REML = 964.03 Scale est. = 1 n = 3243

summary(fglm\_ps\_b1)

##   
## Family: binomial   
## Link function: logit   
##   
## Formula:  
## mort\_5yr ~ -1 + s(smat, by = Age\_m, bs = "ts", k = 30) + s(smat,   
## by = Gender\_m, bs = "ts", k = 30) + s(smat, by = blmat, bs = "cc",   
## k = 30)  
##   
## Approximate significance of smooth terms:  
## edf Ref.df Chi.sq p-value   
## s(smat):Age\_m 1.0000 11.000 201.80 <2e-16 \*\*\*  
## s(smat):Gender\_m 0.9541 24.000 42.30 <2e-16 \*\*\*  
## s(smat):blmat 3.5401 4.084 54.56 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.0961 Deviance explained = 12.4%  
## -REML = 991.31 Scale est. = 1 n = 3243

## RW2D model  
## I have no idea what is this data structure.  
## It does not work, I give up.  
# Cstack\_info()  
# fit\_rw2d <- inla(mort\_5yr ~ -1 +  
# Age + Gender +  
# f(smat, model = "rw2", constr = F),  
# family = "binomial",  
# data = df,  
# control.predictor = list(compute = TRUE),  
# control.compute = list(dic = TRUE) )  
## Error: C stack usage 36254183 is too close to the limit

## SPDE which does not work! need more understanding on inla.  
## the structure is not working.   
## I do not know what is the counterpart of this model.  
# tps <- cbind(df$smat, df$Age)  
# mesh <- inla.mesh.2d(tps, cutoff = 0.05, max.edge = c(.5,1))  
#   
# tps <- bri.tps.prior(mesh)  
# node <- mesh$idx$loc  
# formula <- mort\_5yr ~ -1 + f(smat, model = "rw1") + f(zlmat, model = "rw1")  
# result <- inla(formula, family = "binomial", data = df, verbose=TRUE)

library(INLA)  
library(brinla)  
library(tidyverse)  
library(mgcv)  
  
set.seed(555)  
n <- 100  
x <- seq(0, 1, , n)  
f.true <- (sin(2 \* pi \* x^3))^3  
y <- f.true + rnorm(n, sd = 0.2)

data.inla <- data.frame(y = y, x = x)  
formula1 <- y ~ -1 + f(x, model = "rw1", constr = FALSE)  
system.time(result1 <- inla(formula1, data = data.inla))

## user system elapsed   
## 0.43 0.61 2.67

formula2 <- y ~ -1 + f(x, model = "rw2", constr = FALSE)  
system.time(result2 <- inla(formula2, data = data.inla))

## user system elapsed   
## 0.39 0.66 2.44

# names(inla.models()$latent)

head(result1$summary.random$x) %>% round(4)

## ID mean sd 0.025quant 0.5quant 0.975quant mode kld  
## 1 0.0000 0.0139 0.1329 -0.2471 0.0138 0.2754 0.0135 0  
## 2 0.0101 0.0605 0.1151 -0.1661 0.0605 0.2867 0.0606 0  
## 3 0.0202 0.0827 0.1109 -0.1353 0.0826 0.3007 0.0825 0  
## 4 0.0303 0.1124 0.1117 -0.1062 0.1120 0.3329 0.1112 0  
## 5 0.0404 -0.0182 0.1127 -0.2419 -0.0175 0.2016 -0.0160 0  
## 6 0.0505 0.0553 0.1090 -0.1590 0.0552 0.2697 0.0551 0

head(result2$summary.random$x) %>% round(4)

## ID mean sd 0.025quant 0.5quant 0.975quant mode kld  
## 1 0.0000 0.0504 0.1373 -0.2202 0.0506 0.3199 0.0509 0  
## 2 0.0101 0.0583 0.1073 -0.1526 0.0582 0.2693 0.0581 0  
## 3 0.0202 0.0635 0.0904 -0.1145 0.0635 0.2412 0.0635 0  
## 4 0.0303 0.0645 0.0832 -0.0991 0.0645 0.2281 0.0645 0  
## 5 0.0404 0.0600 0.0809 -0.0988 0.0599 0.2193 0.0597 0  
## 6 0.0505 0.0563 0.0805 -0.1015 0.0560 0.2152 0.0556 0

#### smooth\_spline() ----------  
fit.ss <- smooth.spline(x, y)  
res <- (fit.ss$yin - fit.ss$y) / (1 - fit.ss$lev)  
fhat3 <- fit.ss$y   
## lower bound  
f.lb3 <- fhat3 - 2 \* sd(res) \* sqrt(fit.ss$lev)   
## upper bound  
f.ub3 <- fhat3 + 2 \* sd(res) \* sqrt(fit.ss$lev)

#### gam() Sun Apr 04 09:19:51 2021 ---------  
system.time(fit.gam <- gam(y ~ s(x)))

## user system elapsed   
## 0.09 0.00 0.13

res.gam <- predict(fit.gam, se.fit = TRUE)  
## fitted curve  
fhat4 <- res.gam$fit   
## lower bound  
f.lb4 <- res.gam$fit - 2 \* res.gam$se.fit   
## upper bound  
f.ub4 <- res.gam$fit + 2 \* res.gam$se.fit

gridExtra::grid.arrange(plot1, plot2, plot3, plot4, nrow = 2)

a1 <- 3  
b1 <- 5e-5  
a2 <- -0.5  
b2 <- 5e-5  
  
lgprior1 <- list(prec = list(param = c(a1, b1)))  
lgprior2 <- list(prec = list(param = c(a2, b2)))  
  
formula5 <- y ~ -1 + f(x, model = "rw2",   
 constr = FALSE,   
 hyper = lgprior2)  
  
system.time(  
result5 <- inla(formula5,   
 data = data.inla,   
 ## used to specify the prior on delta  
 ## and hyper in f() in f() on tao  
 control.family = list(hyper = lgprior1))  
)

## user system elapsed   
## 0.33 1.00 3.16

## posterior mean  
fhat <- result5$summary.random$x$mean   
## 2.5% percentile  
f.lb <- result5$summary.random$x$"0.025quant"  
## 97.5% percentile  
f.ub <- result5$summary.random$x$"0.975quant"   
  
  
data.plot5 <- data.frame(y = y, x = x,   
 f.true = f.true,   
 fhat = fhat,   
 f.lb = f.lb,   
 f.ub = f.ub)  
  
plot5 <- data.plot5 %>%  
 ggplot(aes(x = x, y = y)) +  
 geom\_line(aes(y = fhat), size = 2, color = "darkgreen") +  
 geom\_line(aes(y = f.true), size = 2, color = "indianred") +  
 geom\_ribbon(aes(ymin = f.lb, ymax = f.ub), alpha = 0.2) +  
 geom\_point(aes(y = y)) +  
 theme\_bw(base\_size = 20) +  
 ggtitle("prior")

gridExtra::grid.arrange(plot2, plot4, plot5, nrow = 1)