

R-code for ‘A time-heterogeneous D-vine copula model for unbalanced and unequally spaced longitudinal data’

Md Erfanul Hoque, Elif F. Acar and Mahmoud Torabi

The file contains the source code of the analysis of a subset of the Manitoba Follow-up Study data presented in Section 5 of *A time-heterogeneous D-vine copula model for unbalanced and unequally spaced longitudinal data* by Hoque, Acar and Torabi (2020).

Load required packages

```
library(here) # to specify the folder where something is located in
library(VineCopula) # required to work with D-vine copulas
library(nlme) # enables working with linear mixed models (LMMs)
library(condMVNorm) # required for predicting conditional quantiles of
                    # multivariate normal distributions
```

```
# Load data
load(here("DATA", "Data_MFUS.Rdata")) # 'here' function will specify
                                       # the "DATA" folder to load
                                       # the required data

names(MFUS)
```

```
## [1] "id"          "followup"    "px_age"      "bmi"         "IHD"         "DBP"
```

```
# Maximum number of measurements per individual
d <- max(table(MFUS$id))
d
```

```
## [1] 5
```

```
n <- length(unique(MFUS[,1]))
n
```

```
## [1] 462
```

Model fitting

We fit a linear mixed model (LMM), a time-homogeneous D-vine copula model (HOM-P) and a time-heterogeneous D-vine copula model (HET-P) to the data.

Linear mixed model (LMM)

```
# Fit linear mixed model with random intercept and exponential error
# correlation structure ; px_age, bmi, IHD as covariates.

LMM.Exp <- lme(DBP ~ px_age+ bmi+ IHD, # formula
               data = MFUS, # data
               random = ~ 1|id, # grouped random effect
               correlation = corExp(form = ~ px_age), # error structure
               method = "ML" # estimation method
)
summary(LMM.Exp)
```

```
## Linear mixed-effects model fit by maximum likelihood
## Data: MFUS
##      AIC      BIC    logLik
## 11736.46 11774.58 -5861.229
##
## Random effects:
## Formula: ~1 | id
##      (Intercept) Residual
## StdDev:    4.850646 6.386818
##
## Correlation Structure: Exponential spatial correlation
## Formula: ~px_age | id
## Parameter estimate(s):
##      range
## 0.01524487
## Fixed effects: DBP ~ px_age + bmi + IHD
##              Value Std.Error   DF   t-value p-value
## (Intercept) 51.75301 2.2942930 1249 22.557282 0.0000
## px_age      0.20148 0.0347559 1249  5.796886 0.0000
## bmi         0.73968 0.0959485 1249  7.709150 0.0000
## IHD         0.76895 0.5821587  460  1.320862 0.1872
## Correlation:
##      (Intr) px_age bmi
## px_age -0.218
## bmi    -0.839 -0.327
```

```
## IHD      0.019  0.030 -0.126
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -3.174427492 -0.622456357  0.001988641  0.587928529  4.771298638
##
## Number of Observations: 1713
## Number of Groups: 462
```

D-vine copula models

For the D-vine copula models HOM-P and HET-P, we use the univariate marginal distribution implied by the fitted linear mixed model. We first transform the observed measurements to copula data using the distribution functions of $\mathcal{N}(x_{i,j}^\top \eta, \phi^2)$, where $\phi^2 = \text{var}(\gamma_i) + \text{var}(\varepsilon_{i,j})$.

Obtain copula data to fit D-vine copula models

```
# Standard deviation of the random effect gamma
sd.random <- as.numeric(VarCorr(LMM.Exp)[1, 2])

# Standard deviation of the error term epsilon_{i,j}
sd.error <- as.numeric(VarCorr(LMM.Exp)[2, 2])

# Standard deviation of Y_{i,j}
phi <- sqrt(sd.error^2 + sd.random^2)

# Transformation of the observations to the copula scale
u.vec <- pnorm(LMM.Exp$residuals[,1] / phi)

# Matrix to store the information, where unobserved measurements are
# denoted by 'NA'.
U.data <- matrix(NA, n, 5) # copula data matrix
j <- 0 # auxiliary counter

# For individual i fill the first d_i entries of the ith row in the
# matrix U.data with the transformed entries corresponding to the
# measurements of individual i
for (i in 1:n){
  k <- min(5, table(MFUS$id)[i])
  U.data[i, 1:k] <- u.vec[j + 1:k]
  j <- j + k
}
head(U.data) # show the copula data structure
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.3210778 0.98543447 0.6392163 0.6042819 0.5266832
## [2,] 0.3712734 0.64330496 0.2160685          NA          NA
## [3,] 0.3653448 0.08711343          NA          NA          NA
## [4,] 0.4065085 0.29108635 0.5607451 0.6329385 0.4112875
## [5,] 0.6491174 0.32565800 0.5465238 0.1285328 0.1002344
## [6,] 0.7723560 0.89085685 0.7997230 0.8754386 0.9019637
```

Fit time-homogeneous D-vine copula model (HOM-P) to copula data

```
# Set of candidate families
family.set <- c(1,3,4,5,13,14)

# D-vine matrix with order 1-2-3-4-5
Mat<- matrix(c(5,0,0,0,0, 1,4,0,0,0, 2,1,3,0,0, 3,2,1,2,0,
               4,3,2,1,1),5,5, byrow=T)

# Fit HOM-P with fixed order 1:5
Dvine.HOM <- RVineCopSelect(data = U.data, # copula data
                             Matrix = Mat, # vine structure
                             familyset = family.set,
                             selectioncrit = "AIC" # selection criterion
)
```

```
## Warning: In RVineCopSelect: Some of the data are NA. Only pairwise complete
## observations are used.
```

```
# The above warning only confirms that the data are unbalanced.
Dvine.HOM
```

```
## D-vine copula with the following pair-copulas:
## Tree 1:
## 4,5 Gumbel (par = 1.45, tau = 0.31)
## 3,4 Gumbel (par = 1.29, tau = 0.22)
## 2,3 Gumbel (par = 1.25, tau = 0.2)
## 1,2 Gumbel (par = 1.24, tau = 0.19)
##
## Tree 2:
## 3,5;4 Survival Clayton (par = 0.34, tau = 0.15)
## 2,4;3 Gaussian (par = 0.2, tau = 0.13)
## 1,3;2 Survival Gumbel (par = 1.23, tau = 0.19)
##
## Tree 3:
```

```
## 2,5;3,4 Gaussian (par = 0.3, tau = 0.19)
## 1,4;2,3 Frank (par = 0.91, tau = 0.1)
##
## Tree 4:
## 1,5;2,3,4 Frank (par = 0.2, tau = 0.02)
##
## ---
## 1 <-> V1, 2 <-> V2, 3 <-> V3, 4 <-> V4,
## 5 <-> V5

# Copula log-likelihood value for HOM-P model
Cloglik_HOMByPair<- sum(Dvine.HOM$pair.logLik);Cloglik_HOMByPair

## [1] 147.8705

# Store the copula parameters and families from fitted HOM-P model
par<-Dvine.HOM$par
parset.HOM <-list(c(par[5,1],par[5,2],par[5,3],par[5,4]),
                 c(par[4,1],par[4,2],par[4,3]),
                 c(par[3,1],par[3,2]),
                 c(par[2,1]))
fam<-Dvine.HOM$family
familyset.HOM <- list(c(fam[5,1],fam[5,2],fam[5,3],fam[5,4]),
                     c(fam[4,1],fam[4,2],fam[4,3]),
                     c(fam[3,1],fam[3,2]),
                     c(fam[2,1]))
```

Fit time-heterogeneous D-vine copula model (HET-P) to copula data

To fit the HET-P model, we first calculate time interval between two measurements. The fitted model gives the estimated values of the model parameters for each pair copula.

```
# Creating the Time variable (px-age) for the analysis of HET-P
visit.times <- MFUS[,c(1,3,2)] # extract time variable with corresponding id
visit.times <- reshape(visit.times, v.names="px_age", timevar="followup",
                       idvar="id", direction="wide")
visit.times <- visit.times[,-1] # remove the first column
# Data frame of time variable, where unobserved measurements are
# denoted by 'NA'
Visits <- data.frame(visit.times) # Final time variable to use in analysis
head(Visits)

##      px_age.1 px_age.2 px_age.3 px_age.4 px_age.5
## 1  27.41410 34.03970 34.88296 35.38125 35.86858
## 6  28.49555 30.57906 34.87201      NA      NA
## 9  33.41821 39.43600      NA      NA      NA
```

```

## 11 36.97741 38.97878 41.33060 43.82204 45.85626
## 16 29.08419 32.23272 35.46064 38.45038 38.80630
## 21 25.83436 28.89254 32.01916 32.02190 33.87269

## BEGIN fitted model
# Calculate the time intervals for Tree 1
t<- sapply(1:(d-1), function(j) Visits[,j+1] - Visits[,j])
colnames(t)<-c("t.12", "t.23", "t.34", "t.45")
t.12<-t[,1];t.23<-t[,2];t.34<-t[,3];t.45<-t[,4]

# Respective ids of available time intervals in Tree 1
id1 <-sapply(1:(d-1), function(j) which(complete.cases(t[,j])))
id12<-id1[[1]];id23<-id1[[2]];id34<-id1[[3]];id45<-id1[[4]]

# Estimation of model parameters of pair copulas in Tree 1
fit<- sapply(1:(d-1), function(j) LBiCopFit(u1=U.data[,j], u2=U.data[,j+1],
      x=t[,j],family=family.set,repar="pearson",start = c(0.01,0.01)))

fit.beta.T1<- sapply(1:(d-1), function(j) c(fit[1,j],fit[2,j]))
ll.T1<-sum(unlist(fit[3,])) #loglikelihood value in Tree 1
fam.T1<-fit[4,] # selected family of pair copulas in Tree 1

tau.T1<- sapply(1:(d-1), function(j) LDVine.rho2tau(unlist(fit.beta.T1[,j]),
      t[,j][id1[[j]]]))
par.T1<- sapply(1:(d-1), function(j) BiCopTau2Par(fam.T1[[j]], tau.T1[[j]]))

# Obtain the pseudo-conditional marginals for Tree 2
vv<-list()
for (j in 1:(d-2)){
  vv[[j]] <- BiCopHfunc2(u1=U.data[,j][id1[[j]]],u2=U.data[,j+1][id1[[j]]],
      par.T1[[j]],family = fam.T1[[j]])
  vv[[d-2+j]] <- BiCopHfunc2(u1=U.data[,j+2][id1[[j+1]]],
      u2=U.data[,j+1][id1[[j+1]]],par.T1[[j+1]],family = fam.T1[[j+1]])
}

# Store pseudo-conditional marginals with respective ids
vv1.2 <-data.frame(id1[[1]],vv[[1]])
vv3.2 <-data.frame(id1[[2]],vv[[4]])
vv2.3 <-data.frame(id1[[2]],vv[[2]])
vv4.3 <-data.frame(id1[[3]],vv[[5]])
vv3.4 <-data.frame(id1[[3]],vv[[3]])
vv5.4 <-data.frame(id1[[4]],vv[[6]])

# Calculate the time intervals for Tree 2
t<- sapply(1:(d-2), function(j) Visits[,j+2] - Visits[,j])
t.13.2<-t[,1];t.24.3<-t[,2];t.35.4<-t[,3]

```

```

# Respective ids of available time intervals in Tree 2
id2 <-sapply(1:(d-2), function(j) which(complete.cases(t[,j])))
id13.2<-id2[[1]];id24.3<-id2[[2]];id35.4<-id2[[3]]

# List of available conditional data needed for estimation in Tree 2
condata1 <- list(vv1.2[id12%in%id13.2,2],vv3.2[id23%in%id13.2,2],
                vv2.3[id23%in%id24.3,2],vv4.3[id34%in%id24.3,2],
                vv3.4[id34%in%id35.4,2],vv5.4[id45%in%id35.4,2])

# Estimation of model parameters of pair copulas in Tree 2
fit<- sapply(1:(d-2), function(j) LBiCopFit(u1 = condata1[[2*j-1]],
      u2 = condata1[[2*j]], x=t[,j][id2[[j]]],family=family.set,
      repara="pearson",start = c(0.01,0.01)))

fit.beta.T2<- sapply(1:(d-2), function(j) c(fit[1,j],fit[2,j]))
ll.T2<-sum(unlist(fit[3,])) #loglikelihood value in Tree 2
fam.T2<-fit[4,] # selected family of pair copulas in Tree 2

tau.T2<- sapply(1:(d-2), function(j) LDVine.rho2tau(unlist(fit.beta.T2[,j]),
      t[,j][id2[[j]]]))
par.T2<- sapply(1:(d-2), function(j) BiCopTau2Par(fam.T2[[j]], tau.T2[[j]]))

# Obtain the pseudo-conditional marginals for Tree 3
vvv<-list()
for (j in 1:(d-3)){
  vvv[[j]] <- BiCopHfunc2(u1=condata1[[2*j-1]], u2=condata1[[2*j]],
      par.T2[[j]],family = fam.T2[[j]])
  vvv[((d-3)+j)] <- BiCopHfunc2(u1=condata1[[2*j+2]], u2=condata1[[2*j+1]],
      par.T2[[j+1]],family = fam.T2[[j+1]])
}

# Store pseudo-conditional marginals with respective ids
vvv1.23 <-data.frame(id2[[1]],vvv[[1]])
vvv4.23 <-data.frame(id2[[2]],vvv[[3]])
vvv2.34 <-data.frame(id2[[2]],vvv[[2]])
vvv5.34 <-data.frame(id2[[3]],vvv[[4]])

# Calculate the time intervals for Tree 3
t<- sapply(1:(d-3), function(j) Visits[,j+3] - Visits[,j])
t.14.23<-t[,1];t.25.34<-t[,2]

# Respective ids of available time intervals in Tree 3
id3 <-sapply(1:(d-3), function(j) which(complete.cases(t[,j])))
id14.23<-id3[[1]];id25.34<-id3[[2]]

```

```

# List of available conditional data needed for estimation in Tree 3
condata2 <- list(vvv1.23[id13.2%in%id14.23,2],vvv4.23[id24.3%in%id14.23,2],
               vvv2.34[id24.3%in%id25.34,2],vvv5.34[id35.4%in%id25.34,2])

# Estimation of model parameters of pair copulas in Tree 3
fit<- sapply(1:(d-3), function(j) LBiCopFit(u1 = condata2[[2*j-1]],
      u2 = condata2[[2*j]], x=t[,j][id3[[j]]], family=family.set,
      repar="pearson",start = c(0.01,0.01)))

fit.beta.T3<- sapply(1:(d-3), function(j) c(fit[1,j],fit[2,j]))
ll.T3<-sum(unlist(fit[3,])) #loglikelihood value in Tree 3
fam.T3<-fit[4,] # selected family of pair copulas in Tree 3

tau.T3<- sapply(1:(d-3), function(j) LDVine.rho2tau(unlist(fit.beta.T3[j,]),
      t[,j][id3[[j]]]))
par.T3<- sapply(1:(d-3), function(j) BiCopTau2Par(fam.T3[[j]], tau.T3[[j]]))

# Obtain the pseudo-conditional marginals for Tree 4
vvvv<-list()
for (j in 1:(d-4)){
  vvvv[[j]] <- BiCopHfunc2(u1=condata2[[2*j-1]], u2=condata2[[2*j]],
      par.T3[[j]],family = fam.T3[[j]])
  vvvv[[d-4+j]] <- BiCopHfunc2(u1=condata2[[2*j+2]], u2=condata2[[2*j+1]],
      par.T3[[j+1]],family = fam.T3[[j+1]])
}

# Store pseudo-conditional marginals with respective ids
vvvv1.234 <-data.frame(id3[[1]],vvvv[[1]])
vvvv5.234 <-data.frame(id3[[2]],vvvv[[2]])

# Calculate the time intervals for Tree 4
t<- sapply(1:(d-4), function(j) Visits[,j+4] - Visits[,j])
t.15.234<-t[,1]

# Respective ids of available time intervals in Tree 4
id4 <-sapply(1:(d-4), function(j) which(complete.cases(t[,j])))
id15.234<-id4[,1]

# List of available conditional data needed for estimation in Tree 4
condata3 <- list(vvvv1.234[id14.23%in%id15.234,2],
               vvvv5.234[id25.34%in%id15.234,2])

# Estimation of copula parameters of pair copulas in Tree 4
fit<- sapply(1:(d-4), function(j) LBiCopFit(u1 = condata3[[2*j-1]],
      u2 = condata3[[2*j]], x=t[,j][id4[,j]], family=family.set,

```



```

      repara="pearson",start = c(0.01,0.01)))

fit.beta.T4<- sapply(1:(d-4), function(j) c(fit[1,j],fit[2,j]))
ll.T4<-sum(unlist(fit[3,])) #loglikelihood value in Tree 4
fam.T4<-fit[4,] # selected family of pair copulas in Tree 4

tau.T4<- sapply(1:(d-4), function(j) LDVine.rho2tau(unlist(fit.beta.T4[,j]),
                                                    t[,j][id4[,j]]))
par.T4<- sapply(1:(d-4), function(j) BiCopTau2Par(fam.T4[[j]], tau.T4[,j]))

## END fitted model

# Store parameter estimates
beta.fit <-list(fit.beta.T1,fit.beta.T2,fit.beta.T3,fit.beta.T4)

# Store the selected copula families
family.selected<-data.frame(fam.T1,fam.T2,fam.T3,fam.T4)
colnames(family.selected)<-c("fam.T1","fam.T1","fam.T1","fam.T1",
                           "fam.T2","fam.T2","fam.T2",
                           "fam.T3","fam.T3","fam.T4")

family.selected

##   fam.T1 fam.T1 fam.T1 fam.T1 fam.T2 fam.T2 fam.T2 fam.T3 fam.T3 fam.T4
## 1     4     4     4    13    14    14    13     5     1     1

# Copula log-likelihood value for HET-P model
Cloglik_HETByPair <- ll.T1+ll.T2+ll.T3+ll.T4; Cloglik_HETByPair

## [1] 178.7816

# Store the copula parameters and families for quantile prediction
par.fit.HET <- list(par.T1,par.T2,par.T3,par.T4)

fam <- unlist(family.selected)
familyset.HET <- list(c(fam[1],fam[2],fam[3],fam[4]),
                    c(fam[5],fam[6],fam[7]),
                    c(fam[8],fam[9]),c(fam[10]))

```

Conditional Quantile Prediction

We select a subject that has 5 measurements. We pretend that the selected subject has only 4 measurements rather than 5 measurements and predict the conditional 5%-, 50%- and 95%-quantiles for its fifth measurement based on the first four measurements.

```

# id of selected individual and corresponding vector of measurements
id.sel <- 453
y.sel <- MFUS[MFUS$id == id.sel, 6]
y.sel

## [1] 80 70 90 70 90

# Covariate matrix (including intercept)
X.sel <- cbind(1, as.matrix(MFUS[MFUS$id == id.sel, c(3, 4, 5)]))
# Time variable for selected id
time.sel <- MFUS[MFUS$id == id.sel, 3]

# Estimated covariate vector of the fixed effects
b <- LMM.Exp$coefficients$fixed

# Quantile levels to be estimated
alpha.level <- c(0.05, 0.5, 0.95)

```

Linear Mixed Model (LMM)

Since $\mathbf{y}_i = (y_{i,1}, y_{i,2}, y_{i,3}, y_{i,4}, y_{i,5})^\top$ is assumed to be normally distributed, the univariate conditional distribution of $y_{i,5}$ given $(y_{i,1}, y_{i,2}, y_{i,3}, y_{i,4})^\top$ is univariate normal with a certain mean and variance depending on the parameters of the distribution of \mathbf{y}_i . Therefore, we first extract the covariance structure of the linear mixed model.

```

# Range coefficient
Range.par <- as.numeric(coef(LMM.Exp$modelStruct$corStruct,
                             unconstrained = FALSE))
# Extract covariance matrix of the distribution of  $y_i = (y_{i,1}, y_{i,2},$ 
#  $y_{i,3}, y_{i,4}, y_{i,5})'$ 
d <- 5
Var.mat <- matrix(0, d, d)

for (i in 1:d){
  for (j in 1:d){
    Var.mat[i, j] <- exp(- abs(i-j)/Range.par)
  }
}
Var.mat <- Var.mat*sd.error^2 + sd.random^2

# Compute parameters of the conditional distribution of  $y_{i,5}$  given
#  $(y_{i,1}, y_{i,2}, y_{i,3}, y_{i,4})'$ 
cond.param <- condMVN(mean = X.sel%*%b,
                      sigma = Var.mat,

```

```

        dependent.ind = 1,
        given.ind = 2:5,
        X.given = y.sel[1:(d - 1)],
        check.sigma = TRUE)

mu.cond <- cond.param$condMean # mean
sigma.cond <- sqrt(cond.param$condVar) # standard deviation

# Determine conditional quantiles for LMM
y.quant1 <- qnorm(alpha.level, mean = mu.cond, sd = sigma.cond)
y.quant1

## [1] 64.12492 75.50962 86.89431

```

D-vine based copula models

For the D-vine copula models HOM-P and HET-P, we first transform the measurements to the uniform scale using the probability integral transformation. After that we calculate the D-vine quantile of the fifth measurement on the copula scale and then finally we transform these predictions to the original scale using the marginal quantile function of the fifth measurement.

```

u1.sel <- pnorm(q = y.sel[1], mean = X.sel[1, ], sd = phi)
u2.sel <- pnorm(q = y.sel[2], mean = X.sel[2, ], sd = phi)
u3.sel <- pnorm(q = y.sel[3], mean = X.sel[3, ], sd = phi)
u4.sel <- pnorm(q = y.sel[4], mean = X.sel[4, ], sd = phi)

```

Time-homogeneous D-vine copula model (HOM-P)

```

# Calculate the quantiles on the coupla scale
u_quant.p5 <- DvineQuant (p = alpha.level[1], u = data.frame(u1.sel,
    u2.sel, u3.sel, u4.sel), family.set = familyset.HOM,
    par.set = parset.HOM)
u_quant.p50 <- DvineQuant (p = alpha.level[2], u = data.frame(u1.sel,
    u2.sel, u3.sel, u4.sel), family.set = familyset.HOM,
    par.set = parset.HOM)
u_quant.p95 <- DvineQuant (p = alpha.level[3], u = data.frame(u1.sel,
    u2.sel, u3.sel, u4.sel), family.set = familyset.HOM,
    par.set = parset.HOM)

# Transform the estimated quantiles from the copula scale to the
# original scale
y_quant.p5 <- qnorm(p = u_quant.p5, mean = X.sel[5, ], sd = phi)
y_quant.p50 <- qnorm(p = u_quant.p50, mean = X.sel[5, ], sd = phi)
y_quant.p95 <- qnorm(p = u_quant.p95, mean = X.sel[5, ], sd = phi)

```

```
y.quant2<-c(y_quant.p5,y_quant.p50,y_quant.p95)
y.quant2
```

```
## [1] 65.95408 77.50066 89.47259
```

Time-heterogeneous D-vine copula model (HET-P)

```
# Estimated copula parameters from fitted HET-P model
par.id <- sapply(1:(d-1), function(j) which(id1[[j]] == id.sel))
par.T1 <- sapply(1:(d-1), function(j) par.T1[[j]][par.id[j]])

par.id <- sapply(1:(d-2), function(j) which(id2[[j]] == id.sel))
par.T2 <- sapply(1:(d-2), function(j) par.T2[[j]][par.id[j]])

par.id <- sapply(1:(d-3), function(j) which(id3[[j]] == id.sel))
par.T3 <- sapply(1:(d-3), function(j) par.T3[[j]][par.id[j]])

par.id <- sapply(1:(d-4), function(j) which(id4[,j] == id.sel))
par.T4 <- sapply(1:(d-4), function(j) par.T4[,j][par.id[j]])

# Store the estimated copula parameters from fitted HET-P model
parset.HET = list(par.T1, par.T2, par.T3, par.T4)

# Calculate the quantiles on the coupla scale
u_quant.p5 <- DvineQuant (p = alpha.level[1], u = data.frame(u1.sel,
  u2.sel, u3.sel, u4.sel), family.set = familyset.HET,
  par.set = parset.HET)
u_quant.p50 <- DvineQuant (p = alpha.level[2], u = data.frame(u1.sel,
  u2.sel, u3.sel, u4.sel), family.set = familyset.HET,
  par.set = parset.HET)
u_quant.p95 <- DvineQuant (p = alpha.level[3], u = data.frame(u1.sel,
  u2.sel, u3.sel, u4.sel), family.set = familyset.HET,
  par.set = parset.HET)

# Transform the estimated quantiles from the copula scale to the
# original scale
y_quant.p5 <- qnorm(p = u_quant.p5, mean = X.sel[5, ]%%b, sd = phi)
y_quant.p50 <- qnorm(p = u_quant.p50, mean = X.sel[5, ]%%b, sd = phi)
y_quant.p95 <- qnorm(p = u_quant.p95, mean = X.sel[5, ]%%b, sd = phi)

y.quant3<-c(y_quant.p5,y_quant.p50,y_quant.p95)
y.quant3
```

```
## [1] 66.48062 81.31704 93.44709
```

```
# Save the prediction results in "OUTPUT" folder  
# This is the part of result from table 9 (Subject 3) of manuscript.
```

```
y.quant<-rbind(y.quant1,y.quant2,y.quant3)  
rownames(y.quant)<-c("LMM","HOM","HET")  
colnames(y.quant)<-c("5%","50%","95%")  
y.quant;y.sel
```

```
##           5%       50%       95%  
## LMM 64.12492 75.50962 86.89431  
## HOM 65.95408 77.50066 89.47259  
## HET 66.48062 81.31704 93.44709
```

```
## [1] 80 70 90 70 90
```

```
write.table(round(y.quant,2),file=here("OUTPUT","y.quant.txt"))  
write.table(y.sel,file=here("OUTPUT","y.sel.txt"))
```