Series de tiempo (ST)

Los modelos de ST son útiles cuando se tienen datos correlacionados en serie.

Este tipo de modelos se usan principalmente para realizar pronósticvos y predicciones

Para las series de tiempo se necesitan datos que basados en el tiempo como:

años, días, horas, minutos y segundos.

Una ST es una secuencia de observaciones medidas en determinados momentos del tiempo

ordenados cronológicamente y espaciados entre sí de manera uniforme, así los datos

son dependientes entre sí.

El objetivo de una ST es realizar un pronóstico a través del modelaje de los datos.

Ejemplos de ST

- Número de habitantes por año

- PIB anual

- Índices de precio del pretróleo

- Número de accidentes de carro semanal

- Precio de venta de acciones de la BMV diario

## Componentes de una ST

El análisis de ST está basado en el supuesto de que los valores observados son

consecuancia de tres componentes:

a) Tendencia: se define como un cambio a largo plazo en relación a los valores

observados. La tendencia se identifica gráficamente con un

movimiento suave de la serie a largo plazo.

b) Estacional: Algunas ST presentan cierta peridiocidad o variaciones en

cierto periodo de tiempo, a esta serie se le conoce como

estacionaria. Por ejemplo:

- Las ventas de flores aumentan considerablemente en el mes de mayo.

- Las ventas de juguetes se incrementan en el mes de diciembre y

enero.

c) Aleatorio: Estas ST no responden ningún patrón de comportamiento si no que,

se deriva de fenómenos aleatorios que impactan de manera aislada

a una ST.

# Clasificación de las ST

Las ST se pueden clasificar en:

1) Estacionarias: una ST se conoce como estacionaria cuando la media y

la varianza son constantes en el tiempo.

2) No estacionarias: una ST se considera no estacionaria cuando la tendencia

o variabilidad cambien en el tiempo.

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

data("AirPassengers")

class(AirPassengers)

start(AirPassengers) #Año del primer dato

end(AirPassengers) #Año del último dato

frequency(AirPassengers) #(12) datos mensuales

summary(AirPassengers) #Resumen estadístico

plot(AirPassengers, col="red", type="h") #Tendencia a la alza

abline(mean(AirPassengers))

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

#################### CLASE 01 FEBRERO 2017 ############################

temporal <- tempfile()

download.file("http://www.beta.inegi.org.mx/contenidos/proyectos/enchogares/regulares/enoe/microdatos/enoe\_15ymas/2016/2016trim1\_dbf.zip", temporal)

files =unzip(temporal, list=TRUE)$Name

unzip(temporal, files=files[grepl("dbf", files)])

install.packages("foreign")

require(foreign)

SDEMT116 <- data.frame(read.dbf("sdemt116.dbf"))

####################### EJERCICIO FILA 3 #############################

temporal <- tempfile()

download.file("http://www.beta.inegi.org.mx/contenidos/proyectos/enchogares/modulos/motral/2015/microdatos/motral2015\_bases\_datos\_dbf.zip", temporal)

files =unzip(temporal, list=TRUE)$Name

unzip(temporal, files=files[grepl("dbf", files)])

install.packages("foreign")

require(foreign)

MOTRAL2015 <- data.frame(read.dbf("motral2015\_empleos.dbf"))

###################################################### 08 02 17

pob <- sample(100:120,15,replace = F)

pobts <- ts(pob,frequency = 1,start = 2000) #1 si es anual

end(pobts) #Cuando termina la serie de tiempo

start(pobts) #Cuando empieza

plot(aggregate(pobts)) #Graficar la ST

plot(pobts)

infor <- ts(read.csv("C:\\Users\\SALA-D32\\Desktop\\TIL.csv"),

frequency = 4,start = 2005)

infor

plot(infor)

#Descargar 4 series de tiempo

#http://www3.inegi.org.mx/Sistemas/infoenoe/Default\_CONAPO.aspx?s=est&c=27736

var <- data.frame(read.csv("C:\\Users\\SALA-D32\\Desktop\\variables.csv"))

desocupacion <- ts(var[,1],frequency = 4, start = 2005) #Solo puedes llamar 1 serie a la vez

ocupacionparcial <- ts(var[,2],frequency = 4, start = 2005)

trabajo asalariado <- ts(var[,3],frequency = 4, start = 2005)

concriticas <- ts(var[,4],frequency = 4, start = 2005)

###################################################### 10 02 17

# 10 de febrero del 2017

# Generar 4 objetos con datos aleatorios y que tengan 60 datos

# Objeto 1 # de profesionistas (millones) rango de 5 a 8

# Objeto 2 crecimiento del PIB, rango de 0 a 5

# Objeto 3 porcentaje de ocupados, rango de 40 a 60

# Objeto 4 porcentaje de desocupados, rengo de 3 a 8

profesionistas <- sample(5:8, 60, replace = TRUE)

crec\_PIB <- sample(0:5, 60, replace = TRUE)

ocupados\_por <- sample(40:60, 60, replace = TRUE)

desocupados\_por <- sample(3:8, 60, replace = TRUE)

# Generar data. frame

base1 <- data.frame(profesionistas, crec\_PIB, ocupados\_por, desocupados\_por)

# Mostrar en consola los primeros 12 y los últimos 12 datos

base1 [1:12,] #primeros 12 datos

base1 [49:60,] #últimos datos

base2 <- base1[1:12,]

# Convertir los objetos en series de tiempo comenzando en 2010 mensualmente

profesionistas\_serie <- ts(profesionistas, frequency = 12, start = 2010)

profesionistas\_serie

crec\_PIB\_serie <- ts(crec\_PIB, frequency = 12, start = 2010)

crec\_PIB\_serie

ocupados\_por\_serie <- ts(ocupados\_por, frequency = 12, start = 2010)

ocupados\_por\_serie

desocupados\_por\_serie <- ts(desocupados\_por, frequency = 12, start = 2010)

desocupados\_por\_serie

# Graficar series de tiempo en una sola gráfica (profesionistas y ocupados)

plot(cbind(profesionistas\_serie, ocupados\_por\_serie))

#Graficar serie de tiempo (PIB y desocupados)

plot(cbind(crec\_PIB\_serie, desocupados\_por\_serie))

#Graficar todas las series de tiempo

plot(cbind(profesionistas\_serie, ocupados\_por\_serie, crec\_PIB\_serie, desocupados\_por\_serie))

base3 <- data.frame(profesionistas\_serie,crec\_PIB\_serie,ocupados\_por\_serie,desocupados\_por\_serie)

plot.ts(base3[,c(1,3)])

PIB\_Mexico <- data.frame(read.csv("C:\\Users\\SALA-D31\\Desktop\\PIB México.csv"))

PIB\_1980a1990 <- ts(PIB\_Mexico[1:11,], frequency = 1, start = 1980)

PIB\_1980a1990

PIB\_1991a2000 <- ts(PIB\_Mexico[12:21,], frequency = 1, start = 1991)

PIB\_1991a2000

PIB\_2001a2010 <- ts(PIB\_Mexico[22:31,], frequency = 1, start = 2001)

PIB\_2001a2010

PIB\_2011a2015 <- ts(PIB\_Mexico[32:36,], frequency = 1, start = 2011)

PIB\_2011a2015

plot(cbind(PIB\_1991a2000,PIB\_2001a2010))

############################ 13 de febrero 2017 #########################################

profesionistas <- sample(5:8, 60, replace = TRUE)

crec\_PIB <- sample(0:5, 60, replace = TRUE)

ocupados\_por <- sample(40:60, 60, replace = TRUE)

desocupados\_por <- sample(3:8, 60, replace = TRUE)

# Convertir los objetos en series de tiempo comenzando en 2010 mensualmente

profesionistas\_serie <- ts(profesionistas, frequency = 12, start = 2010)

profesionistas\_serie

crec\_PIB\_serie <- ts(crec\_PIB, frequency = 12, start = 2010)

crec\_PIB\_serie

ocupados\_por\_serie <- ts(ocupados\_por, frequency = 12, start = 2010)

ocupados\_por\_serie

desocupados\_por\_serie <- ts(desocupados\_por, frequency = 12, start = 2010)

desocupados\_por\_serie

# Graficar series de tiempo en una sola gráfica (profesionistas y ocupados)

plot(cbind(profesionistas\_serie, ocupados\_por\_serie))

#Graficar serie de tiempo (PIB y desocupados)

plot(cbind(crec\_PIB\_serie, desocupados\_por\_serie))

#Graficar todas las series de tiempo

plot(cbind(profesionistas\_serie, ocupados\_por\_serie, crec\_PIB\_serie, desocupados\_por\_serie))

base3 <- data.frame(profesionistas\_serie,crec\_PIB\_serie,ocupados\_por\_serie,desocupados\_por\_serie)

plot.ts(base3[,c(1,3)])

seriemultiple <- ts.intersect(desocupados\_por\_serie, profesionistas\_serie, ocupados\_por\_serie, desocupados\_por\_serie )

plot(seriemultiple, main ="Serie de tiempo", xlab="Años", ylab = "Número de personas", col = "red",

type= "overplotted", pch = 17)

help ("plot")

seriemultiple10.12 <- window(seriemultiple, start=c(2010), end= c(2012))

seriemultiple12.14 <- window(seriemultiple, start=c(2012), end= c(2014))

plot(seriemultiple10.12, main ="Serie de tiempo", xlab="Años", ylab = "Número de personas", col = "blue",

type= "overplotted", pch = 17)

plot(seriemultiple12.14, main ="Serie de tiempo", xlab="Años", ylab = "Número de personas", col = "purple",

type= "overplotted", pch = 17)

########################### Tipo de cambio ############################

# Del 01 de Febrero de 2016 al 31 de enero de 2017

TipoDeCambio <- data.frame(read.csv("C://Users//SALA-D32//Desktop//SerieTiempoTipoCambio.csv"))

USD <- ts(TipoDeCambio[,1], frequency = 262, start = c(2016,32))

end(USD)

EUR <- ts(TipoDeCambio[,2], frequency = 262, start = c(2016,32))

end(EUR)

GBP <- ts(TipoDeCambio[,3], frequency = 262, start = c(2016,32))

end(GBP)

CAD <- ts(TipoDeCambio[,4], frequency = 262, start = c(2016,32))

end(CAD)

plot(cbind(USD))

plot(cbind(EUR))

plot(cbind(GBP))

plot(cbind(CAD))

SMTipoCambio <- ts.intersect(USD,EUR, GBP, CAD)

plot(SMTipoCambio, main ="Serie de tiempo", xlab="Años", ylab = "Número de personas", col = "red",

type= "overplotted", pch = 17)

SMTipoCambio1 <- window(SMTipoCambio, start=c(2016,32), end= c(2016,118))#86

SMTipoCambio2 <- window(SMTipoCambio, start=c(2016,119), end= c(2016,206))#87

SMTipoCambio3 <- window(SMTipoCambio, start=c(2016,207), end= c(2017,31))#86

plot(SMTipoCambio1, main ="Serie de tiempo", xlab="Años", ylab = "Número de personas", col = "blue",

type= "overplotted", pch = 17)

plot(SMTipoCambio2, main ="Serie de tiempo", xlab="Años", ylab = "Número de personas", col = "green",

type= "overplotted", pch = 17)

plot(SMTipoCambio3, main ="Serie de tiempo", xlab="Años", ylab = "Número de personas", col = "grey",

type= "overplotted", pch = 17)

plot(SMTipoCambio3, main ="Serie de tiempo", xlab="Años", ylab = "Número de personas", col = "grey",

type= "overplotted", pch = 17)

plot(SMTipoCambio3, main ="Serie de tiempo", xlab="Años", ylab = "Número de personas", col = c("red","blue","yellow","green")[DF$SMTipoCambio3],

type= "overplotted", pch = 17)

############################ 15 de febrero 2017 #########################################

Para generar un modelo de una ST, la serie se tiene que expresar como una función con 3 elementos:

m\_t -> Tendencia -> (Efecto tendencia)

S\_t -> Estacionalidad -> (Efecto estacional)

e\_t -> Error -> (Efecto aleatoriedad)

Los modelos que mas se utilizan con el aditivo y multiplicativo

################ Modelo aditivo ################

x\_t = m\_t + S\_t + e\_t

################ Modelo multiplicativo ################

Puro -> m\_t \* S\_t \* e\_t

Mixto -> m\_t \* S\_t + e\_t

La elección del modelo que se va a utilizar para la ST varía dependiendo de las

caractersticas que se observan en la gráfica.

Por ejemplo: Si tenemos una ST estacional con tendencia se aplica el modelo aditivo

############################ 17 de febrero 2017 #########################################

desoc <- sample(3:8,44,replace = T)

tdesoc <- ts(desoc, frequency = 4,start = 2005)

x11()

plot(decompose(tdesoc)) #Descompone la serie de tiempo en sus componenetes

#1.- Datos observados

#2.- Tendencia (suaviza la grafica)

#3.- Estacionalidad

#4.- Aleatoriedad/Error

plot(tdesoc)

# en este modelo tenemos tendencia y estacionalidad no constante,, el ideal es el modelo multiplicativo

tiie <- read.csv("C://Users//SALA-D32//Desktop//baseT.csv")

STtiie <- ts(tiie,frequency = 52,start = 2005)

plot(STtiie)

plot(decompose(STtiie)) #No tiene estacionalidad y tendencia

pib <- read.csv("C://Users//SALA-D32//Desktop//basePIB.csv")

TSpib <- ts(pib,frequency = 4,start = c(2007,1))

plot(TSpib)

plot(decompose(TSpib)) # Tenemos tendencia, estacionalidad y aleatoriedad

summary(TSpib)

depib <- decompose(TSpib)

depib

names(depib) #Elementos de una serie de tiempo!!

depib$trend #Datos asociados a la tendencia

depib$seasonal

desoc <- read.csv("C://Users//SALA-D32//Desktop//baseDesocup.csv")

tsdesoc <- ts(desoc,frequency = 4,start=2005)

plot(TSdesoc)

plot(decompose(TSdesoc))

#La Tasa de desocupacion aumenta notoriamente en el 2008 debido a la crisis de los creditos suprime (hipotecas),

#que afecto gravemente al PIB nacional y al tipo de cambio (devaluación del peso frente al dólar),

#provocando un aumento en la tasa de desocupacion, y se establece en el segundo trimestre de 2011

#y sigue teniendo aleatoriedad.

decomposedesoc <- decompose(TSdesoc)

names(decomposedesoc)

decomposedesoc$seasonal

#Muestra estacionalidad en todo el intervalo

decomposedesoc$trend

#Incremento importante en el 2008, de 2011 a 2015 se mantiene y después decrementa

############################ 22 de febrero 2017 #########################################

IVE

des <- read.csv("/Users/rodrigomondragondillanes/Documents/Series de tiempo/tarea2.csv")

tsdes <- ts(des,frequency = 4,start = 2005)

a <- rep(0,45)

for(i in 1:45){

q<-3+i

a[i] <- mean(des[i:q,1])

}

a

descomponer <- decompose(tsdes)

descomponer$trend

b <- rep(0,44)

for(i in 1:44){

b[i] <- (a[i]+a[i+1])/2

}

b #TENDENCIA

c <- rep(0,44)

for(i in 1:44){ # MODELO ADITIVO

d <- 2+i

c[i]<-des[d,1]-b[i]

}

c

#c <- rep(0,44)

#for(i in 1:44){ # MODELO MULTIPLICATIVO

# d <- 2+i

# c[i]<-des[d,1]/b[i]

#}

#c

dd <- c(NA,NA)

prueba <- c(dd,c,dd)

matriz <- matrix(prueba,nrow = 4)

sum(matriz[1,],na.rm = T)

IVE <- apply(matriz,1,mean,na.rm=T) # INDICES DE VARIACIONES ESTACIONALES

IVE

descomponer$seasonal

matriz

############################ 24 de febrero 2017 #########################################

u1 <- (1:10)

for(i in 1:10) {

u1[i]<- u1[i]+2

print(u1[i])

}

u1

s <- 7

while(s<6) {s <- s\*2; print (s);}

s <-1

while (s<5) {s <-s+1; ; if(s == 4) break; print(s);}

s <- 1

while (s<5) {s <-0+1; ; if(s == 2) break; print(s);}

# Corre todos los valores porque no encuentra el 2

s <- 2

while (s<3) {s <- s+1; ; if(s ==3) break; print(s);}

des <- read.csv("C://Users//SALA-D32//Documents//desocupacion.csv")

tsdes <- ts(des,frequency = 4,start = 2005)

detsdes <- decompose (tsdes)

detsdes$trend

detsdes$seasonal

plot(tsdes, col="blue", main="Tasa de desocupacion trimestral", ylab="Tasas", xlab="Años")

lines (detsdes$trend, col="red", ylab="Tendencia")

monthplot(detsdes$seasonal, main="IVEc",ylab="IVEc trimestral")

#Tarea para el lunes

#Encontrar una ST con tendencia a la alza y que la estacionalidad tambien se incremente

#Aplicar el moldeo multiplicativo y graficar tendencia

############################ 27 de febrero 2017 #########################################

#### Descomposicion clasica

# se utiliza la funcion decompose

# otro tipo de descomposicion de la aleatoriedad, tendencia y estacionalidad es la descomposicion por Loess, la cual esta basada

#en un algoritmo que permite descomponer las ST en 3 elementos: aleatoriedad, tendencia y estacionalida, una de las ventajas de

#esta descomposicion es el calculo de la estacionalidad que se obtienen valores de estacionalidad no constantes, en contrario

#con el metodo clasico de descomposicion

## DESCOMPOSICION CLASICA

desocupacion <- read.csv("C://Users//SALA-D//Desktop//tarea2.csv")

des <- ts(desocupacion,frequency = 4,start = 2005)

desco <- decompose(des)

tend1 <- desco$trend

x11();plot(tend1,col="blue")

## DESCOMPOSICION POR LOOSE

# se usa la funcion stl(), entonces para descomponer por Loose utilizamos

des <- ts(desocupacion[1:48,1],frequency = 4,start = 2005) #Refedino mi serie siendo mas especifico al marcar los datos

desco2 <- stl(des,s.window = "periodic",robust = T)

names(desco2)

desco2 #Nos muestra la tendencia, estacionalidad y leatoriedad/error

lines(desco2$time.series[,2],col="red") #Obtenemos la tendencia de nuestro objeto, que esta en la segunda columna

esta2 <- desco2$time.series[,1] #Estacionalidad

#install.packages("fpp")

require(fpp)

ajus <- seasadj(desco2) #Ajuste de la estacionalidad, para que el pronostico tenga mayor certidumbre

## PRONOSTICO METODO INGENUO

inge1 <- naive(ajus) #Funcion que pronostica, h es otro argumento donde le indicamos el numero de pronostico, buscar funcion naive#El momento de picos y caídas es predecible con los datos estacionales

#pero impredecible a largo plazo con datos cíclicos

install.packages("fpp")

require(fpp)

plot(window(elec,start=1980), ylab="Gwh", xlab = "year",main = "Australian electricity production")

#ESTACIONARIEDAD,TENDENCIA,CICLICO

plot(bricksq, ylab="million units", xlab="year", main="Australian clay brick production")

#ESTACIONARIEDAD Y CICLICO, NO TENDENCIA

plot(hsales, main="Sales of new one-family houses, USA", ylab="Total sales", xlab="year")

#NO ESTACIONARIEDAD, NI TENDENCIA, POSIBLE CICLO

seasonplot(elec, s=12, year.labels= TRUE, main="Electricidad", ylab = "Año", col = rainbow(20),year.labels.left=TRUE,pch=15)

seasonplot(bricksq, s=12, year.labels= F, main="Sales of new one-family houses", ylab = "Año", col = rainbow(20),year.labels.left=TRUE,pch=15)

seasonplot(hsales, s=12, year.labels= F, main="Valor acciones ICA", ylab = "Año", col = rainbow(20),year.labels.left=TRUE,pch=15)

dec<-decompose(bricksq)

sinesta<-bricksq-dec$seasonal

sintend<-bricksq-dec$trend

plot(sinesta)

plot(sintend)

x11();plot(inge1) #pronostica una constante con limites de confianza de 80 y 95

inge1

## EJERCICIO PIB

pib <- read.csv("C://Users//SALA-D//Desktop//PIB México.csv")

tspib <- ts(pib[1:36,1],frequency = 4,start = 2007)

end(tspib)

x11();plot(tspib)

clasica <- decompose(tspib)

lines(clasica$trend,main="Tendencia del PIB",xlab="anio",col="red")

loose <- stl(tspib,s.window = "periodic",robust = T)

loose

lines(loose$time.series[,2],col="blue")

##Podemos notar que alguno de los dos metodos tiene mas picos que otra, debemos escoger la descomposicion que este

#mas suavizada

############################ 06 marzo 2017 #########################################

#El momento de picos y caídas es predecible con los datos estacionales

#pero impredecible a largo plazo con datos cíclicos

install.packages("fpp")

require(fpp)

plot(window(elec,start=1980), ylab="Gwh", xlab = "year",main = "Australian electricity production")

#ESTACIONARIEDAD,TENDENCIA,CICLICO

plot(bricksq, ylab="million units", xlab="year", main="Australian clay brick production")

#ESTACIONARIEDAD Y CICLICO, NO TENDENCIA

plot(hsales, main="Sales of new one-family houses, USA", ylab="Total sales", xlab="year")

#NO ESTACIONARIEDAD, NI TENDENCIA, POSIBLE CICLO

seasonplot(elec, s=12, year.labels= TRUE, main="Electricidad", ylab = "Año", col = rainbow(20),year.labels.left=TRUE,pch=15)

seasonplot(bricksq, s=12, year.labels= F, main="Sales of new one-family houses", ylab = "Año", col = rainbow(20),year.labels.left=TRUE,pch=15)

seasonplot(hsales, s=12, year.labels= F, main="Valor acciones ICA", ylab = "Año", col = rainbow(20),year.labels.left=TRUE,pch=15)

dec<-decompose(bricksq)

sinesta<-bricksq-dec$seasonal

sintend<-bricksq-dec$trend

plot(sinesta)

plot(sintend)

############################ 08 marzo 2017 #########################################

## Descomponer en modelo aditivo

desoc <- data.frame(read.csv("Ruta"))

desocTS <- ts(desoc, frequency = 4, start = 2005)

plot (desocTS)

### Añadir al final

lines (eeadj, col = "darkcyan")

###

dectTSD <- descompose(desocTS)

dectTSDMul <- descompose(desocTS, type = "mult")

# Graficamos los 2 modelos

plot (dectTSD)

plot (dectTSDMul)

## Desestacionalizar una serie

# Se desestacionaliza una serie para poder predecir los datos con mayor

# precisión y sin contar con factores externos al comportamiento de nuestros

# datos, dos elementos que permiten desentralizar una serie

require(fpp)

eeadj <- seasonadj (dectTSD)

eeadj

plot (eeadj)

SinEstacionalidad <- desocTS - dectTSD

SinEstacionalidad

## Para comprobar...

eeadj - SinEstacionalidad

### Debe dar una matriz de ceros.

############################ 22 marzo 2017 #########################################

desocupacion <- read.cvs("")

pib <- rear.cvs("")

pibTS <- ts(pib)

desocupTS <- ts(desocupacion)

acf(pibTS) #Graficar el correlograma

acf(desocupTS)

################# MÈTODO INGENUO ###################

# (se basa en la ùltima observaciòn)

install.packages ("forecast")

require(forecast)

# naive(serie de tiempo, h = nùmero de pronòsticos)

naive(pibTS, h = 4)

# Hi -> intervalos de confianza

################# MÈTODO PROMEDIO ##################

# (Todas las observaciones tienen la misma importancia, el pronòstico es un promedio de las observaicones)

meanf(desocupTS, h = 4)

plot (meanf(desocupTS, h = 4))

# Esto sirve para la imputaciòn

################# MÈTODO INGENUO ESTACIONAL ##################

# Se utiliza en series de tiempo con alta estacionalidad, se le da mayor peso/ponderaciòn\_

# al dato del mismo periodo en el año anterior.

snaive (desocupTS, h = 4)

asa <- read.csv("C://Users//SALA-D32//Downloads//IndicadoresENOE\_2017-03-27.csv")

asa1 <- t( data.frame

(read.csv("C://Users//SALA-D32//Downloads//IndicadoresENOE\_2017-03-27.csv",

header = T)[7, 2:49]))

asa <- ts(asa1)

asa <- as.numeric(asa)

asat <- ts(asa1, start=2005, frequency = 4)

############################ 27 marzo 2017 #########################################

####################### Modelo suavizado exponencial en R #######################

install.packages("forecast")

require (forecast)

mod1 <- ses (asat, alpha=.1, intitial ="simple", h=8)

mod2 <- ses (asat, alpha=.3, intitial ="simple", h=8)

mod3 <- ses (asat, alpha=.95, h=8) # el pronóstico que mejor se ajusta a los datos observados

x11()

plot (mod1, ylab="Asalariados", xlab = "Años", main= "Asalariados", type ="o")

lines(mod1$fitted, col="brown", type = "o")

lines(mod2$fitted, col="red", type = "o")

lines(mod3$fitted, col="green", type = "o")

lines(mod1$mean, col="brown", type = "o")

lines(mod2$mean, col="red", type = "o")

lines(mod3$mean, col="green", type = "o")

leggend ("topleft", lty=1, col=c(1, "blue", "red","green"),

c("datos", expression(alpha==0.1), expression(alpha==.3),

expression(alpha==.95), pch=64)