

Análisis de datos funcionales

Análisis de datos funcionales

CAMILO AVELLANEDA^a

AUTOR

DEPARTAMENTO DE ESTADÍSTICA, FACULTAD DE CIENCIAS, UNIVERSIDAD NACIONAL DE COLOMBIA, BOGOTÁ, COLOMBIA

1. Preguntas

- a. Qué representa la función aleatoria?
- b. Usando regresión spline ajuste una curva para cada uno de los registros de la base de datos. Indique qué base de funciones empleó y justifique su elección. Indique el número de bases recomendado para lograr el mejor ajuste y explique cómo llega a ese número
- c. Usando suavizamiento spline ajuste cada una de las observaciones espectrales. Indique que base de funciones empleó y cuál es el valor del parámetro de suavizamiento. Justifique su respuesta
- d. Seleccione arbitrariamente una curva estimada y estime su varianza
- e. Para la curva seleccionada presente su banda de confianza.
- f. Para cada grupo encuentre estimaciones
 - i. La función media.
 - ii. La función media recortada al 10 %.
 - iii. La función varianza.
 - iv. La función covarianza.
- g. Encuentre la primera y segunda derivada de cada grupo.
- h. Encuentre las eigenfunctions que expliquen el 90 % de la variabilidad de la función aleatoria.
- i. Seleccione una curva cualquiera y presente su representación usando la descomposición de Karhunen-Loeve.

^aEstudiante de estadística. E-mail: caavellanedag@unal.edu.co

2. Desarrollo

- a. Las funciones denotadas por $y(t)$ corresponden a mediciones en diferentes espectros de absorbanza infrarroja de trozos de carne. La realización de valores recolectados corresponden a 100 valores de dichas mediciones en 100 espectros diferentes. A cada función le corresponde una medición de humedad, grasa y proteína de las diferentes unidades tenidas en cuenta dentro del estudio. A nivel de bases de datos, en términos del data frame incluido en el paquete *caret* [1] de R [2] se encuentran dos tablas de datos. La primera contiene la información de la absorbanza correspondiente a cada una de las unidades en los diferentes espectros, mientras que la segunda tabla contiene la información correspondiente a la humedad, grasa y proteína. Mediante la variable de grasa se contemplarán dos grupos, el primero con valores superiores a 20 % y el segundo con valores menores o iguales a dicho umbral.
- b. Para determinar el tipo de bases a utilizar para realizar el tipo de ajuste mediante la regresión se tienen dos consideraciones. La primera de ellas es si la función presenta algún tipo de periodicidad. Las figuras 1 y 2 muestran las curvas construidas mediante segmentos de líneas unidos a partir de las realizaciones de los 215 trozos de carne incluidos. La primer figura muestra las curvas observadas en un solo grupo, mientras que la segunda imagen considera los grupos segmentando por la partición realizada mediante su porcentaje de grasa. En ambos casos se observa que no hay una periodicidad. De esta manera las funciones base a utilizar son B -splines. La segunda consideración es el grado del spline a ajustar. Por defecto, la literatura apunta al uso de splines cúbicos, por lo cual el grado considerado es cuatro.

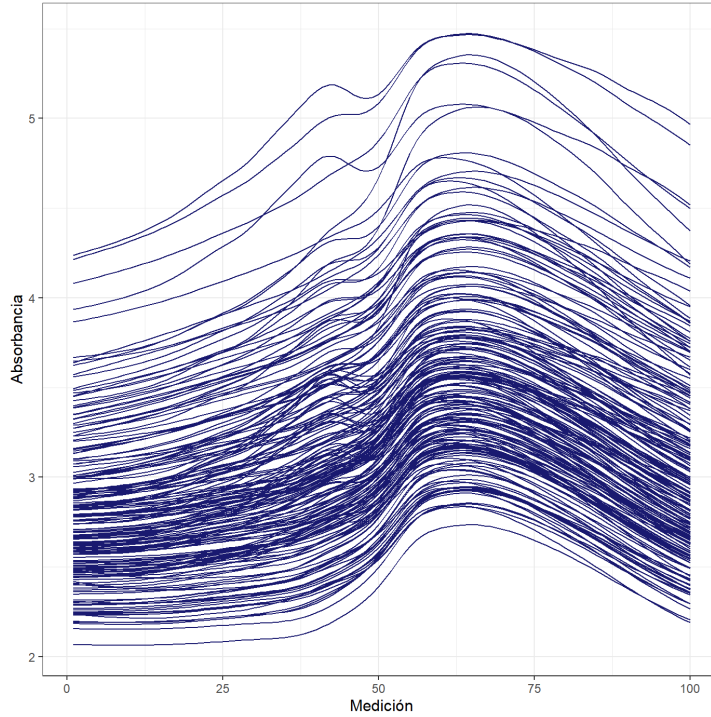


FIGURA 1: Realizaciones de las absorbanzas en los 215 trozos de carne en los 100 espectros diferentes.

Por otra parte, para encontrar el número de bases óptimo, se utiliza la partición del cuadrado medio del error dada entre su sesgo al cuadrado y su varianza. Es decir, lo que se muestra a continuación:

$$CME[\hat{y}(t)] = Sesgo^2[\hat{y}(t)] + Var[\hat{y}(t)], \quad (1)$$

donde $Sesgo[\hat{y}(t)] = \hat{y}(t) - y(t)$, $Var[\hat{y}(t)] = \hat{\sigma}^2 \Phi(\Phi^t \Phi)^{-1} \Phi^t$, Φ es la matriz de diseño o de variables explicativas conformada por el conjunto de funciones base y $\hat{\sigma}^2$ es una estimación de la varianza de los

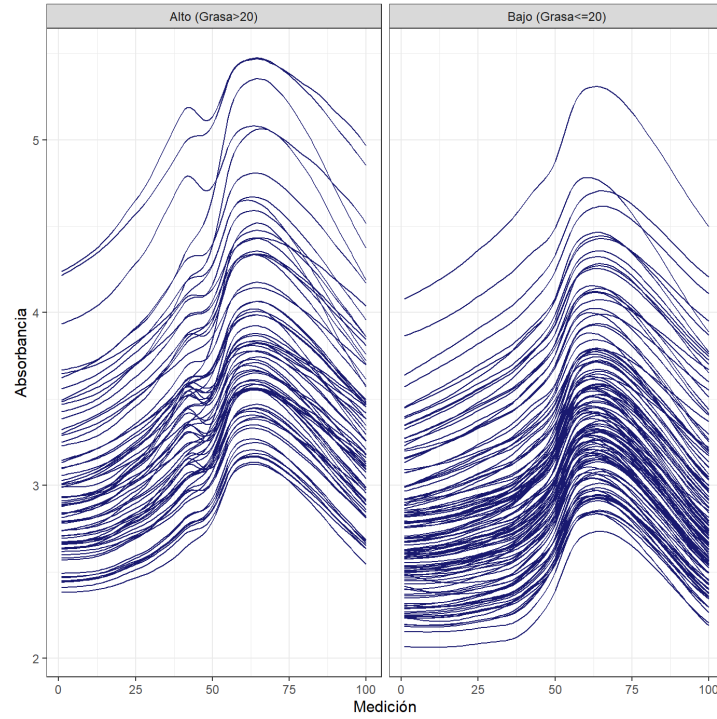


FIGURA 2: Realizaciones de las absorbancias en los 215 trozos de carne en los 100 espectros diferentes, segmentando por los grupos a partir de la variable de porcentaje de grasa.

errores de la regresión realizada a partir del cuadrado medio del error de la misma. El anterior resultado hace uso de la teoría clásica de regresión con múltiples covariables, la cual asume independencia entre las observaciones. Este supuesto se tomó por simplicidad, pero el caso en el que no se tiene la independencia entre las realización del proceso se puede abordar de manera análoga mediante la inclusión de una matriz de varianzas y covarianzas \mathbf{W} .

Al realizar el cálculo de los valores utilizados en la ecuación (1) para cada una de las curvas y para una variedad de número de bases, se obtuvo la figura 3, la cual ilustra el sesgo al cuadrado (línea azul), la varianza del estimador (línea verde) y el cuadrado medio del error (línea roja), en donde se observa que éste último deja de decrecer al rededor de 30 funciones base, por lo cual este valor se toma como óptimo. Los valores en dicha figura corresponden a totales de las 215 curvas realizadas en los diferentes valores de los números de bases. Al realizar este gráfico se esperaba ver un comportamiento al alza de la varianza del estimador al igual que del cuadrado medio del error a partir de determinado umbral, lo cual no se dio en este caso dado que al llegar a 30 funciones base la curva no presenta demasiadas curvaturas u oscilaciones y a su vez residuales cercanos a 0, lo cual se ve reflejado en una varianza del estimador que decrece a medida que el número de funciones aumenta. En adelante, en lo que resta de este documento se consideran 34 funciones base como número óptimo.

- c. Para la determinación del suavizamiento spline óptimo a partir del conjunto de datos se desea obtener un parámetro de penalización adecuado λ . Para ello se genera una secuencia de valores y se calculan las potencias de 10 a partir de dicha secuencia. En cada uno de estos valores se calcula el coeficiente de validación cruzada generalizada, el cual consiste en determinar el cuadrado medio del error dejando fuera los diferentes puntos correspondientes a la realización en los 100 espectros en este caso. Lo anterior se realiza para cada una de las 215 curvas consideradas y se totaliza para tener un único valor en cada uno de los valores generados en la secuencia inicial.

La figura 4 muestra los valores de la validación cruzada generalizada para valores de $\log_{10}(\lambda)$ entre -5 y 5. Al realizar la verificación, el valor óptimo corresponde a $\log_{10}(\lambda) = -1.75$ o lo que es equivalente

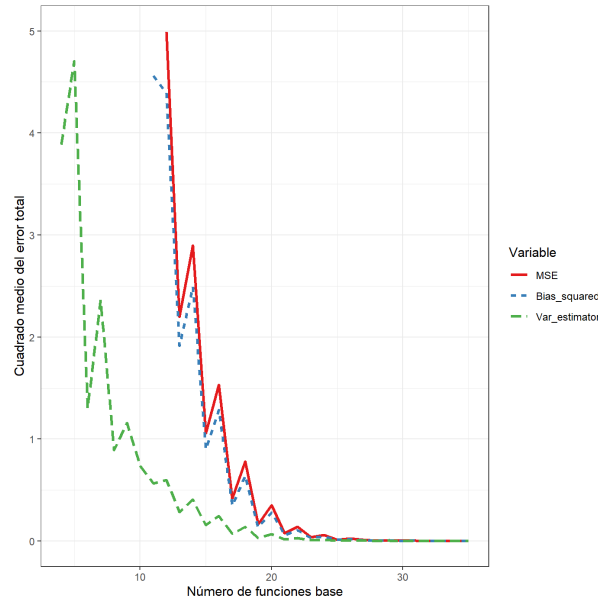


FIGURA 3: Valores del cuadrado medio del error y de su descomposición entre el sesgo al cuadrado y la varianza del predictor, según el número de bases utilizado.

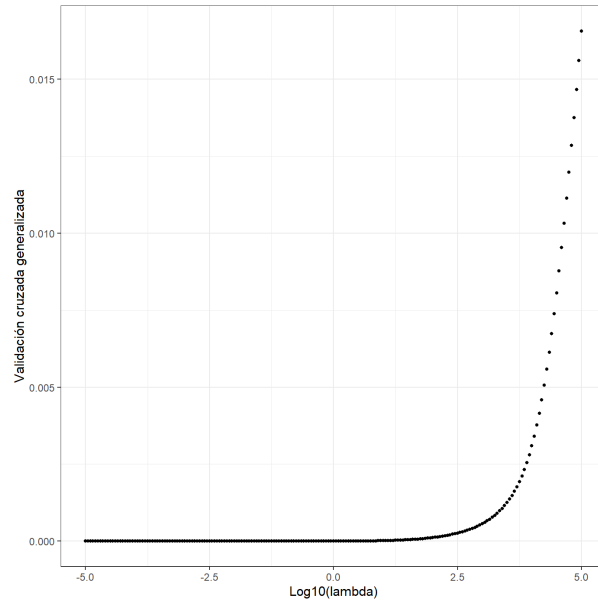


FIGURA 4: Valores de la validación cruzada generalizada calculados a partir de los diferentes valores de $\log_{10}(\lambda)$, los cuales fueron generados entre -5 y 5.

a $\lambda = 0.0178$, el cual es el parámetro de penalización por curvatura. En adelante, en lo que resta de este documento, las curvas ajustadas considerarán ese valor del parámetro de suavizado.

- d. De manera arbitraria se seleccionó la curva número 75 de la muestra de 215 curvas. En este caso la estimación de su varianza se realizó mediante la formula dada en la ecuación (2)

$$Var_X(t) = (n - 1)^{-1} \sum_{i=1}^n [x_i(t) - \bar{x}(t)]^2. \quad (2)$$

En este caso, el valor obtenido para la ecuación (2) es de 0.07189111.

- e. Al realizar la estimación de la varianza de la curva número 75 y calculando su respectivo intervalo de confianza, se obtiene la figura 5, donde se visualiza la función estimada en la línea sólida, mientras que en las líneas punteadas se encuentran los límites inferior y superior, respectivamente.

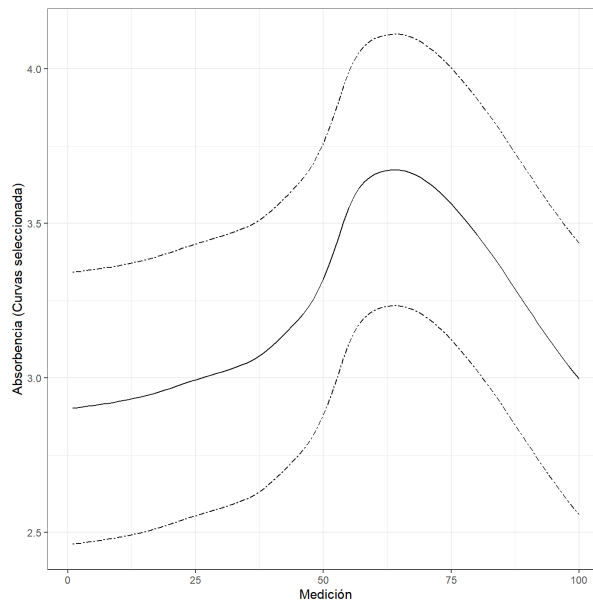


FIGURA 5: Curva estimada y su respectiva banda de confianza para la curva número 75.

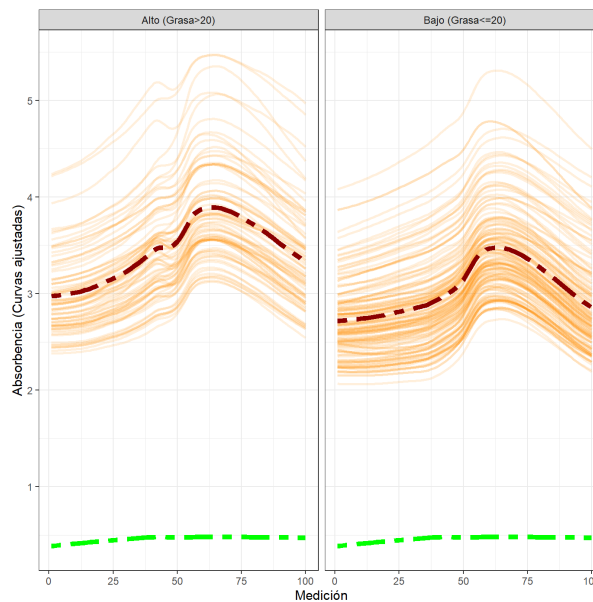


FIGURA 6: Funciones estimadas junto con sus funciones de media (curva en rojo oscuro) y desviación estándar (curva en verde) para los dos grupos.

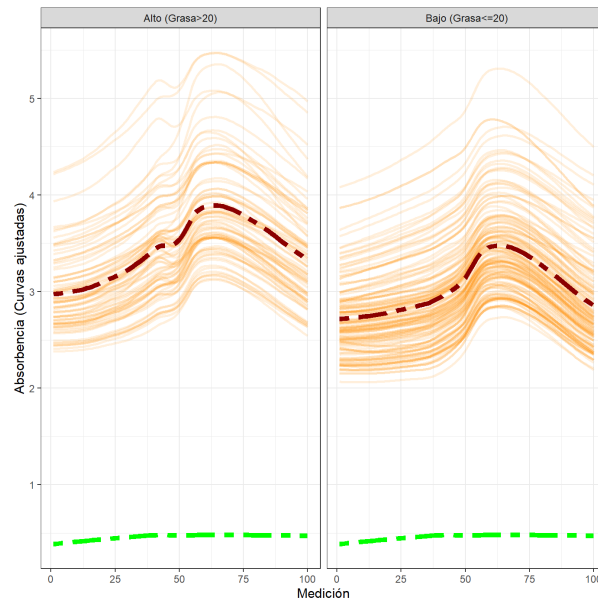


FIGURA 7: Funciones estimadas junto con sus funciones de media recortada al 10 % (curva en rojo oscuro) y desviación estándar (curva en verde) para los dos grupos.

f. La estimación de las curvas promedio y de la desviación estándar se realizó por grupos. La figura 6 muestra las curvas ajustadas por grupos de grasa junto con su respectiva función promedio en color rojo oscuro y la desviación estándar en color verde. De manera análoga, la figura 7 muestra el promedio recortado al 10 % y la desviación estándar en color verde.

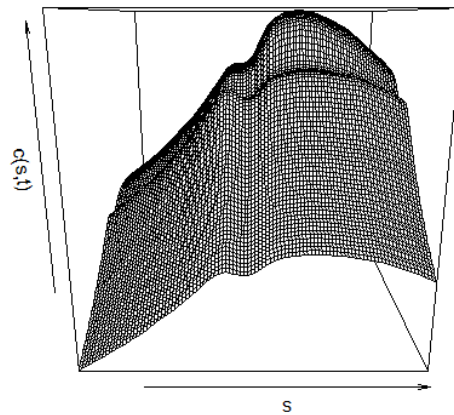


FIGURA 8: Superficie que representa la función de varianza y covarianza del proceso para el grupo de trozos altos en grasa.

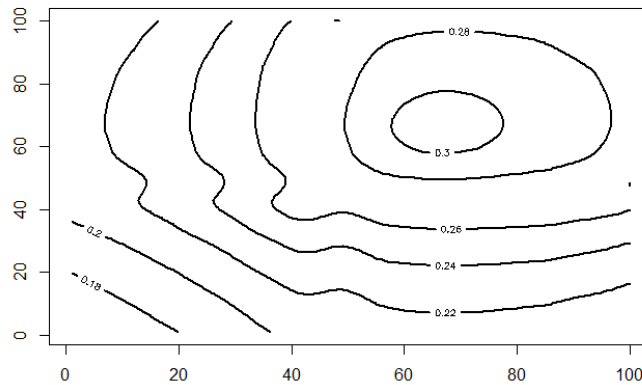


FIGURA 9: Curvas de nivel que representan la función de varianza y covarianza del proceso para el grupo de trozos altos en grasa.

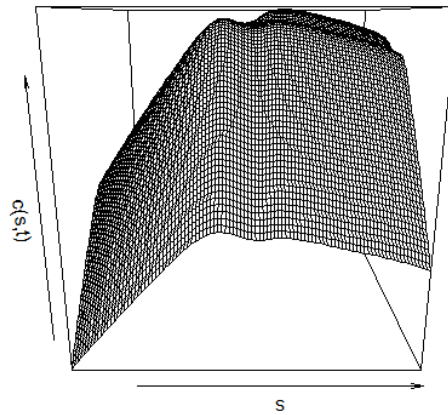


FIGURA 10: Superficie que representa la función de varianza y covarianza del proceso para el grupo de trozos bajos en grasa.

La superficie que se ilustra en la figura 8 muestra la función $c(s, t)$ para el grupo alto en grasa, que representa la función de varianza y covarianza. A manera complementaria, la figura 10 muestra las respectivas curvas de nivel para la dicha función. De manera análoga, las figuras 10 y 11 muestran la superficie y las curvas de nivel asociadas a la función de varianza y covarianza en el grupo bajo en grasa.

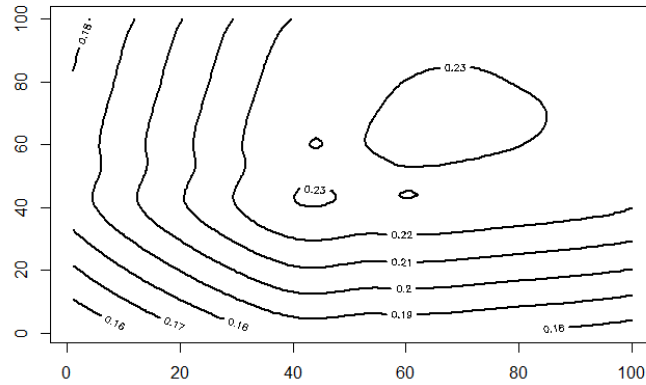


FIGURA 11: Curvas de nivel que representan la función de varianza y covarianza del proceso para el grupo de trozos bajos en grasa.

g. Mediante la función *deriv.fd* del paquete *fda* se realizó la estimación de la primera y segunda derivada para las funciones de los dos grupos. Las figuras 12 y 13 muestran la primera y segunda derivada de las curvas de absorbancia en los diferentes espectros considerados, respectivamente.

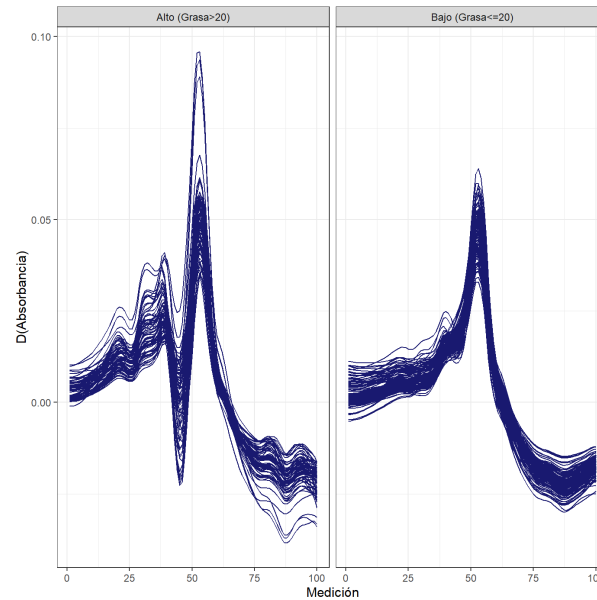


FIGURA 12: Curvas correspondientes a la primera derivada de las curvas estimadas en puntos anteriores, según su grupo de grasa.

h. Al realizar la descomposición en términos de las componentes principales funcionales se encontró que la primera de éstas retiene el 98 % de la variabilidad total. De esta manera, dicha componente principal funcional se muestra en la figura 14. La obtención de esta función se realizó mediante la función *pca.fd* del paquete *fda*.

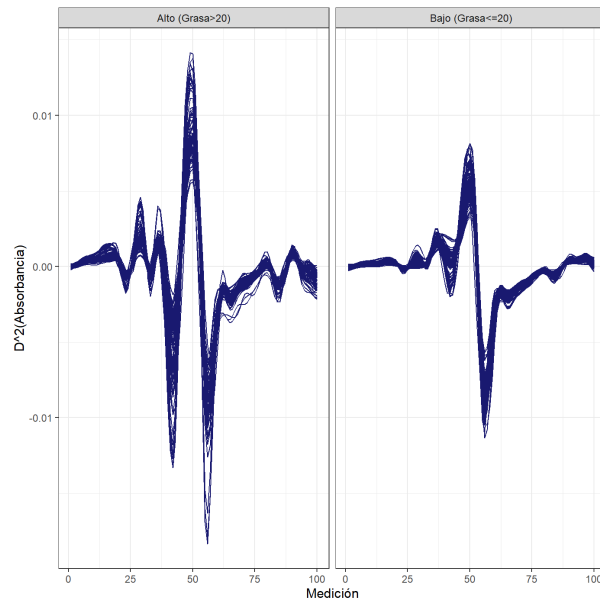


FIGURA 13: Curvas correspondientes a la segunda derivada de las curvas estimadas en puntos anteriores, según su grupo de grasa.

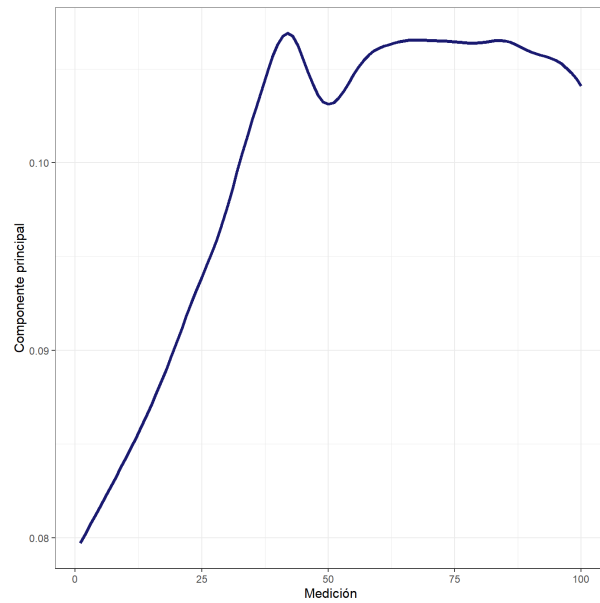


FIGURA 14: Gráfico de la primer componente principal funcional.

- i. La descomposición de Karhunen-Loeve se muestra en la ecuación (3), en donde el vector $\hat{\xi}$ es un vector de coeficientes estimados a partir de la ecuación de las funciones propias. Por otro lado $\hat{\mathbf{v}}$ corresponde a un conjunto de funciones base. El objetivo de esta descomposición es encontrar las funciones propias que mayor porcentaje de variabilidad retengan, de manera similar a lo que sucede en los componentes principales provenientes de la estadística multivariada.

$$y_n(t) - \bar{y}(t) \approx \sum_{i=1}^p \hat{\xi}_{ni} \hat{v}_i(t) = \hat{\xi}^t \hat{\mathbf{v}} \quad (3)$$


```

absorp_4<-merge(absorp_3,predict_fd,by.x=c("Curva","Medicion"),by.y=c("Variable","labels"),all.x=TRUE)

absorp_4_altos<-absorp_4[Grupo=="Alto (Grasa>20)",c("Curva","Medicion","Absorbencia")]
absorp_4_bajos<-absorp_4[Grupo=="Bajo (Grasa<=20)",c("Curva","Medicion","Absorbencia")]

absorp_4_altos<-dcast(absorp_4_altos,Medicion~Curva,value.var="Absorbencia")
absorp_4_altos<-absorp_4_altos[,~"Medicion"]

absorp_4_bajos<-dcast(absorp_4_bajos,Medicion~Curva,value.var="Absorbencia")
absorp_4_bajos<-absorp_4_bajos[,~"Medicion"]

fit_fd_altos=smooth.basis(y=as.matrix(absorp_4_altos), fdParobj=fdParobj)
fit_fd_bajos=smooth.basis(y=as.matrix(absorp_4_bajos), fdParobj=fdParobj)

mean_fd_altos=mean.fd(fit_fd_altos$fd)
trimmed_mean_fd_altos=mean.fd(fit_fd_altos$fd,trim=0.1)
std_fd_altos=std.fd(fit_fd_altos$fd)

mean_fd_bajos=mean.fd(fit_fd_bajos$fd)
trimmed_mean_fd_bajos=mean.fd(fit_fd_bajos$fd,trim=0.1)
std_fd_bajos=std.fd(fit_fd_bajos$fd)

summary_stats<-data.frame(
  rbind(
    data.frame(Medicion=1:100,Mean=predict(mean_fd_bajos,newdata=1:100),Sd=predict(std_fd_bajos,newdata=1:100),Grupo="Bajo (Grasa<=20)"),
    data.frame(Medicion=1:100,Mean=predict(mean_fd_altos,newdata=1:100),Sd=predict(std_fd_bajos,newdata=1:100),Grupo="Alto (Grasa>20)"))

predict(std_fd_altos,newdata=1:100)

ggplot(absorp_4)+
  geom_line(data=absorp_4, aes(x=Medicion,y=Fitted,group=Curva), color="darkorange",size=1,alpha=0.14)+
  geom_line(data=summary_stats,aes(x=Medicion,y=mean),size=2,color="darkred",linetype="twodash")+
  geom_line(data=summary_stats,aes(x=Medicion,y=Sd),size=2,color="green",linetype="twodash")+
  guides(color=FALSE)+theme_bw()+xlab("Medición")+ylab("Absorbencia (Curvas ajustadas)")+
  facet_grid(.~Grupo)

# mean_fd=predict(mean_pinch,newdata=1:nrow(pinch))
# sd_fd=predict(sd_pinch,newdata=1:nrow(pinch))
# summary_stats=data.frame(labels=1:nrow(pinch),mean_fd,sd_fd) %>% as.data.table() %>%
# melt(id.vars="labels",value.name="Value",variable.name="Stat")

var_fd_bajos<-var.fd(fit_fd_bajos$fd)
var_fd_altos<-var.fd(fit_fd_altos$fd)

grid=(1:100)
W.cov.mat=eval.bifd(grid, grid, var_fd_bajos)
persp(grid, grid, W.cov.mat, xlab="s",
      ylab="t", zlab="c(s,t)")
contour(grid, grid, W.cov.mat, lwd=2)

W.cov.mat=eval.bifd(grid, grid, var_fd_altos)
persp(grid, grid, W.cov.mat, xlab="s",
      ylab="t", zlab="c(s,t)")
contour(grid, grid, W.cov.mat, lwd=2)

GCV<-map_dfc(1:ncol(t(absorp)),function(.y){
  map_dbl(seq(-5,5,by=0.05),function(.x){
    lambda_10<-10^-x
    spline_basis=create.bspline.basis(rangeval=c(1,100),nbasis=34)
    fdParobj = fdPar(fdobj=spline_basis, Lfdobj=2, lambda=lambda_10)
    absorb_fd=smooth.basis(y=t(absorp)[,y], fdParobj=fdParobj)
    #mean((eval.fd(1:100, fdobj=absorb_fd$fd) - t(absorp)[,y])^2)
    absorb_fd$gcv
  })
})

df_gcv<-data.frame(log10_lambda=seq(-5,5,by=0.05),
  lambda=10^log10_lambda,
  GCV=rowMeans(GCV))
df_gcv[which.min(df_gcv$GCV),]

.y<-75
.x<-34
#lambda_10<-10^-10
y<-t(absorp)[,y]
#y = (x-20)^2+200*norm(1000)0
x = 1:100
# put noise to the true curve and generate noisy data
spline_basis=create.bspline.basis(rangeval=c(1,100),nbasis=.x)
fdParobj = fdPar(fdobj=spline_basis, Lfdobj=2, lambda=10^(-1.75))
#fdParobj = fdPar(fdobj=spline_basis, Lfdobj=2, lambda=0)
absorb_fd=smooth.basis(y=y, fdParobj=fdParobj)
# Bias_2<-t(eval.fd(x, fdobj=absorb_fd$fd) - y) %*% (eval.fd(x, fdobj=absorb_fd$fd) - y)
CME<-Bias_2/(100-.x)
X<-eval.basis(1:100, spline_basis)
Var_estimator<-diag(X%*% solve(t(X) %*% X) %*% t(X))*CME

return(data.frame(Prediction=mean.fd(absorb_fd$fd)
  Error=sqrt(Var_estimator),
  LI=mean.fd(absorb_fd$fd)-200*sqrt(Var_estimator),
  LS=mean.fd(absorb_fd$fd)+200*sqrt(Var_estimator)))
#data.frame(Num_function=.y,Num_basis=.x,Bias_squared=Bias_2,Var_estimator=Var_estimator,MSE=Var_estimator+Bias_2)
}
selection_curve_df<-as.data.table(Selection_curve_confidence_band(.x=34,.y=75))
selection_curve_df<-selection_curve_df[,labels=1:100]
ggplot()+geom_line(data=selection_curve_df,aes(labels,Prediction))+
  geom_line(data=selection_curve_df,aes(labels,LI),linetype="twodash")+

```

```

descriptive_1<-absorp_2 %>% ggplot()+geom_line(aes(Medicion,Absorbancia,group=Var2),color="midnightblue")+
  guides(color=F)+theme_bw()+xlab("Medición")+ylab("Absorbancia")

descriptive_2<-absorp_3 %>%
  ggplot()+
  geom_line(aes(Medicion,Absorbancia,group=Var2),color="midnightblue")+
  guides(color=F)+theme_bw()+xlab("Medición")+ylab("Absorbancia")+
  facet_grid(.~Grupo)

tiff(paste0(date,"_Descriptive_1_.tiff"), units="in", width=7, height=7, res=200)
print({
  descriptive_1
})
dev.off()

tiff(paste0(date,"_Descriptive_2_.tiff"), units="in", width=7, height=7, res=200)
print({
  descriptive_2
})
dev.off()

tiff(paste0(date,"_Selection_num_basis_.tiff"), units="in", width=7, height=7, res=200)
print({
  df[,.(MSE=sum(MSE),Bias_squared=sum(Bias_squared),
    Var_estimator=sum(Var_estimator)),
    by=(Num_basis)] %>% melt(id.vars="Num_basis",variable.name="Variable",value.name="Value") %>%
  ggplot()+geom_line(aes(x=Num_basis,Value,linetype=Variable,color=Variable),size=1.1)+
  scale_color_brewer(palette="Set1")+theme_bw()+ylab("Cuadrado medio del error total")+
  xlab("Número de funciones base")+ylim(0,5)
})
dev.off()

tiff(paste0(date,"_Selection_lambda_.tiff"), units="in", width=7, height=7, res=200)
print({
  df_gcv %>% ggplot()+geom_point(aes(x=log10_lambda,GCV),size=1)+theme_bw()+
  ylab("Validación cruzada generalizada")+xlab("Log10(lambda)")
})
dev.off()

absorp_selected_curve<-absorp_4[,c("Curva","Medicion","Absorbancia")] %>% dcast(Medicion~Curva,value.var="Absorbancia")
absorp_selected_curve<-absorp_selected_curve[,-"Medicion"]
fit_selected_curve=smooth.basis(y=as.matrix(absorp_selected_curve), fdParobj=fdParobj)
std_fd=sd(fit_selected_curve$fd)

fdParobj = fdPar(fdobj=spline_basis, Lfdobj=2, lambda=10^(-1.75))
selected_curve_fd<-smooth.basis(y=as.matrix(absorp_3[Curva==75,"Absorbancia"]), fdParobj=fdParobj)
df_selected<-data.frame(fitted=predict(selected_curve_fd),se=predict(std_fd,newdata=1:100)) %>% as.data.table()
var(as.matrix(absorp_3[Curva==75,"Absorbancia"]))

tiff(paste0(date,"_Selected_curve_bandwidth_.tiff"), units="in", width=7, height=7, res=200)
print({
  df_selected[,c("labels","LI","LS"):=list(1:100,fitted-1.64*sqrt(0.07189111),
    fitted+1.64*sqrt(0.07189111))] %>%
  ggplot()+
  geom_line(aes(labels,fitted))+geom_line(aes(labels,LI),linetype="twodash")+
  geom_line(aes(labels,LS),linetype="twodash")+ylab("Absorbancia (Curvas seleccionada)")+
  xlab("Medición")+theme_bw()
})
dev.off()

mean_fd_alto=mean.fd(fit_fd_alto$fd)
trimmed_mean_fd_alto=mean.fd(fit_fd_alto$fd,trim=0.1)
std_fd_alto=std.fd(fit_fd_alto$fd)

mean_fd_bajo=mean.fd(fit_fd_bajo$fd)
trimmed_mean_fd_bajo=mean.fd(fit_fd_bajo$fd,trim=0.1)
std_fd_bajo=std.fd(fit_fd_bajo$fd)

summary_stats<-data.frame(
  rbind(
    data.frame(Medicion=1:100,Mean=predict(mean_fd_bajos,newdata=1:100),Sd=predict(std_fd_bajos,newdata=1:100),Grupo="Bajo (Grasa<=20)"),
    data.frame(Medicion=1:100,Mean=predict(mean_fd_alto,newdata=1:100),Sd=predict(std_fd_alto,newdata=1:100),Grupo="Alto (Grasa>20)"))

tiff(paste0(date,"_Mean_sd_groups.tiff"), units="in", width=7, height=7, res=200)
print({
  ggplot(absorp_4)+
  geom_line(data=absorp_4 ,aes(x=Medicion,y=Fitted,group=Curva),color="darkorange",size=1,alpha=0.14)+
  geom_line(data=summary_stats,aes(x=Medicion,y=mean),size=2,color="darkred",linetype="twodash")+
  geom_line(data=summary_stats,aes(x=Medicion,y=Sd),size=2,color="green",linetype="twodash")+
  guides(color=FALSE)+theme_bw()+xlab("Medición")+ylab("Absorbancia (Curvas ajustadas)")+
  facet_grid(.~Grupo)
})
dev.off()

mean_fd_alto=mean.fd(fit_fd_alto$fd)
trimmed_mean_fd_alto=mean.fd(fit_fd_alto$fd,trim=0.1)
std_fd_alto=std.fd(fit_fd_alto$fd)

mean_fd_bajo=mean.fd(fit_fd_bajo$fd)
trimmed_mean_fd_bajo=mean.fd(fit_fd_bajo$fd,trim=0.1)
std_fd_bajo=std.fd(fit_fd_bajo$fd)

summary_stats_2<-data.frame(
  rbind(

```

```

data.frame(Medicion=1:100,Mean=predict(trimmed_mean_fd_bajos,newdata=1:100),Sd=predict(std_fd_bajos,newdata=1:100),Grupo="Bajo (Grasa<=20)"),
data.frame(Medicion=1:100,Mean=predict(trimmed_mean_fd_altos,newdata=1:100),Sd=predict(std_fd_bajos,newdata=1:100),Grupo="Alto (Grasa>20)"))

tiff(paste0(date,"_Trimmed_mean_sd_per_groups.tiff"), units="in", width=7, height=7, res=200)
print({
  ggplot(absorp_4)+
    geom_line(data=absorp_4 ,aes(x=Medicion,y=Fitted,group=Curva),color="darkorange",size=1,alpha=0.14)+
    geom_line(data=summary_stats_2,aes(x=Medicion,y=mean),size=2,color="darkred",linetype="twodash")+
    geom_line(data=summary_stats_2,aes(x=Medicion,y=Sd),size=2,color="green",linetype="twodash")+
    guides(color=FALSE)+theme_bw()+xlab("Medición")+ylab("Absorbencia (Curvas ajustadas)")+
    facet_grid(.~Grupo)
})
dev.off()

#Derivadas

deriv_altos_1<-deriv.fd(fit_fd_altos$fd,1)
deriv_altos_1.1<-predict(deriv_altos_1,1:100) %>% as.data.table()
names(deriv_altos_1.1)<-paste0("X",1:ncol(deriv_altos_1.1))

deriv_bajos_1<-deriv.fd(fit_fd_bajos$fd,1)
deriv_bajos_1.1<-predict(deriv_bajos_1,1:100) %>% as.data.table()
names(deriv_bajos_1.1)<-paste0("X",1:ncol(deriv_bajos_1.1))

deriv_altos_2<-deriv.fd(fit_fd_altos$fd,2)
deriv_altos_2.1<-predict(deriv_altos_2,1:100) %>% as.data.table()
names(deriv_altos_2.1)<-paste0("X",1:ncol(deriv_altos_2.1))

deriv_bajos_2<-deriv.fd(fit_fd_bajos$fd,2)
deriv_bajos_2.1<-predict(deriv_bajos_2,1:100) %>% as.data.table()
names(deriv_bajos_2.1)<-paste0("X",1:ncol(deriv_bajos_2.1))

derivadas_1<-rbind(deriv_altos_1.1 %>% mutate(labels=1:100) %>%
  melt(id.vars="labels",variable.name="Variable",value.name="Valor") %>% mutate(Grupo="Alto (Grasa>20)"),
  deriv_bajos_1.1 %>% mutate(labels=1:100) %>%
  melt(id.vars="labels",variable.name="Variable",value.name="Valor") %>% mutate(Grupo="Bajo (Grasa<=20)"))

derivadas_2<-rbind(deriv_altos_2.1 %>% mutate(labels=1:100) %>%
  melt(id.vars="labels",variable.name="Variable",value.name="Valor") %>% mutate(Grupo="Alto (Grasa>20)"),
  deriv_bajos_2.1 %>% mutate(labels=1:100) %>%
  melt(id.vars="labels",variable.name="Variable",value.name="Valor") %>% mutate(Grupo="Bajo (Grasa<=20)"))

tiff(paste0(date,"_Primera_derivada.tiff"), units="in", width=7, height=7, res=200)
print({
  derivadas_1 %>% ggplot()+geom_line(aes(labels,Valor,group=Variable),color="midnightblue")+
    guides(color=F)+theme_bw()+xlab("Medición")+ylab("D(Absorbancia)")+facet_grid(.~Grupo)
})
dev.off()

tiff(paste0(date,"_Segunda_derivada.tiff"), units="in", width=7, height=7, res=200)
print({
  derivadas_2 %>% ggplot()+geom_line(aes(labels,Valor,group=Variable),color="midnightblue")+
    guides(color=F)+theme_bw()+xlab("Medición")+ylab("D^2(Absorbancia)")+facet_grid(.~Grupo)
})
dev.off()

W.pca$values
W.pca = pca.fd(fit_selected_curve$fd, nharm=1)
plot(W.pca$harmonics, lwd=3)
sum(W.pca$varprop)
class(W.pca)

tiff(paste0(date,"_Componente_principal.tiff"), units="in", width=7, height=7, res=200)
print({
  data.table(labels=1:100,Valor=predict(W.pca$harmonics,newdata=1:100)) %>%
    ggplot()+geom_line(aes(labels,Valor,PC1),color="midnightblue",size=1.2)+
    ylab("Componente principal")+xlab("Medición")+theme_bw()
})
dev.off()

```

Referencias

- [1] Max Kuhn. caret: Classification and Regression Training, 2020. R package version 6.0-86.
- [2] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2020.
- [3] J. O. Ramsay, Spencer Graves, and Giles Hooker. fda: Functional Data Analysis, 2020. R package version 5.1.5.
- [4] James O Ramsay. Functional data analysis. Encyclopedia of Statistical Sciences, 4, 2005.
- [5] JO Ramsay. Matlab, r and s-plus functions for functional data analysis. McGill University, 2003.