Códigos\_Ondaletas.txt

######################################################

#################### APLICACAO 1 ####################

######################################################

###### EXEMPLO 1 ######

library(wavethresh)

 par(mfrow=c(3,2))

 draw(filter.number=4, family="DaubExPhase", enhance=FALSE, main="")

 draw(filter.number=6, family="DaubExPhase", enhance=FALSE, main="")

 draw(filter.number=4, family="DaubLeAsymm", enhance=FALSE, main="")

 draw(filter.number=10, family="DaubLeAsymm", enhance=FALSE, main="")

 draw(filter.number=2, family="DaubExPhase", enhance=FALSE, main="")

 draw(filter.number=8, family="DaubLeAsymm", enhance=FALSE, main="")

##############################################################################

##############################################################################

##############################################################################

###### EXEMPLO 2 ###### SIMULADO

 x<-1:1024

 a1<-1.5\*sin(pi\*x/6)

 a2<-2\*sin(pi\*x/12)

 a3<-4\*sin(pi\*x/18)

 par(mfrow=c(2,2))

 plot.ts(a1, ylim=c(-4.5,4.5))

 plot.ts(a2, ylim=c(-4.5,4.5),col="blue")

 plot.ts(a3, ylim=c(-4.5,4.5),col="red")

 plot.ts(a1, ylim=c(-4.5,4.5))

 lines(a2, col="blue")

 lines(a3, col="red")

 #Grafico de Combinacao

 par(mfrow=c(3,1))

 plot.ts(a1+a2+a3)

 plot.ts(a1+a3)

 B=rep(0,1024)

 B[1:512]<-(a1+a2+a3)[1:512]

 B[513:1024]<-(a1+a3)[1:512]

 plot.ts(B)

 #### Analise Fourier para serie "suma" ####

 suma<-a1+a2+a3

 par(mfrow=c(2,1))

 plot.ts(suma)

 per<-spec.pgram(suma, log="no") # Grafica da densidade espectral estimada

 names(per)

 per$freq # Freqs donde se calculan as Densidades espectrales

 per$spec # Densidades espectrales estimadas

 wmax<-per$freq[which.max(per$spec)] # Em qual freq atinge o maximo

 (s<-1/wmax) #Para calcular o periodo principal

 #### Analise de Fourier para serie "B" ####

 plot.ts(B)

 per1<-spec.pgram(B, log="no") # Grafico da densidad espectral estimada

 par(mfrow=c(3,1)) # Nao podemos identificar a mudanca

 plot.ts(B)

 per<-spec.pgram(suma, log="no")

 per1<-spec.pgram(B, log="no")

 #### Analise com Ondaletas para serie "B" ####

 tB<-scale(B)

 tB.dwt<-wd(tB, filter.number = 8)

 plot(tB.dwt)

 names(tB.dwt)

 tB.dwt$nlevels # Numero de niveis

 length(tB.dwt$D) # Coeficientes Detalhes

 #Energia total suave nivel 5, 2^(J-1)<n^1/2<2^J -> J=5

 ETs<-sum(accessC(tB.dwt, level=5)^2)

 length(accessC(tB.dwt, level=5))

 plot.ts(accessC(tB.dwt, level=5), type="h")

#Energia por nivel

# Ed10<-sum(accessD(tB.dwt, level=0)^2) #2^0 coeficientes

# Ed9<-sum(accessD(tB.dwt, level=1)^2) #2^1 "

# Ed8<-sum(accessD(tB.dwt, level=2)^2) #2^2 "

# Ed7<-sum(accessD(tB.dwt, level=3)^2) #2^3 "

# Ed6<-sum(accessD(tB.dwt, level=4)^2) #2^4 "

 Ed5<-sum(accessD(tB.dwt, level=5)^2) #2^5 "

 Ed6<-sum(accessD(tB.dwt, level=6)^2) #2^6 "

 Ed7<-sum(accessD(tB.dwt, level=7)^2) #2^7 "

 Ed8<-sum(accessD(tB.dwt, level=8)^2) #2^8 "

 Ed9<-sum(accessD(tB.dwt, level=9)^2) #2^9 "

 ETd<-Ed9+Ed8+Ed7+Ed6+Ed5 #Energia total detalhes

 #Energia total

 ET<-ETs+ETd

 #Energia oportado or nivel

 ETs/ET #Proporcao de energia suave

 Ed5/ET #Proporcao de energia nivel 5

 Ed6/ET #Proporcao de energia nivel 6

 Ed7/ET #Proporcao de energia nivel 7

 Ed8/ET #Proporcao de energia nivel 8

 Ed9/ET #Proporcao de energia nivel 9

barplot(100\*c(ETs/ET, Ed5/ET, Ed6/ET, Ed7/ET, Ed8/ET, Ed9/ET),

 names.arg=c("s5","d5","d6","d7","d8","d9"),main="Contribucao por nivel")

##############################################################################

##############################################################################

##############################################################################

###### EXEMPLO 3 ###### BASE DE DADOS

 library(wavethresh)

y<-simchirp()

z<-scale(y$y)

length(z)

zwd <- wd(z, filter.number=2, family="DaubExPhase")

par(mfrow=c(2,1))

plot.ts(z)

plot(zwd, scaling="by.level", main="")

length(z)

########################

#Coeficientes suaves

Ts<-sum(accessC(zwd, level=5)^2)

plot(accessC(zwd, level=5), type="h")

#Coeficientes finos

#Td10<-sum(accessD(zwd, level=0)^2) #2^0 coeficientes

#Td9<-sum(accessD(zwd, level=1)^2) #2^1 "

#Td7<-sum(accessD(zwd, level=3)^2) #2^3 "

#Td6<-sum(accessD(zwd, level=4)^2) #2^4 "

Td5<-sum(accessD(zwd, level=5)^2) #2^5 "

Td6<-sum(accessD(zwd, level=6)^2) #2^6 "

Td7<-sum(accessD(zwd, level=7)^2) #2^7 "

Td8<-sum(accessD(zwd, level=8)^2) #2^8 "

Td9<-sum(accessD(zwd, level=9)^2) #2^9 "

Td<-Td9+Td8+Td7+Td6+Td5 #Energia total detalhes

#Energia total

TP<-Ts+Td

#Energia oportado or nivel

Ts/TP #Proporcao de energia suave

Td5/TP #Proporcao de energia nivel 5

Td6/TP #Proporcao de energia nivel 6

Td7/TP #Proporcao de energia nivel 7

Td8/TP #Proporcao de energia nivel 8

Td9/TP #Proporcao de energia nivel 9

barplot(100\*c(Ts/TP,Td5/TP,Td6/TP,Td7/TP,Td8/TP,Td9/TP),

 names.arg=c("s5","d5","d6","d7","d8","d9"),main="Contribucao por nivel")

###### EXEMPLO 3 ALTERNATIVO ###### BASE DE DADOS

library(waveslim)

y<-simchirp()

z<-scale(y$y)

length(z)

zwd<-dwt(z, "d8", n.levels = 9)

names(zwd)

par(mfrow=c(5,1))

plot(zwd$s9, type="h")

plot(zwd$d5, type="h")# 2^5

plot(zwd$d4, type="h")# 2^6

plot(zwd$d3, type="h")# 2^7

plot(zwd$d2, type="h")# 2^8

plot(zwd$d1, type="h")# 2^9

##############################################################################

##############################################################################

##############################################################################

###### EXEMPLO 4 ###### SERIES DE BBABY ECG

library(wavethresh)

data(BabyECG)

dba<-diff(BabyECG)

length(BabyECG)

plot.ts(dba)

baby = c(scale(dba),0) # standardize the series e comprimento adequado

plot.ts(baby)

ba.dwt = wd(baby, filter.number=8)

plot(ba.dwt)

#Coeficientes suaves

BAs<-sum(accessC(ba.dwt, level=6)^2)

plot(accessC(ba.dwt, level=6), type="h")

#Coeficientes finos

Bd6<-sum(accessD(ba.dwt, level=6)^2) #2^6 "

Bd7<-sum(accessD(ba.dwt, level=7)^2) #2^7 "

Bd8<-sum(accessD(ba.dwt, level=8)^2) #2^8 "

Bd9<-sum(accessD(ba.dwt, level=9)^2) #2^9 "

Bd10<-sum(accessD(ba.dwt, level=10)^2) #2^10 "

#Energia total detalhes

BAd<-Bd6+Bd7+Bd8+Bd9+Bd10

#Energia total

PBA<-BAs+BAd

#Energia oportado por nivel

BAs/PBA #Proporcao de energia suave

Bd6/PBA #Proporcao de energia nivel 6

Bd7/PBA #Proporcao de energia nivel 7

Bd8/PBA #Proporcao de energia nivel 8

Bd9/PBA #Proporcao de energia nivel 9

Bd10/PBA #Proporcao de energia nivel 10

barplot(100\*c(BAs/PBA, Bd6/PBA, Bd7/PBA, Bd8/PBA, Bd9/PBA, Bd10/PBA),

 names.arg=c("s6","d6","d7","d8","d9","d10"),main="Contribucao por nivel")

##############################################

############ LIMIARIZACAO BabyECG ############

par(mfrow=c(3,1))

baby.dwt=wd(baby, filter.number = 8)

baby.smo=wr(threshold(baby.dwt,levels=6:10))

ts.plot(baby, main="Dados")

ts.plot(baby.smo, main="Sinal")

ts.plot(baby-baby.smo, main="Residuais")

################################# FIM APLICACAO 1 ###############################

######################################################

#################### APLICACAO 2 ####################

######################################################

###### ONDALETA HAAR ######

set.seed(512)

e<-rnorm(512,0,1)

TT<-512; t<-1:512

fl12<-function(s){-1\*cos(pi\*(2\*s/TT))}

at<-fl12(t)

plot.ts(at)

xt<-rep(0,512)

for(i in 1:511){

 xt[i+1]<-at[i+1]\*xt[i]+e[i+1]

}

length(xt)

plot.ts(xt)

acf(xt, lag.max = 100)

pacf(xt, lag.max = 100)

###############################

library(waveslim)

tamanho.T<-512 # filas

valor.J<-5 #2^{J-1}<sqrt{T}<2^{J}

tamanho.J<-2^(valor.J) # 32 colunas

############Criar funcao que calcula as funcoes wavelet HAAR

matrix.coef<-function(k){vetor<-rep(0,tamanho.T)

vetor[k]<-1

wav.dwt<-c(dwt(vetor,wf="haar",n.levels=6)$s6, dwt(vetor,wf="haar",n.levels=6)$d6, #1+1#

 dwt(vetor,wf="haar",n.levels=6)$d5, dwt(vetor,wf="haar",n.levels=6)$d4, #2+4#

 dwt(vetor,wf="haar",n.levels=6)$d3, dwt(vetor,wf="haar",n.levels=6)$d2)#, #8+16# = 32 = tamanho.J

#dwt(vetor,wf="d8",n.levels=9)$d1)#, dwt(vetor,wf="d8",n.levels=9)$d3,

#dwt(vetor,wf="d8",n.levels=9)$d2, dwt(vetor,wf="d8",n.levels=9)$d1)

vetor.Psi<-wav.dwt[1:tamanho.J]

vetor.Psi} # coloca los primeros tamanho.J wavelet en el tiempo k.

############Criar matrix contendo os valores de phi e psi

matrix.wav<-matrix(0, tamanho.T, tamanho.J) #512x32

dim(matrix.wav)

indiceU<-seq(1,tamanho.T,1)

matrix.wav<-t(apply(as.array(indiceU),1, matrix.coef))

#############Construcao das matrizes Psi

ThetaX<-matrix(0,tamanho.T-1,tamanho.J)

dim(ThetaX)

for(i in 1:511){

 ThetaX[i,]<-matrix.wav[i+1,]\*xt[i]

}

dim(ThetaX)

Beta<-(solve(t(ThetaX)%\*%ThetaX))%\*%(t(ThetaX))%\*%xt[2:512]

length(Beta)

hat<-matrix.wav%\*%Beta

plot.ts(hat)

lines(at)

##############################################################################

##############################################################################

##############################################################################

###### ONDALETA D8 ######

set.seed(512)

e<-rnorm(512,0,1)

TT<-512; t<-1:512

fl<-function(s){-1\*cos(pi\*(2\*s/TT))}

at<-fl(t)

plot.ts(at)

xt<-rep(0,512)

for(i in 1:511){

 xt[i+1]<-at[i+1]\*xt[i]+e[i+1]

}

length(xt)

plot.ts(xt)

acf(xt, lag.max = 100)

pacf(xt, lag.max = 100)

###############################

tamanho.T<-512 # filas

valor.J<-5 #2^{J-1}<sqrt{T}<2^{J}

tamanho.J<-2^(valor.J) # 32 colunas

############Criar funcao que calcula as funcoes wavelet HAAR

matrix.coef<-function(k){vetor<-rep(0,tamanho.T)

vetor[k]<-1

wav.dwt<-c(dwt(vetor,wf="d8",n.levels=6)$s6, dwt(vetor,wf="d8",n.levels=6)$d6, #1+1#

 dwt(vetor,wf="d8",n.levels=6)$d5, dwt(vetor,wf="d8",n.levels=6)$d4, #2+4#

 dwt(vetor,wf="d8",n.levels=6)$d3, dwt(vetor,wf="d8",n.levels=6)$d2)#, #8+16# = 32 = tamanho.J

#dwt(vetor,wf="d8",n.levels=9)$d1)#, dwt(vetor,wf="d8",n.levels=9)$d3,

#dwt(vetor,wf="d8",n.levels=9)$d2, dwt(vetor,wf="d8",n.levels=9)$d1)

vetor.Psi<-wav.dwt[1:tamanho.J]

vetor.Psi} # coloca los primeros tamanho.J wavelet en el tiempo k.

############Criar matrix contendo os valores de phi e psi

matrix.wav<-matrix(0, tamanho.T, tamanho.J) #512x32

dim(matrix.wav)

indiceU<-seq(1,tamanho.T,1)

matrix.wav<-t(apply(as.array(indiceU),1, matrix.coef))

#############Construcao das matrizes Psi

ThetaX<-matrix(0,tamanho.T-1,tamanho.J)

dim(ThetaX)

for(i in 1:511){

 ThetaX[i,]<-matrix.wav[i+1,]\*xt[i]

}

dim(ThetaX)

Beta<-(solve(t(ThetaX)%\*%ThetaX))%\*%(t(ThetaX))%\*%xt[2:512]

length(Beta)

hat<-matrix.wav%\*%Beta

plot.ts(hat)

lines(at)

##############################################################################

##############################################################################

##############################################################################

###### ONDALETA D8 COEFICIENTE PERIODICO ######

set.seed(512)

e<-rnorm(512,0,1)

TT<-512; t<-1:512

fl1<-function(s){-1\*cos(pi\*(6\*s/TT))}

at<-fl1(t)

plot.ts(at)

xt<-rep(0,512)

for(i in 1:511){

 xt[i+1]<-at[i+1]\*xt[i]+e[i+1]

}

length(xt)

plot.ts(xt)

acf(xt, lag.max = 100)

pacf(xt, lag.max = 100)

###############################

tamanho.T<-512 # filas

valor.J<-5 #2^{J-1}<sqrt{T}<2^{J}

tamanho.J<-2^(valor.J) # 32 colunas

############Criar funcao que calcula as funcoes wavelet HAAR

matrix.coef<-function(k){vetor<-rep(0,tamanho.T)

vetor[k]<-1

wav.dwt<-c(dwt(vetor,wf="d8",n.levels=6)$s6, dwt(vetor,wf="d8",n.levels=6)$d6, #1+1#

 dwt(vetor,wf="d8",n.levels=6)$d5, dwt(vetor,wf="d8",n.levels=6)$d4, #2+4#

 dwt(vetor,wf="d8",n.levels=6)$d3, dwt(vetor,wf="d8",n.levels=6)$d2)#, #8+16# = 32 = tamanho.J

#dwt(vetor,wf="d8",n.levels=9)$d1)#, dwt(vetor,wf="d8",n.levels=9)$d3,

#dwt(vetor,wf="d8",n.levels=9)$d2, dwt(vetor,wf="d8",n.levels=9)$d1)

vetor.Psi<-wav.dwt[1:tamanho.J]

vetor.Psi} # coloca los primeros tamanho.J wavelet en el tiempo k.

############Criar matrix contendo os valores de phi e psi

matrix.wav<-matrix(0, tamanho.T, tamanho.J) #512x32

dim(matrix.wav)

indiceU<-seq(1,tamanho.T,1)

matrix.wav<-t(apply(as.array(indiceU),1, matrix.coef))

#############Construcao das matrizes Psi

ThetaX<-matrix(0,tamanho.T-1,tamanho.J)

dim(ThetaX)

for(i in 1:511){

 ThetaX[i,]<-matrix.wav[i+1,]\*xt[i]

}

dim(ThetaX)

Beta<-(solve(t(ThetaX)%\*%ThetaX))%\*%(t(ThetaX))%\*%xt[2:512]

length(Beta)

hat<-matrix.wav%\*%Beta

plot.ts(hat)

lines(at)

##############################################################################

##############################################################################

##############################################################################

###### ONDALETA HAAR COEFICIENTE PERIODICO ######

set.seed(512)

e<-rnorm(512,0,1)

TT<-512; t<-1:512

fl2<-function(s){-1\*cos(pi\*(6\*s/TT))}

at<-fl2(t)

plot.ts(at)

xt<-rep(0,512)

for(i in 1:511){

 xt[i+1]<-at[i+1]\*xt[i]+e[i+1]

}

length(xt)

plot.ts(xt)

acf(xt, lag.max = 100)

pacf(xt, lag.max = 100)

###############################

tamanho.T<-512 # filas

valor.J<-5 #2^{J-1}<sqrt{T}<2^{J}

tamanho.J<-2^(valor.J) # 32 colunas

############Criar funcao que calcula as funcoes wavelet HAAR

matrix.coef<-function(k){vetor<-rep(0,tamanho.T)

vetor[k]<-1

wav.dwt<-c(dwt(vetor,wf="haar",n.levels=6)$s6, dwt(vetor,wf="haar",n.levels=6)$d6, #1+1#

 dwt(vetor,wf="haar",n.levels=6)$d5, dwt(vetor,wf="haar",n.levels=6)$d4, #2+4#

 dwt(vetor,wf="haar",n.levels=6)$d3, dwt(vetor,wf="haar",n.levels=6)$d2)#, #8+16# = 32 = tamanho.J

#dwt(vetor,wf="d8",n.levels=9)$d1)#, dwt(vetor,wf="d8",n.levels=9)$d3,

#dwt(vetor,wf="d8",n.levels=9)$d2, dwt(vetor,wf="d8",n.levels=9)$d1)

vetor.Psi<-wav.dwt[1:tamanho.J]

vetor.Psi} # coloca los primeros tamanho.J wavelet en el tiempo k.

############Criar matrix contendo os valores de phi e psi

matrix.wav<-matrix(0, tamanho.T, tamanho.J) #512x32

dim(matrix.wav)

indiceU<-seq(1,tamanho.T,1)

matrix.wav<-t(apply(as.array(indiceU),1, matrix.coef))

#############Construcao das matrizes Psi

ThetaX<-matrix(0,tamanho.T-1,tamanho.J)

dim(ThetaX)

for(i in 1:511){

 ThetaX[i,]<-matrix.wav[i+1,]\*xt[i]

}

dim(ThetaX)

Beta<-(solve(t(ThetaX)%\*%ThetaX))%\*%(t(ThetaX))%\*%xt[2:512]

length(Beta)

hat<-matrix.wav%\*%Beta

plot.ts(hat)

lines(at)

################################# FIM APLICACAO 2 ###############################