Web Materials for:

Modeling the salivary cortisol profile in population research—The Multi Ethnic Study of Atherosclerosis

By Sánchez BN, Wu M, Raghunathan TE, Diez-Roux AV

Web Appendix 1: Code for parametric non linear models

For all estimation approaches, we assume the data is stored such that one line represents one of \( n_i \) observations for the \( i \)th participant, and a participant id number is a variable in the dataset (idno). The outcome variable is log transformed cortisol (lncort), time since wake up is measured in hours (corthours) or as a portion of the 24 hour period (cortday=corthours/24). Here we use age and gender as examples of covariates, namely age65 is 1 if age≥65 and 0 otherwise, and gender is 1 if the participant is male and 0 otherwise. Other predictors, including continuous predictors would be entered into the models in the same fashion.

Parametric non linear models can be estimated using maximum likelihood in software such as Proc NLMIXED in SAS, or approximate maximum likelihood using the nlmix() function in R. This approach would be particularly useful for modeling the peak of the curve, but data would need to be frequently sampled in the morning to ensure accurate inferences.

The SAS code is:

```sas
/*the following specifies the model;*/
pred = t0_id + t1_id*corthours
   + (exp(t2_id))*corthours*exp(-(exp(t3_id))*corthours);
   *note exponential transformation applied to t2_id and t3_id to ensure that peak is;
   * positive and that it occurs after wake up;

model lncort ~ normal (pred,s2);
random t0 t1 t2-normal( [0,0,0],[vart0,
   0, vart1,
   0, 0, vart2] ) subject=idno;
Run;
```
Estimate statements can be added before the `run;' statement to obtain estimates of parameters or combinations of parameters (e.g., variances and model-based inferences regarding group differences).

```sas
ESTIMATE "sigmasq" exp(2*logS2); * these statements produce variance estimates;
ESTIMATE "vart0"   exp(2*logsigt0) ; * in the regular (vs. log) scale         ;
ESTIMATE "vart1"   exp(2*logsigt1);
ESTIMATE "vart2"   exp(2*logsigt2);

ESTIMATE "AUC(16)" 16*total_t0 + 128*total_t1 + exp(total_t2)/ exp(2*total_t3) ;
   * this gives the estimated AUC at 16 hours post wake up. ;
   * to obtain estimated AUC at X hours post wake up, repeat the statement would be;
   * ESTIMATE "AUC(X)"  X*total_t0 + (X^2)/2*total_t1
                            + exp(total_t2)/ exp(2*total_t3)   ;

ESTIMATE "Change AUC(16) associated with Age" 16*BAT0 + 128*BAT1 +
   exp(total_t2 + BAT2)/exp(2*total_t3 + 2*BAT3) -
   exp(total_t2)/exp(2*total_t3) ;

ESTIMATE "Predicted value at t=1/2, f(1/2), for reference group"
   total_t0 + (total_t1)*0.5 + (exp(total_t2))*0.5*exp(-(exp(total_t3))*0.5);
   * to obtain prediction at any other point in the day,
   * replace the 0.5 by the desired time;

ESTIMATE "Predicted difference at t=0.25 associated with age"
   total_t0 + BAT0+ (total_t1+BAT1)* 0.25 +
   (exp(total_t2+BAT2))*0.25*exp(-(exp(total_t3+BAT3)) * 0.25 )
   - (total_t0 + total_t1*0.25 + exp(total_t2)*0.25*exp(-(exp(total_t3)*0.25 ));
   * to obtain the difference at any other point in the day,
   * simply replace the 0.25 by the desired time;

ESTIMATE "f(1/2)-f(0) associated with age"
   total_t0 + BAT0+ (total_t1+BAT1)* 0.5 +
   (exp(total_t2+BAT2))*0.5*exp(-(exp(total_t3+BAT3)) *0.5)
   -(total_t0 + total_t1*0.5 + exp(total_t2)*0.5*exp(-(exp(total_t3) *0.5))
   -(total_t0 + BAT0+ (total_t1+BAT1)*0 +
   (exp(total_t2+BAT2))*0*exp(exp(total_t3+BAT3)) *0)
   -(total_t0 + total_t1*0+ exp(total_t2)*0*exp(-exp(total_t3) *0));

ESTIMATE "Timing of peak for reference" (exp(1)*total_t1 +
   exp(total_t2))/(exp(total_t2) * exp(total_t3));

ESTIMATE "Absolute difference in timing of Peak associated with age"
   ((exp(1)*total_t1 +BAT1)+
   exp(total_t2+BAT2))/(exp(total_t2+BAT2) * exp(total_t3+BAT3)) -
   ((exp(1)*total_t1 + exp(total_t2))/(exp(total_t2) * exp(total_t3)));
```

To include a second level of nesting, see end of this document, section: NESTED_NLMIXED_MACRO.sas
Web Appendix 2: Functional mixed models with piece-wise linear splines

If only a few of observations per participant are collected, and all participants collect the samples at approximately the same times, a piece-wise linear model may be the only suitable approach.

**Basic code for piece-wise linear model**

```plaintext
data cortisoldata; *create piece-wise linear splines
set cortisol data;
piece1=corthours;
piece2=max(0, corthours-.5);
piece3=max(0, corthours-2);
run;

proc mixed data=cortisoldata noclprint empirical;
title "piecewise regression splines";
class idno day ;
model ln_cort = piece1 piece2 piece3
    age65 gender
    age65*piece1 age65*piece2 age65*piece3
    gender*piece1 gender*piece2 gender*piece3 /solution ;
random intercept piece1 piece2 piece3/subject=idno type=un;
run;
```

Additional estimate statements can be added before the `run;` statement to obtain estimates of such as those in Figure 4 and Table 2 of the paper:

```plaintext
Estimate 'reference curve average at 0.0625 hours' intercept 1 piece1 0.0625 /cl ;
Estimate 'reference curve average at 2.125 hours'
    intercept 1 piece1 2.125 piece2 1.625 piece3 0.125 /cl ;
Estimate "age association at 0.0625 hours" age65 1 piece1*age65 0.0625 /cl;
Estimate "age effect 2.125 hours"
    age65 1 piece1*age65 2.125 piece2*age65 1.625 piece3*age65 0.125 /cl ;
Estimate "difference in AUC(16) associated with age"
    age65 16 piece1*age65 128 piece2*age65 120.125 piece3*age65 98 /cl ;
estimate "age association with intercept"  age65 1 /cl ;
estimate "age association with slope 1"  age65*piece1 1 /cl ;
estimate "age association with slope 2"  age65*piece1 1 age65*piece2 1/cl ;
estimate "age association with slope 3"  
    age65*piece1 1 age65*piece2 1 age65*piece3 1/cl ;
```

To expand model to nested days within person, add a second random statement:

```plaintext
random intercept piece1 piece2 piece3/subject=day(idno) type=un;
```
Penalized spline models with repeated measures can be estimated with the `gamm()` function in R, from the mgcv package. The code below shows how the example model in the paper was fitted.

R code:

```r
#Install libraries
library(foreign)
library(stats)
library(splines)
library(mgcv)
library(nlme)
library(MASS)

#Read data
cortisol.data<-read.csv("CortisolData.csv")

cortisol.data$corthours.day<- cortisol.data$corthours/24

cortisol.data$sqrt.time<-sqrt(cortisol.data$corthours.day)

#declare variables
y <- cortisol.data$lncort
time <- cortisol.data$sqrt.time
subject<- cortisol.data$idno
age65<-cortisol.data$age65
gender<- cortisol.data$gender

#this fit is to set up the basis functions g(t,tau)
initialfit <- gamm(y ~s(time, k=25 )
  , data=cortisol.data,method="REML")

#chooses the first 5 basis functions
Z.subjectnew <- initialfit$lme$data[,5][,1:5]
random.time<- initialfit$lme$data$X[,2]

#model with covariates
penalizedsplinemodel.final.covariates <- gamm(y ~s(time , k=25)
  +s(time ,by=age65, k=25)+s(time , by=gender, k=25) ,
  random=list(subject=pdSymm(~random.time),
  subject=pdIdent(~2.subjectnew - 1)),method="REML")

#print model summary and fit criteria
penalizedsplinemodel.final.covariates$gam
foo<-summary(penalizedsplinemodel.final.covariates$gam)
print( c(foo$AIC,foo$BIC,foo$logLik) )
```

After fitting the model, all other inferences can be derived by manipulating the output from the model. The code below shows how to obtain the graphs in Figure 4 of the manuscript.

```r
#obtain predictions from 0 to 18 hours
newcorthours <- seq(0,18,length=2000) ## predictions will be evaluated at this times

## transform time to match the scale on which the model was fitted
x.mesh1 <- sqrt(newcorthours/24)

#create a ‘fake’ dataset that will be used for predictions
predict.data<- data.frame(cbind(gender=1, age65=1, time=x.mesh1))

#use predict function to obtain predictions and standard errors
```
X0 <- predict(penalizedsplinemodel.final.covariates$gam, predict.data,type="lterms",se.fit=TRUE)

#obtain curve for reference group and confidence interval beta_0(t)
mean.fitted<- summary(penalizedsplinemodel.final.covariates$lme)$coefficients$fixed[1] + X0$fit[,1]
mean.lower<- mean.fitted - 1.96 * X0$se.fit[,1]
mean.upper<- mean.fitted + 1.96 * X0$se.fit[,1]

#plot average curve
matplot(newcorthours,cbind(mean.fitted,mean.lower,mean.upper),lty=c(1,2,2), xlab="Time Since Wake Up (Hours)",ylab="Ln(Cortisol)", xaxt = "n",main=" ")

#functional coefficient for age association, beta_1(t)
age.fitted<- X0$fit[,2]
age.lower<- age.fitted - 1.96 * X0$se.fit[,2]
age.upper<- age.fitted + 1.96 * X0$se.fit[,2]

#plot association with age
matplot(newcorthours,cbind(age.fitted,age.lower,age.upper),lty=c(1,2,2), xlab="Time Since Wake Up (Hours)",ylab="Ln(Cortisol)", xaxt = "n",main=" ")

#numerical derivatives of functional coefficient beta_1'(t)
x.mesh1.real <- seq(.01,18,by=0.01) ## where to evaluate
x.mesh1 <- sqrt(x.mesh1.real/24)
predict.data<- data.frame(cbind(gender=1, age65=1, time=x.mesh1))
X0 <- predict(penalizedsplinemodel.final.covariates$gam,predict.data,type="lpmatrix")
eps <- 1e-7 ## finite difference interval
x.mesh2.real <- seq(.01,18,by=0.01) + .01 ## shift the evaluation mesh
x.mesh2 <- sqrt(x.mesh2.real/24)
newd <- data.frame(cbind(gender=1, age65=1,  time = x.mesh2))
X1 <- predict(penalizedsplinemodel.final.covariates$gam,newd,type="lpmatrix")
Xp <- (X1-X0)/.01 ## maps coefficients to (finite difference approx.) derivatives
colnames(Xp) ## can check which cols relate to which smooth

i<-2 #i=2 for age, i=3 for sex
X <- Xp*0
X[, (i-1)*25+1:25] <- Xp[, (i-1)*25+1:25] ## Xi%*%coef(b) = smooth deriv i

# ith smooth derivative
age.df <- Xi%*%coef(penalizedsplinemodel.final.covariates$gam)
age.df.sd <- rowSums(Xi%*% penalizedsplinemodel.final.covariates$gam$Vp*Xl)^.5
age.df.lower<- age.df - 1.96 * age.df.sd
age.df.upper<- age.df + 1.96 * age.df.sd

## prediction for each individual, b_i(t)
.o.time<order(random.time)
RandomPredictors.groupedtime <- groupedData( intercept~1 | random.time, data=data.frame(cbind( time=time[o.time], intercept=1, random.time=random.time[o.time], Z=Z.subjectnew[o.time,] )))
RandomPredictors <- gsummary(RandomPredictors.groupedtime)
unique.times<- RandomPredictors$time
real.times<- unique.times*unique.times*24
RandomPredictors <- cbind( as.numeric(RandomPredictors[,2]),
unique(random.time[o.time]),
as.numeric(RandomPredictors[,4]),
as.numeric(RandomPredictors[,5]),
Inferences regarding features can also be obtained. The code below shows sample calculations for Table 2.

```r
## Obtaining f(1/2) - f(0) - CAR related to age
x.mesh1.real <- c(0,0.5)
x.mesh1 <- sqrt(x.mesh1.real/24)
predict.data<- data.frame(cbind(gender=1, age65=1, time=x.mesh1))
X0 <- predict(penalizedsplinemodel.final.covariates$gam,predict.data,type="lpmatrix" )
a<-c(-1,1) # contrast (we will subtract f(0) from f(1/2)

diff.CAR <- Xs %*% coef(penalizedsplinemodel.final.covariates$gam)[26:(26+24)]
var.diff <- Xs %*% penalizedsplinemodel.final.covariates$gam$Vp[26:(26+24),26:(26+24)] %*% t(Xs)
diff.CAR
diff.CAR-sqrt( var.diff)*1.96
diff.CAR+sqrt( var.diff)*1.96

## (f(2) - f(1/2))/1.5
x.mesh1.real <- c(0.5,2)
x.mesh1 <- sqrt(x.mesh1.real/24)
predict.data<- data.frame(cbind(gender=1, age65=1, time=x.mesh1))
X0 <- predict(penalizedsplinemodel.final.covariates$gam,predict.data,type="lpmatrix" )
a<-c(-1,1)/1.5 # we divide by 1.5 because we want the per hour difference in slope

diff.2nd <- Xs %*% coef(penalizedsplinemodel.final.covariates$gam)[26:(26+24)]
var.diff <- Xs %*% penalizedsplinemodel.final.covariates$gam$Vp[26:(26+24),26:(26+24)] %*% t(Xs)
diff.2nd
diff.2nd-sqrt( var.diff)*1.96
diff.2nd+sqrt( var.diff)*1.96

## 24*(f(16) - f(4))/12
x.mesh1.real <- c(4,16)
```
x.mesh1 <- sqrt(x.mesh1.real/24)

predict.data<- data.frame(cbind(gender=1, age65=1, time=x.mesh1))

X0 <- predict(penalizedsplinemodel.final.covariates$gam,predict.data,type="lpmatrix")

a<- 24*c(-1,1)/12

# age
Xs <- t(a) %*% X0[,26:(26+24)]

diff.3rd <- Xs %*% coef(penalizedsplinemodel.final.covariates$gam)[26:(26+24)]
var.diff <- Xs %*% penalizedsplinemodel.final.covariates$gam$Vp[26:(26+24),26:(26+24)] %*% t(Xs)

diff.3rd

diff.3rd-sqrt( var.diff)*1.96

diff.3rd+sqrt( var.diff)*1.96

## AUC(16)

# AUC(16)

x.mesh1.real <- seq(0,16,length=2000) ## where to evaluate

predict.data<- data.frame(cbind(gender=1, age65=1, time=x.mesh1))

X0 <- predict(penalizedsplinemodel.final.covariates$gam,predict.data,type="lpmatrix")

a<- rep(x.mesh1.real[2]-x.mesh1.real[1],2000)  # we will add up all entries

# age
Xs <- t(a) %*% X0[,26:(26+24)]

auc.16 <- Xs %*% coef(penalizedsplinemodel.final.covariates$gam)[26:(26+24)]
var.auc <- Xs %*% penalizedsplinemodel.final.covariates$gam$Vp[26:(26+24),26:(26+24)] %*% t(Xs)

auc.16

auc.16-sqrt(var.auc)*1.96

auc.16+sqrt(var.auc)*1.96

To expand model to days nested within person:

day<- cortisol.data$day

penalizedsplinemodel.final.covariates.nested.day  <- gamm(y ~s(time , k=25)+s(time , by=age65, k=25)+s(time , by=gender, k=25) ,
random=list(subject=pdSymm(~random.time),
subject=pdIdent(~Z.subjectnew  - 1),
day=pdSymm(~random.time)) ,method="REML")
Fitting a nested nonlinear model is challenging. The macros below first get initial values using the firo option of nlmixed (a ‘quick’ approximation to the solution), and subsequently obtain the more exact solution.

```sas
/*%random_nested(datain=, option_nlmixed= );
datain=dataset ;
/*assume covariates are age65 gender
/*option_nlmixed=option for proc Nlmixed;
***********************************************************************************************/

**********************************************************
*                     Main macro
**********************************************************;
%macro random_nested(datain=, option_nlmixed= );
%let cov_t0=BAT0*age65+BGT0*gender;
%let cov_t1=BAT1*age65+BGT1*gender;
%let cov_t2=BAT2*age65+BGT2*gender;
%let cov_t3=BAT3*age65+BGT3*gender;
*get initial value using method=firo for proc nlmixed;
%est_initial(data=&datain);
*final estimate from proc nlmixed;
%nlmixed(data=&datain);
%mend random_nested;

*      macros used in main macro
*************************************************************;
%macro est_initial(data=);
proc nlmixed data=&data method=firo;
*set up initial values is not necessary, but try to control order of coefficient;
  PARMS total_t0=2.4 total_t1=1 total_t2=1 t3=1
  bat0=1 bgt0=1 bat1=1 bgt1=1 bat2=1 bgt2=1
  bat3=1 bgt3=1 s2=0.24 logsigt0=0 logsigt1=0 logsigt2=0;
array t0 { 1  };
array t1 { 1  };
array t2 { 1  };
array bt0 { 3  };
array bt1 { 3  };
array bt2 { 3  };
t0_id =total_t0+t0{1} + bt0{day}; *id= ith person dth day;
t1_id =total_t1+t1{1}+ bt1{day};  *day is coded 1, 2, 3;
t2_id =total_t2+t2{1}+ bt2{day};  *day is coded 1, 2, 3;
t3_id =t3;
t0_id =t0_id + &cov_t0;
t1_id =t1_id + &cov_t1;
t2_id =t2_id + &cov_t2;
t3_id =t3_id + &cov_t3;
vard0  = exp(2*logsigbt0);
vard1  = exp(2*logsigbt1);
vard2  = exp(2*logsigbt2);
varbt0 = exp(2*logsigbt0);
varbt1 = exp(2*logsigbt1);
varbt2 = exp(2*logsigbt2);
pred  =t0_id + t1_id*corthours
  + (exp(t2_id))corthours*exp(-(exp(t3_id))corthours);
model ln_cort ~ normal (pred,s2);
random t0t1 bt01 bt02 bt03 t11 bt12 bt13 t21 bt21 bt22 bt23~
```
normal( [0,0,0,0,0,0,0,0,0,0,0,0],
    [vart0,  
    0, varbt0,  
    0, 0, varbt0,  
    0, 0, 0, varbt0,  
    0, 0, 0, 0, varbt1,  
    0, 0, 0, 0, 0, varbt1,  
    0, 0, 0, 0, 0, 0, varbt1,  
    0, 0, 0, 0, 0, 0, 0, varbt2,  
    0, 0, 0, 0, 0, 0, 0, 0, varbt2,  
    0, 0, 0, 0, 0, 0, 0, 0, 0, varbt2,  
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, varbt2],
    ) subject=idno;
ods output ParameterEstimates=int_est;
run;
%mend est_initial;

%macro nlmixed(data=);
proc nlmixed data=&data &option_nlmixed;
  parms / data=int_est;
  array t0 { 1  ];
  array t1 { 1  ];
  array t2 { 1  ];
    array bt0 { 3  ];
    array bt1 { 3  ];
    array bt2 { 3  ];

  t0_id =total_t0+t0{1} + bt0{day}; *id= ith person dth day;
  t1_id =total_t1+t1{1}+ bt1{day};  *day is coded 1, 2, 3;
  t2_id =total_t2+t2{1}+ bt2{day};  *day is coded 1, 2, 3;
  t3_id=t3;

  t0_id =t0_id + &cov_t0;
  t1_id =t1_id + &cov_t1;
  t2_id =t2_id + &cov_t2;
  t3_id =t3_id + &cov_t3;

  vart0   = exp(2*logsigt0);
  vart1   = exp(2*logsigt1);
  vart2   = exp(2*logsigt2);
  varbt0  = exp(2*logsigb_t0);
  varbt1  = exp(2*logsigb_t1);
  varbt2  = exp(2*logsigb_t2);

  pred =t0_id + t1_id*corthours
       + (exp(t2_id))*corthours*exp(-(exp(t3_id))*corthours);

  model ln_cort ~ normal (pred,s2);
  random t01 bt02 bt03 t11 bt12 bt13 t21 bt22 bt23~
    normal( [0,0,0,0,0,0,0,0,0],
        [vart0,  
        0, varbt0,  
        0, 0, varbt0,  
        0, 0, 0, varbt0,  
        0, 0, 0, 0, varbt1,  
        0, 0, 0, 0, 0, varbt1,  
        0, 0, 0, 0, 0, 0, varbt1,  
        0, 0, 0, 0, 0, 0, 0, varbt2,  
        0, 0, 0, 0, 0, 0, 0, 0, varbt2,  
        0, 0, 0, 0, 0, 0, 0, 0, 0, varbt2,  
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, varbt2,  
        ] ) subject=idno;
  ods output ParameterEstimates=estimate;
run;
%mend;