tareasssssssexposicion.R

INTEL

2024-06-25

# Librer?as  
library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(forecast)

## Warning: package 'forecast' was built under R version 4.3.3

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(tseries)  
library(TSA)

## Warning: package 'TSA' was built under R version 4.3.3

## Registered S3 methods overwritten by 'TSA':  
## method from   
## fitted.Arima forecast  
## plot.Arima forecast

##   
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':  
##   
## acf, arima

## The following object is masked from 'package:utils':  
##   
## tar

library(urca)

## Warning: package 'urca' was built under R version 4.3.3

library(lmtest)

## Warning: package 'lmtest' was built under R version 4.3.3

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(nortest)  
library(mFilter)

## Warning: package 'mFilter' was built under R version 4.3.3

library(strucchange)

## Warning: package 'strucchange' was built under R version 4.3.3

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.3.3

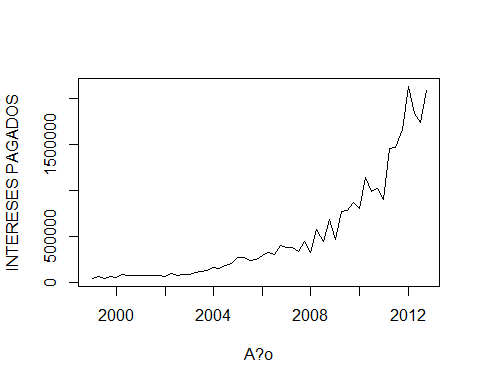
library(fitdistrplus)

## Warning: package 'fitdistrplus' was built under R version 4.3.3

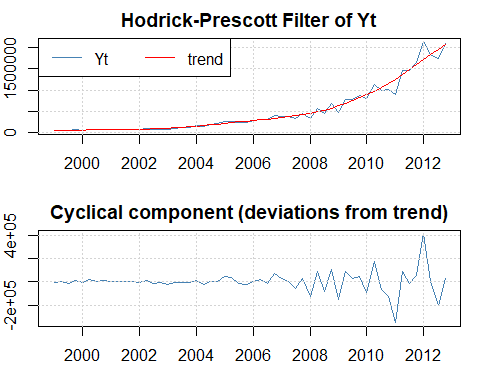
## Loading required package: survival

## Warning: package 'survival' was built under R version 4.3.3

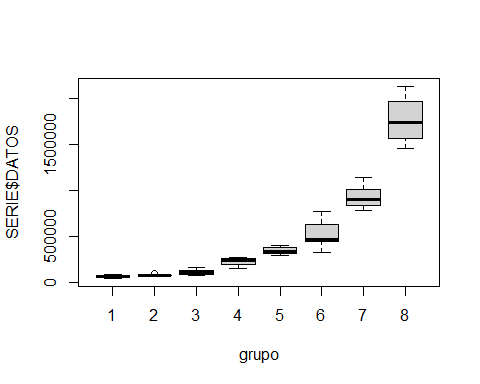
library(readxl)  
  
SERIE <- read\_excel("F:\\777--Programacion repos\\Una\\r\\jesus-task\\TAREA3\_\_2.xlsx")  
   
  
# IDENTIFICACION  
# Grafica de la serie  
Yt <- ts(SERIE$DATOS, start = 1999, frequency = 4)  
plot(Yt, xlab = "A?o", ylab = "INTERESES PAGADOS")



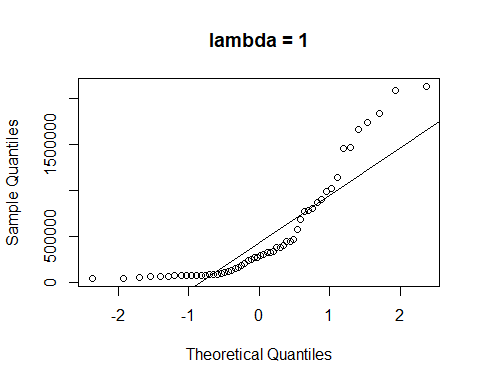
# UTILIZAREMOS EL FILTRO DE HODRIK- PRESCOT  
lambda\_hp = 100 # para datos anuales  
Yt\_hp <- hpfilter(Yt, type = "lambda", freq = lambda\_hp)  
plot(Yt\_hp)



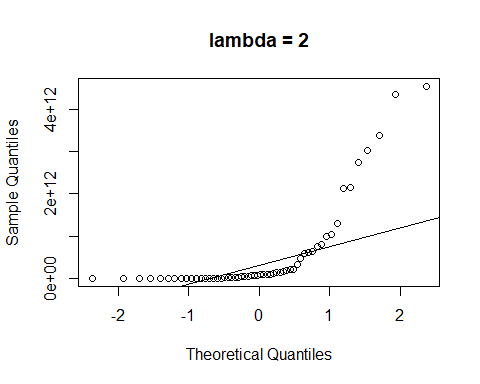
##########################################################  
# estacionariedad en varianza  
###########################################################  
Grupo <- rep(1:8, each = 7)  
boxplot(SERIE$DATOS~Grupo, xlab= "grupo", Ylab="Yt")



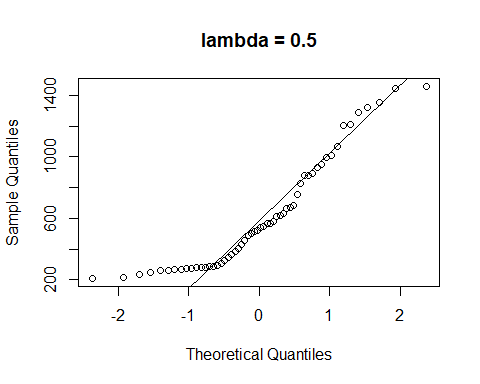
#CORRECCION DE LA ESTACIONARIEDAD EN VARIANZA  
  
# 1) para lambda = 1 <- Yt  
qqnorm(Yt, main ="lambda = 1")   
qqline(Yt)



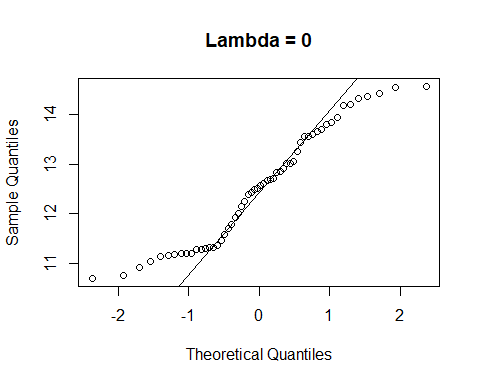
# 2) para lambda = 2 <- Yt^2  
T1.Yt <- Yt^2  
qqnorm(T1.Yt, main ="lambda = 2")   
qqline(T1.Yt)



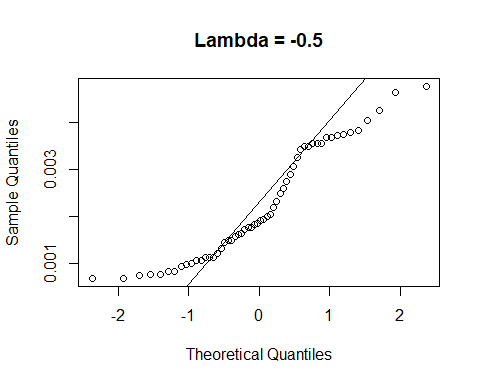
# 3) para lambda = 0.5 <- RAIZ 2  
T3.Yt <- sqrt(Yt)  
qqnorm(T3.Yt, main ="lambda = 0.5")   
qqline(T3.Yt)



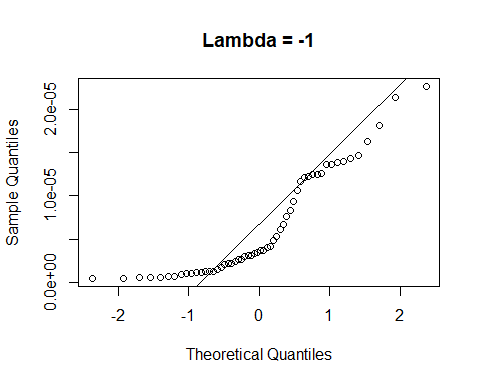
# 4) para lambda = 0  
T4.Yt <- log(Yt)  
qqnorm(T4.Yt, main = "Lambda = 0")  
qqline(T4.Yt)



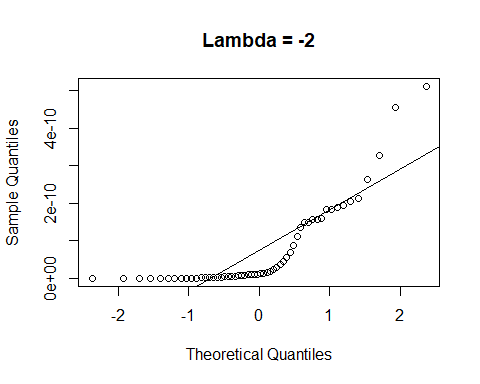
# 5) Para lambda = -0.5  
T5.Yt <- 1/sqrt(Yt)  
qqnorm(T5.Yt, main = "Lambda = -0.5")  
qqline(T5.Yt)



# 6)Para lambda = -1  
T6.Yt <- 1/Yt  
qqnorm(T6.Yt, main = "Lambda = -1")  
qqline(T6.Yt)



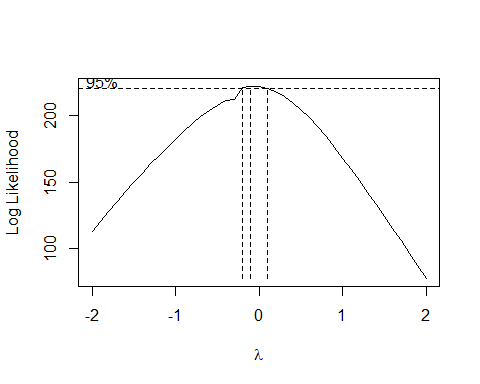
# 7) Para lambda = -2  
T7.Yt <- 1/(Yt^2)  
qqnorm(T7.Yt, main = "Lambda = -2")  
qqline(T7.Yt)



# determinacion de valor de lambda  
b <- BoxCox.ar(Yt)

## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1

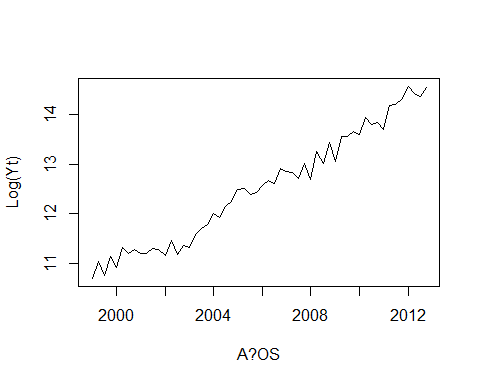
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1  
  
## Warning in arima0(x, order = c(i, 0L, 0L), include.mean = demean): possible  
## convergence problem: optim gave code = 1



# lambda optimo  
lambda <- b$mle  
round(lambda,1)

## [1] -0.1

# serie transfomormada T.Yt =   
T.Yt <- log(Yt)  
plot(T.Yt, xlab = "A?OS", ylab="Log(Yt)")



########################################  
#ESTACIONARIEDAD EN MEDIA  
#######################################  
# CORREOLOGRAMA  
FAS <- acf(T.Yt, lag.max = 15, main = "FAS - T.Yt", level = 0.05)

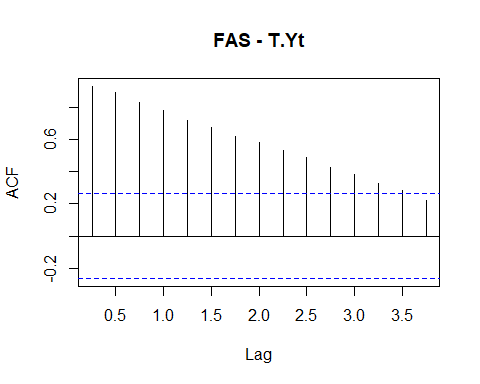
## Warning in plot.window(...): "level" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "level" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter

## Warning in box(...): "level" is not a graphical parameter

## Warning in title(...): "level" is not a graphical parameter



# MA()  
  
FAP <- pacf(T.Yt, lag.max = 15, main = "FAP - T.Yt", level = 0.05) # tiene que ser menor a 9.0

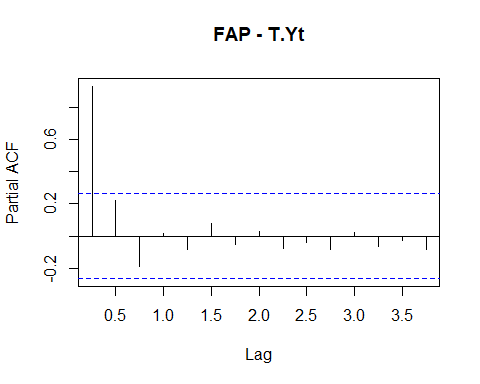
## Warning in plot.window(...): "level" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "level" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter

## Warning in box(...): "level" is not a graphical parameter

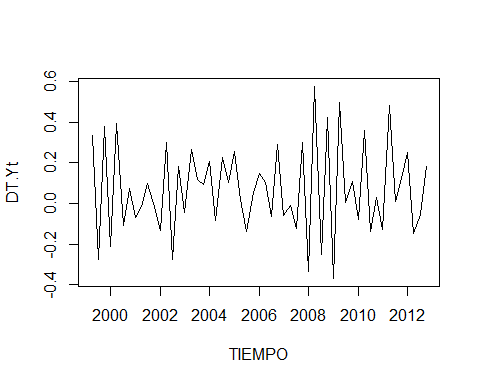
## Warning in title(...): "level" is not a graphical parameter



#AR()  
  
# PRUEBA DE RAIZ UNITARIA DIKEN FULER  
TY\_dfa <- ur.df(T.Yt, type = "trend", lags = 1)  
summary(TY\_dfa)

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression trend   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.28694 -0.11716 0.01847 0.11901 0.24365   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.520903 1.379493 2.552 0.0138 \*   
## z.lag.1 -0.327874 0.131403 -2.495 0.0159 \*   
## tt 0.024011 0.009248 2.596 0.0123 \*   
## z.diff.lag -0.547654 0.112264 -4.878 1.13e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.148 on 50 degrees of freedom  
## Multiple R-squared: 0.5748, Adjusted R-squared: 0.5493   
## F-statistic: 22.53 on 3 and 50 DF, p-value: 2.279e-09  
##   
##   
## Value of test-statistic is: -2.4952 12.0028 3.5166   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau3 -4.04 -3.45 -3.15  
## phi2 6.50 4.88 4.16  
## phi3 8.73 6.49 5.47

# SI ES NO ESTACIONARIA DIFERENCIAMOS LA SERIE  
DT.Yt <- diff(T.Yt)  
plot(DT.Yt, xlab = "TIEMPO")



# verificando la estacionalidad   
FAS <- acf(DT.Yt, lag.max = 15, main = "FAS D1Y", level = 0.95)

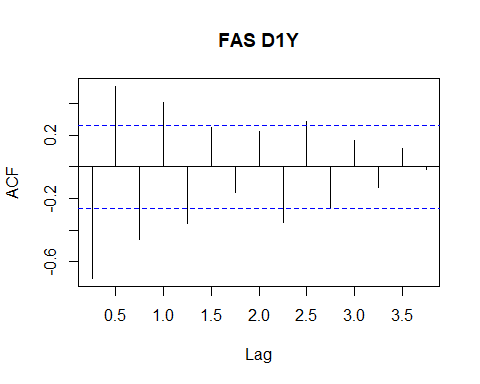
## Warning in plot.window(...): "level" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "level" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter

## Warning in box(...): "level" is not a graphical parameter

## Warning in title(...): "level" is not a graphical parameter



FAP <-pacf(DT.Yt, lag.max = 15, main = "FAP D1Y", level = 0.95)

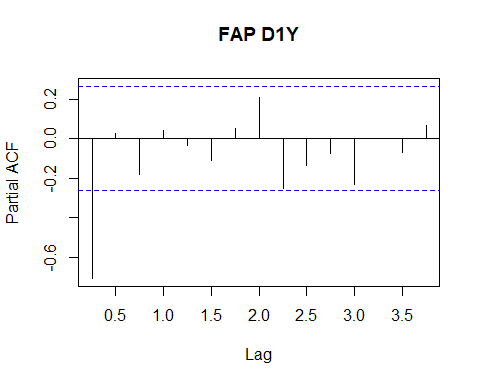
## Warning in plot.window(...): "level" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "level" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter

## Warning in box(...): "level" is not a graphical parameter

## Warning in title(...): "level" is not a graphical parameter



# PRUEBA DE RAIZ UNITARIA   
Ydfa1 <- ur.df(DT.Yt,type = "drift", lags = 1)  
summary(Ydfa1)

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.27334 -0.12772 0.02768 0.11931 0.28580   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.10946 0.02751 3.979 0.000224 \*\*\*  
## z.lag.1 -1.60644 0.25779 -6.232 9.54e-08 \*\*\*  
## z.diff.lag -0.05061 0.13839 -0.366 0.716119   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1564 on 50 degrees of freedom  
## Multiple R-squared: 0.8575, Adjusted R-squared: 0.8518   
## F-statistic: 150.5 on 2 and 50 DF, p-value: < 2.2e-16  
##   
##   
## Value of test-statistic is: -6.2316 19.4227   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.51 -2.89 -2.58  
## phi1 6.70 4.71 3.86

# ar(1) <- ARIMA(1,1,0)  
# ma(1) <- ARIMA(0,1,1)  
  
# Modelos ARIMA  
modelo1 <- Arima(T.Yt, order = c(1, 1, 0), include.constant = TRUE)  
coeftest(modelo1)

##   
## z test of coefficients:  
##   
## Estimate Std. Error z value Pr(>|z|)   
## ar1 -0.715918 0.092892 -7.7070 1.288e-14 \*\*\*  
## drift 0.067188 0.012105 5.5505 2.849e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

modelo2 <- Arima(T.Yt, order = c(0, 1, 1), include.constant = TRUE)  
coeftest(modelo2)

##   
## z test of coefficients:  
##   
## Estimate Std. Error z value Pr(>|z|)   
## ma1 -0.6238314 0.0953391 -6.5433 6.018e-11 \*\*\*  
## drift 0.0684654 0.0089266 7.6698 1.722e-14 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

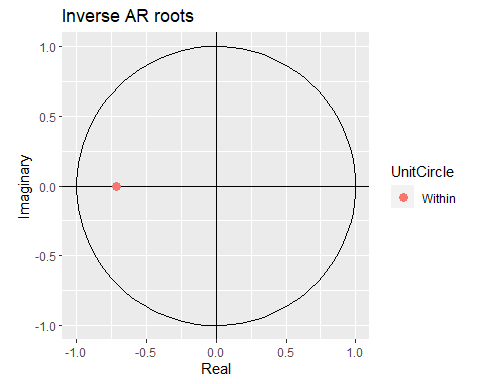
# Validaci?n de modelos  
vcov(modelo1)

## ar1 drift  
## ar1 8.628880e-03 2.049485e-05  
## drift 2.049485e-05 1.465264e-04

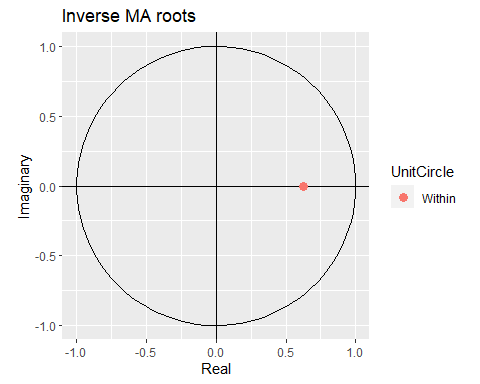
vcov(modelo2)

## ma1 drift  
## ma1 9.089551e-03 -1.703775e-06  
## drift -1.703775e-06 7.968381e-05

mod1 <- Arima(T.Yt, order = c(1, 1, 0), include.constant = TRUE)  
autoplot(mod1)



mod2 <- Arima(T.Yt, order = c(0, 1, 1), include.constant = TRUE)  
autoplot(mod2)



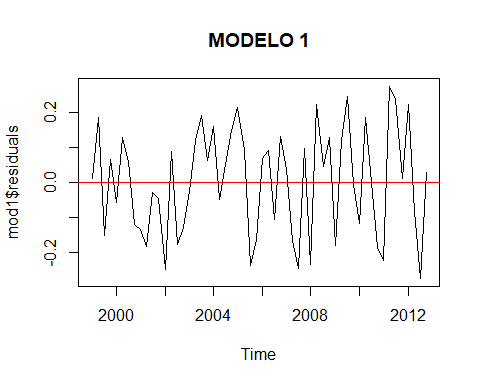
# Test de Chow  
Chow\_mod1 <- Fstats(mod1$fitted ~ 1, from = 0.43)  
sctest(Chow\_mod1)

##   
## supF test  
##   
## data: Chow\_mod1  
## sup.F = 168.3, p-value < 2.2e-16

Chow\_mod2 <- Fstats(mod2$fitted ~ 1, from = 0.43)  
sctest(Chow\_mod2)

##   
## supF test  
##   
## data: Chow\_mod2  
## sup.F = 173.75, p-value < 2.2e-16

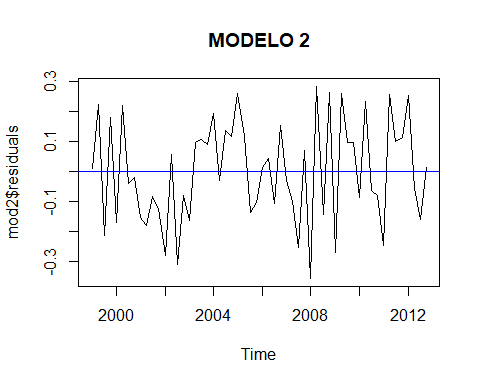
# An?lisis de los residuos  
plot(mod1$residuals, main = "MODELO 1")  
abline(h = 0, col = "red")



t.test(mod1$residuals, mu = 0)

##   
## One Sample t-test  
##   
## data: mod1$residuals  
## t = 0.10559, df = 55, p-value = 0.9163  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.03876991 0.04308281  
## sample estimates:  
## mean of x   
## 0.002156447

plot(mod2$residuals, main = "MODELO 2")  
abline(h = 0, col = "blue")



t.test(mod2$residuals, mu = 0)

##   
## One Sample t-test  
##   
## data: mod2$residuals  
## t = 0.074744, df = 55, p-value = 0.9407  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.04397660 0.04738405  
## sample estimates:  
## mean of x   
## 0.001703725

# Varianza constante de residuos  
scatter.smooth(sqrt(abs(mod1$residuals)), tpars = list(col ~ 2), main = "MODELO 1")

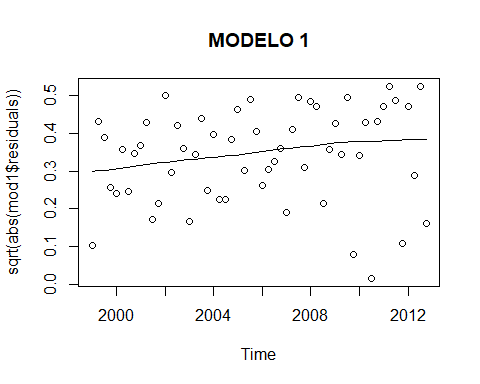
## Warning in plot.window(...): "tpars" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "tpars" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tpars" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "tpars" is not a  
## graphical parameter

## Warning in box(...): "tpars" is not a graphical parameter

## Warning in title(...): "tpars" is not a graphical parameter



scatter.smooth(sqrt(abs(mod2$residuals)), tpars = list(col ~ 2), main = "MODELO 2")

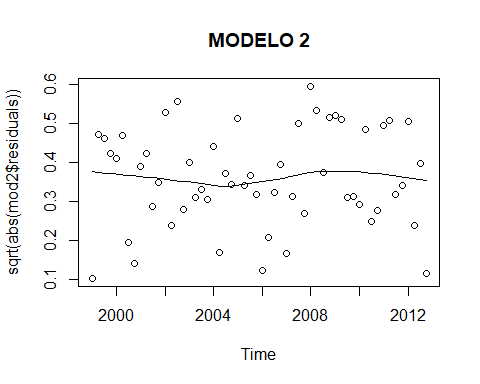
## Warning in plot.window(...): "tpars" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "tpars" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tpars" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "tpars" is not a  
## graphical parameter

## Warning in box(...): "tpars" is not a graphical parameter

## Warning in title(...): "tpars" is not a graphical parameter



obs = get(mod1$series)  
bptest(resid(mod1) ~ I(obs - resid(mod1)))

##   
## studentized Breusch-Pagan test  
##   
## data: resid(mod1) ~ I(obs - resid(mod1))  
## BP = 4.9887, df = 1, p-value = 0.02551

obs = get(mod2$series)  
bptest(resid(mod2) ~ I(obs - resid(mod2)))

##   
## studentized Breusch-Pagan test  
##   
## data: resid(mod2) ~ I(obs - resid(mod2))  
## BP = 0.16589, df = 1, p-value = 0.6838

# Ausencia de correlaci?n serial  
resid\_m1 <- as.vector(mod1$residuals)  
resid\_m2 <- as.vector(mod2$residuals)  
  
  
FAS\_e.m1 <- acf(resid\_m1, lag.max = 25, main = "FAS Modelo 1", level = 0.95)

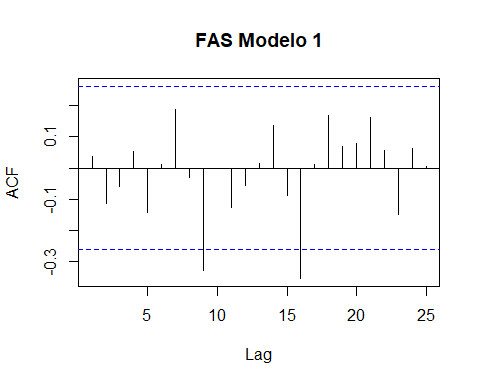
## Warning in plot.window(...): "level" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "level" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter

## Warning in box(...): "level" is not a graphical parameter

## Warning in title(...): "level" is not a graphical parameter



FAS\_e.m2 <- acf(resid\_m2, lag.max = 25, main = "FAS Modelo 2", level = 0.95)

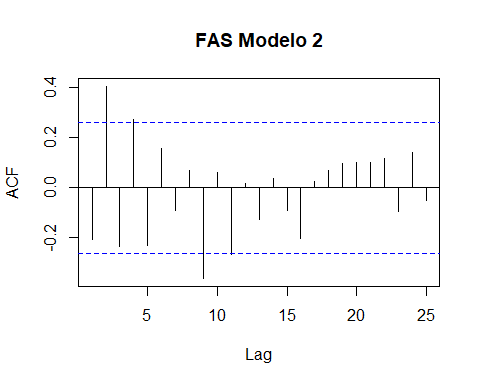
## Warning in plot.window(...): "level" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "level" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "level" is not a  
## graphical parameter

## Warning in box(...): "level" is not a graphical parameter

## Warning in title(...): "level" is not a graphical parameter



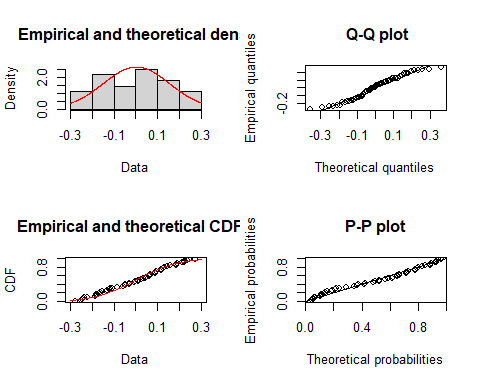
Box.test(resid\_m1, lag = 3, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: resid\_m1  
## X-squared = 1.0782, df = 3, p-value = 0.7823

Box.test(resid\_m2, lag = 3, type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: resid\_m2  
## X-squared = 15.77, df = 3, p-value = 0.001264

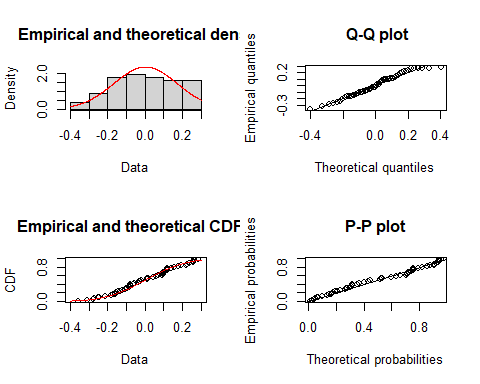
# Distribuci?n de residuos  
library(fitdistrplus)  
ajuste\_m1<-fitdist(data = resid\_m1, distr="norm")  
plot(ajuste\_m1)



JB\_m1 <- jarque.bera.test(resid\_m1)#Prueba de normalidad de Jarque-bera  
JB\_m1

##   
## Jarque Bera Test  
##   
## data: resid\_m1  
## X-squared = 3.0068, df = 2, p-value = 0.2224

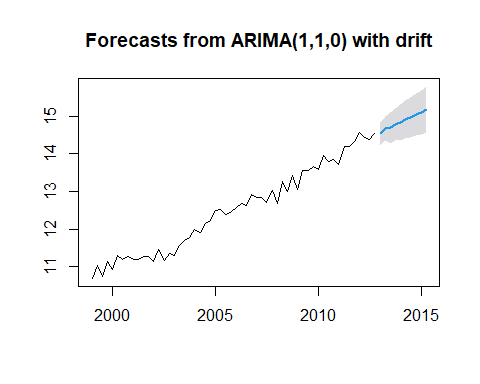
ajuste\_m2<-fitdist(data = resid\_m2, distr="norm")  
plot(ajuste\_m2)



JB\_m2 <- jarque.bera.test(resid\_m2)  
JB\_m2

##   
## Jarque Bera Test  
##   
## data: resid\_m2  
## X-squared = 2.1316, df = 2, p-value = 0.3445

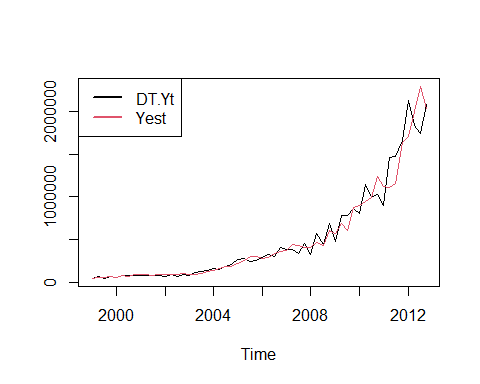
#########################3  
# PRONOSTICO  
#########################  
Pron1 <- forecast(mod1, level = c(95), h=10)  
plot(Pron1)



summary(Pron1)

##   
## Forecast method: ARIMA(1,1,0) with drift  
##   
## Model Information:  
## Series: T.Yt   
## ARIMA(1,1,0) with drift   
##   
## Coefficients:  
## ar1 drift  
## -0.7159 0.0672  
## s.e. 0.0929 0.0121  
##   
## sigma^2 = 0.02424: log likelihood = 24.91  
## AIC=-43.82 AICc=-43.35 BIC=-37.8  
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.002156447 0.1514681 0.1302622 -0.001002706 1.037708 0.4450654  
## ACF1  
## Training set 0.03672105  
##   
## Forecasts:  
## Point Forecast Lo 95 Hi 95  
## 2013 Q1 14.53611 14.23095 14.84127  
## 2013 Q2 14.66134 14.34410 14.97857  
## 2013 Q3 14.68697 14.28731 15.08664  
## 2013 Q4 14.78391 14.36328 15.20453  
## 2014 Q1 14.82980 14.35909 15.30051  
## 2014 Q2 14.91223 14.41701 15.40746  
## 2014 Q3 14.96851 14.43628 15.50074  
## 2014 Q4 15.04351 14.48612 15.60090  
## 2015 Q1 15.10510 14.51730 15.69290  
## 2015 Q2 15.17630 14.56397 15.78862

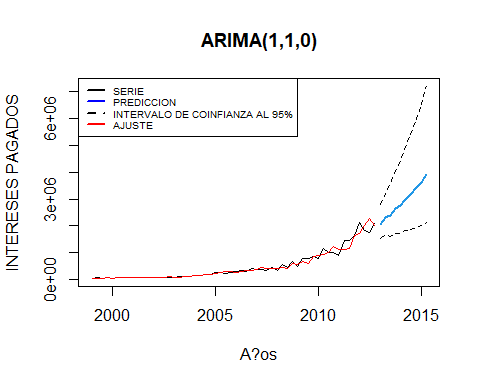
Yt\_arima1 <- exp(mod1$fitted)  
grafico\_comparativo <- cbind(Yt, Yt\_arima1)  
ts.plot(grafico\_comparativo, col = c(1:2), lwd = 1)  
legend("topleft", c("DT.Yt", "Yest"), lty = c(1,1), col = c(1:2), lwd = 2)



Pron1$mean <- exp(Pron1$mean)  
Pron1$lower <- exp(Pron1$lower)  
  
Pron1$upper <- exp(Pron1$upper)  
Pron1$x <- exp(Pron1$x)  
Pron1$fitted <- exp(Pron1$fitted)  
Pron1$residuals <- exp(Pron1$residuals)  
summary(Pron1)

##   
## Forecast method: ARIMA(1,1,0) with drift  
##   
## Model Information:  
## Series: T.Yt   
## ARIMA(1,1,0) with drift   
##   
## Coefficients:  
## ar1 drift  
## -0.7159 0.0672  
## s.e. 0.0929 0.0121  
##   
## sigma^2 = 0.02424: log likelihood = 24.91  
## AIC=-43.82 AICc=-43.35 BIC=-37.8  
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 6211.225 132113.9 71919.14 -0.9399903 13.13878 0.4836564 0.0547565  
##   
## Forecasts:  
## Point Forecast Lo 95 Hi 95  
## 2013 Q1 2055669 1515042 2789213  
## 2013 Q2 2329903 1696545 3199708  
## 2013 Q3 2390408 1602876 3564876  
## 2013 Q4 2633723 1729399 4010930  
## 2014 Q1 2757406 1722164 4414961  
## 2014 Q2 2994341 1824848 4913330  
## 2014 Q3 3167672 1860355 5393673  
## 2014 Q4 3414391 1955433 5961884  
## 2015 Q1 3631306 2017363 6536445  
## 2015 Q2 3899254 2113747 7192999

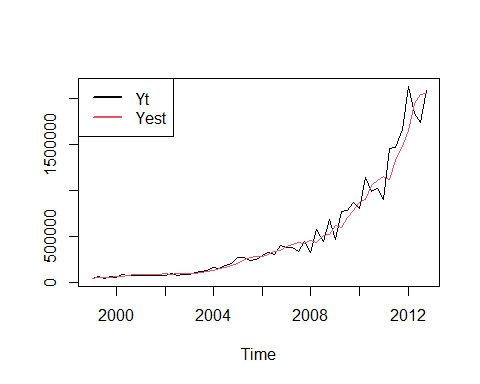
plot(Pron1, shaded = FALSE, xlab = "A?os", ylab = "INTERESES PAGADOS", main = "ARIMA(1,1,0)")  
lines(Pron1$fitted, col = "red")  
legend("topleft", legend = c("SERIE", "PREDICCION", "INTERVALO DE COINFIANZA AL 95%", "AJUSTE"),  
 col = c("black", "blue", "black", "red"), lty = c(1, 1, 2, 1), lwd = 2, cex = 0.6)  
abline(v = 1930, lwd = 1, col = "green")



#########################################################################################3  
# Modelo 2  
Pron2 <- forecast(mod2, level = c(95), h = 10)  
summary(Pron2)

##   
## Forecast method: ARIMA(0,1,1) with drift  
##   
## Model Information:  
## Series: T.Yt   
## ARIMA(0,1,1) with drift   
##   
## Coefficients:  
## ma1 drift  
## -0.6238 0.0685  
## s.e. 0.0953 0.0089  
##   
## sigma^2 = 0.0302: log likelihood = 18.98  
## AIC=-31.97 AICc=-31.5 BIC=-25.94  
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.001703725 0.1690539 0.1447522 -0.01583613 1.163839 0.4945729  
## ACF1  
## Training set -0.2064637  
##   
## Forecasts:  
## Point Forecast Lo 95 Hi 95  
## 2013 Q1 14.61009 14.26951 14.95068  
## 2013 Q2 14.67856 14.31467 15.04245  
## 2013 Q3 14.74702 14.36124 15.13281  
## 2013 Q4 14.81549 14.40899 15.22199  
## 2014 Q1 14.88395 14.45774 15.31017  
## 2014 Q2 14.95242 14.50737 15.39747  
## 2014 Q3 15.02089 14.55776 15.48401  
## 2014 Q4 15.08935 14.60883 15.56987  
## 2015 Q1 15.15782 14.66051 15.65512  
## 2015 Q2 15.22628 14.71274 15.73983

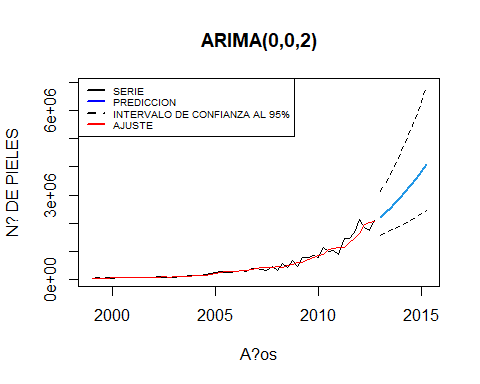
Yt\_arima2 <- exp(mod2$fitted)  
grafico\_comparativo <- cbind(Yt, Yt\_arima2)  
ts.plot(grafico\_comparativo, col = c(1:2), lwd = 1)  
legend("topleft", c("Yt", "Yest"), lty = c(1, 1), col = c(1:2), lwd = 2)



Pron2$mean <- exp(Pron2$mean)  
Pron2$lower <- exp(Pron2$lower)  
Pron2$upper <- exp(Pron2$upper)  
Pron2$x <- exp(Pron2$x)  
Pron2$fitted <- exp(Pron2$fitted)  
Pron2$residuals <- mod2$residuals   
summary(Pron2)

##   
## Forecast method: ARIMA(0,1,1) with drift  
##   
## Model Information:  
## Series: T.Yt   
## ARIMA(0,1,1) with drift   
##   
## Coefficients:  
## ma1 drift  
## -0.6238 0.0685  
## s.e. 0.0953 0.0089  
##   
## sigma^2 = 0.0302: log likelihood = 18.98  
## AIC=-31.97 AICc=-31.5 BIC=-25.94  
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 13998.35 119019.8 71411.71 -1.269675 14.58315 0.4802439  
## ACF1  
## Training set -0.06014883  
##   
## Forecasts:  
## Point Forecast Lo 95 Hi 95  
## 2013 Q1 2213517 1574590 3111704  
## 2013 Q2 2370375 1647337 3410763  
## 2013 Q3 2538348 1725868 3733316  
## 2013 Q4 2718224 1810273 4081563  
## 2014 Q1 2910848 1900717 4457809  
## 2014 Q2 3117121 1997421 4864493  
## 2014 Q3 3338011 2100653 5304215  
## 2014 Q4 3574554 2210722 5779758  
## 2015 Q1 3827860 2327974 6294106  
## 2015 Q2 4099116 2452788 6850470

plot(Pron2, shaded = FALSE, xlab = "A?os", ylab = "N? DE PIELES", main = "ARIMA(0,0,2)")  
lines(Pron2$fitted, col = "red")  
legend("topleft", legend = c("SERIE", "PREDICCION", "INTERVALO DE CONFIANZA AL 95%", "AJUSTE"),  
 col = c("black", "blue", "black", "red"), lty = c(1, 1, 2, 1), lwd = 2, cex = 0.6)  
abline(v = 1930, lwd = 1, col = "green")



#Error medio absoluto escalado (MASE - Mean Absolute Scaled Error)  
accuracy(Pron1)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 6211.225 132113.9 71919.14 -0.9399903 13.13878 0.4836564 0.0547565

accuracy(Pron2)

## ME RMSE MAE MPE MAPE MASE  
## Training set 13998.35 119019.8 71411.71 -1.269675 14.58315 0.4802439  
## ACF1  
## Training set -0.06014883