Supplementary Materials: Scalar Variance and Scalar Correlation for Functional Data

Cristhian Leonardo Urbano-Leon [1,†,‡,*], Manuel Escabias [1,‡], Diana Paola Ovalle-Muñoz [1,‡] and Javier Olaya-Ochoa [2,‡]

1. Simulation Code

In this supplementary material, we show the code in R language with which the brief simulation study was carried out.

```
#########################################
#######  THE BASE OF FUNCTIONS  #########
#########################################

# The basis is orthonormal, the sum of the variances per component is taken
# according to Theorem 1
k=10  #dimension of the functional subspace
t= seq(0,1, by=1/99)  # 100 points between 0 and 1
Fb = function(sk,x) sqrt(2)*cos(sk*pi*x) #sk integer (function pointer), x
     variable
p=1   # thickness of the lines
windows(title = "BASE OF FUNCTIONS")
for (i in 1:k)
  {
  par(new = TRUE)
  curve(Fb(i,x), col="black", lwd=p, xlim=c(0,1), xlab = "", ylab = "", main="Base
     of Subspace")
  }

################################################################
########### Simulation Guide Functions #########################
################################################################

# Coefficients of the guide curves on which the simulation will be based
C0 = cbind(50, 0, 0, 0, 0, 0, 0, 0, 0, 0)  # Constant Curve Guide
DF0 = function(x)  C0[1]  +  C0[2]*Fb(1,x)  +  C0[3]*Fb(2,x)  +  C0[4]*Fb(3,x)  +
      C0[5]*Fb(4,x)  +  C0[6]*Fb(5,x)  +  C0[7]*Fb(6,x)  +  C0[8]*Fb(7,x)  +  C0
     [9]*Fb(8,x)  +  C0[10]*Fb(9,x)

C1 = cbind(15, -2, -10, -50, -40, 2, -10, -5, -60, -5)  # No Constant Curve Guide
DF1 = function(x)  C1[1]  +  C1[2]*Fb(1,x)  +  C1[3]*Fb(2,x)  +  C1[4]*Fb(3,x)  +
      C1[5]*Fb(4,x)  +  C1[6]*Fb(5,x)  +  C1[7]*Fb(6,x)  +  C1[8]*Fb(7,x)  +  C1
     [9]*Fb(8,x)  +  C1[10]*Fb(9,x)

windows()
par(mfrow=c(2,1))
curve(DF0,lwd= 3,ylab="", xlab = "" ,main = "Type A curve", ylim=c(-100, 200), col
     ="black" )
curve(DF1,lwd= 3,ylab="", xlab = "" , main = "Type B curve", ylim=c(-100, 200),
     col= "black")

################################################################
########### Function to calculate the scalar variancer ##########
################################################################

FVS = function(CoefMatrix)
{
  VC=0
  for (i in 1:length(CoefMatrix[1,]))
  {
    VC[i] = (sum((CoefMatrix[,i] - mean(CoefMatrix[,i]))^2))/length(CoefMatrix[,i
    ])
  }
  VS = sum(VC)
  return(VS)}
```

```
####################################################################
######################### 2.1 Case 1 ##############################
####################################################################

nc = 50 # Number of curves to simulate
MC0 = matrix(nrow = nc, ncol = k)   # Coefficient matrix
MC1 = matrix(nrow = nc, ncol = k)   # Coefficient matrix

rmin = -10     # Minimum value of the range of variability
rmax = cbind(100, 50, 10)  # Maximum values for the ranges of variability to
    consider
linf=-250      #lower bound for graphs
lsup = 260     #upper limit for graphics
e = matrix(nrow = nc, ncol = k, 0)
lc = cbind(1,1)



windows(title = "Case 1")
par(mfrow = c(2,3))
for (j in 1:length(rmax))
{
  e[,lc] <- runif(nc, min = rmin, max = rmax[j])
  for (i in 1:nc)
  {
    MC0[i,]   <-  C0 + e[i,]
  }

  Vartext = paste("Variance", as.character(round( FVS(MC0), 2) ), sep = " = ",
    collapse = NULL)

  curve(DF0, ylim = c(linf, lsup), ylab = "",xlab="",  main= paste("Type A: Case 1
     .",as.character(j)), cex.main=3)
  for (i in 1:nc)
  {
    DF0 = function(x)  MC0[i,1]  + MC0[i,2]*Fb(1,x)  + MC0[i,3]*Fb(2,x)  + MC0[
    i,4]*Fb(3,x)  + MC0[i,5]*Fb(4,x)  + MC0[i,6]*Fb(5,x)  + MC0[i,7]*Fb(6,x)
    + MC0[i,8]*Fb(7,x)  + MC0[i,9]*Fb(8,x)  + MC0[i,10]*Fb(9,x)
    par(new = TRUE)
    curve(DF0, col = "blue", ylim = c(linf, lsup), ylab = "", xlab = "", xaxt = "n
    ", yaxt = "n" )
  }
  legend(0.1, -140, Vartext, box.col = "lightblue", bg = "lightblue", adj = 0.3,
    cex = 2.3)

}

for (j in 1:length(rmax))
{
  e[,lc] <- runif(nc, min = rmin, max = rmax[j])
  for (i in 1:nc)
  {
    MC1[i,]   <-  C1 + e[i,]
  }

  Vartext = paste("Variance", as.character(round( FVS(MC1), 2) ), sep = " = ",
    collapse = NULL)


  curve(DF1, ylim = c(linf, lsup), ylab = "",xlab="",  main= paste("Type B: Case 1
     .", as.character(j)), cex.main=3)
  for (i in 1:nc)
  {
    DF1 = function(x)  MC1[i,1]  + MC1[i,2]*Fb(1,x)  + MC1[i,3]*Fb(2,x)  + MC1[
    i,4]*Fb(3,x)  + MC1[i,5]*Fb(4,x)  + MC1[i,6]*Fb(5,x)  + MC1[i,7]*Fb(6,x)
    + MC1[i,8]*Fb(7,x)  + MC1[i,9]*Fb(8,x)  + MC1[i,10]*Fb(9,x)
    par(new = TRUE)
```

```
    curve(DF1, col = "blue", ylim = c(linf, lsup), ylab = "", xlab = "", xaxt = "n
    ", yaxt = "n" )
  }
  legend(0.1, -140, Vartext, box.col = "lightblue", bg = "lightblue", adj = 0.3,
    cex = 2.3)

}

##################################################################
############################ Case 2 ############################
##################################################################

nc = 50
MC0 = matrix(nrow = nc, ncol = k)
MC1 = matrix(nrow = nc, ncol = k)
rmean0 = C0
rmean1 = C1
rvar = cbind(30, 20, 15, 10, 5, 2)
linf =-350
lsup = 350
e = matrix(nrow = nc, ncol = k, 0)
lc = cbind(1, 8)


windows(title = "Caso 2")
par(mfrow = c(2,3))
for (j in 1:length(rmax))
{
  for (i in 1:k)
  {
    MC0[,i]   <-  rnorm(nc, mean = rmean0[i], sd = rvar[j])
  }

  Vartext0 = paste("Variance", as.character(round( FVS(MC0), 2) ), sep = " = ",
    collapse = NULL)


  curve(DF0, ylim = c(linf, lsup), ylab = "", xlab = "", main = paste("Type A:
    Case 2 .", as.character(j) ), cex.main=3  )
  for (i in 1:nc)
  {
    DF0 = function(x) MC0[i,1]  + MC0[i,2]*Fb(1,x)  + MC0[i,3]*Fb(2,x)  + MC0[
    i,4]*Fb(3,x)  + MC0[i,5]*Fb(4,x)  + MC0[i,6]*Fb(5,x)  + MC0[i,7]*Fb(6,x)
    + MC0[i,8]*Fb(7,x)  + MC0[i,9]*Fb(8,x)  + MC0[i,10]*Fb(9,x)
    par(new = TRUE)
    curve(DF0, col = "blue", ylim = c(linf, lsup), ylab = "", xlab = "", xaxt = "n
    ", yaxt = "n")
  }
  legend(0.1, -180, Vartext0, box.col = "lightblue", bg = "lightblue", adj = 0.3,
    cex = 2.3)

}

for (j in 1:length(rmax))
{
  e[,lc] <- runif(nc, min = rmin, max = rmax[j])
  for (i in 1:k)
  {
    MC1[,i]   <-  rnorm(nc, mean = rmean1[i], sd = rvar[j])
  }

  Vartext1 = paste("Variance", as.character(round( FVS(MC1), 2) ), sep = " = ",
    collapse = NULL)


  curve(DF1, ylim = c(linf, lsup), ylab = "", xlab = "", main = paste("Type B:
    Case 2 .", as.character(j) ), cex.main=3 )
  for (i in 1:nc)
  {
```

```
    DF1 = function(x)  MC1[i,1]  + MC1[i,2]*Fb(1,x)  + MC1[i,3]*Fb(2,x)  + MC1[
    i,4]*Fb(3,x)  + MC1[i,5]*Fb(4,x)  + MC1[i,6]*Fb(5,x)  + MC1[i,7]*Fb(6,x)
    + MC1[i,8]*Fb(7,x)  + MC1[i,9]*Fb(8,x)  + MC1[i,10]*Fb(9,x)
    par(new = TRUE)
    curve(DF1, col = "blue", ylim = c(linf, lsup), ylab = "", xlab = "", xaxt = "n
    ", yaxt = "n")
  }
  legend(0.1, -180, Vartext1, box.col = "lightblue", bg = "lightblue", adj = 0.3,
    cex=2.3)

}


######## To check the consistency ####

set.seed(10)

nc = 500 # Number of curves to simulate (The population)
MC = matrix(nrow = nc, ncol = k)   # Matrix of population coefficients
rmean = C0    # Means to be considered by component (of guide curve)
rvar = 30     # variability to consider
linf =-570    #lower bound for graphs
lsup = 650    #upper bound for graphs
for (i in 1:k) # Here the coefficients of the population are simulated
{
  MC[,i]   <-  rnorm(nc, mean = rmean[i], sd = rvar)
}

Vartextpob = paste("Population Variance", as.character(round( FVS(MC), 2) ), sep =
     " = ", collapse = NULL)


windows(title =  "Population of functional data considered")
curve(DF0, ylim = c(linf, lsup), ylab = "")
for (i in 1:nc)
{
  DF = function(x)  MC[i,1]  + MC[i,2]*Fb(1,x)  + MC[i,3]*Fb(2,x)  + MC[i,4]*Fb
    (3,x)  + MC[i,5]*Fb(4,x)  + MC[i,6]*Fb(5,x)  + MC[i,7]*Fb(6,x)  + MC[i
    ,8]*Fb(7,x)  + MC[i,9]*Fb(8,x)  + MC[i,10]*Fb(9,x)
  par(new = TRUE)
  curve(DF, col = "blue", ylim = c(linf, lsup), ylab = "")
}
legend(0.1, -500, Vartextpob, box.col = "lightblue", bg = "lightblue", adj = 0.1)


#######
### Getting samples
#####

TM = c( 5, 10, 20, 50, 100, 150, 200, 250, 300, 400, 450) #Sample sizes to
    consider

RM = 500 #resamples

DVar = matrix(ncol = length(TM) , nrow = RM )  # Save the difference between the
    sample and the population variance
Vvar = matrix(ncol = length(TM) , nrow = RM )  # Save the sample and population
    variances
for (i in 1:RM)
{
  for (j in 1:length(TM))
  {
    tm = TM[j]  # Sample sizes
    IM = sample(1:nc, tm, replace = FALSE )
    MCM = MC[IM,] # MCM is the matrix of the sample coefficients
    VartextMuestra = paste("Sample Variance", as.character(round( FVS(MCM), 2) ),
    sep = " = ", collapse = NULL)
    Vvar[i,j] = FVS(MCM)
```

```
    DVar[i,j] = FVS(MCM) - FVS(MC) #Difference Between Sample and Population
    Variance
  }
}

FVar=0
for (j in 1:length(TM))
{
  FVar[j] = abs(mean(DVar[,j]))
}



windows(title = "Variance behavior")
plot(TM, FVar, pch=16 ,type="b", lwd=2 , main = "Variance behavior", xlab = "
    Sample Size", ylab = "Mean absolute difference" )
abline(h=0 )


rbind(round(TM,1), round(FVar, 2) )
```