

Apéndice A

Código R

```
##ANÁLISIS INICIAL

#Definimos los valores de la variable 'nivelT' como un objeto de R de tipo serie de
  tiempo para R. La serie empieza el día 332 del 2004 y finaliza el día 99 de 2012.
  La serie tiene una frecuencia de 365 ya que se recogen datos diarios desde 2004 a
  2012
nivel<-ts(NivelTudela$nivelT, start=c(2004,322),end=c(2012,99),frequency=365)
nivel

# comprobación de que los datos coinciden y se ha definido bien la serie de tiempo
nivel [1]
nivel[2698]

#Dibujamos la serie
plot(nivel, xlab="Años", ylab="Nivel en m del rio", main="Nivel de río Ebro en
  Tudela")
#Separamos en años
abline(v=c(2004:2005),lty=3,col=4)
abline(v=c(2005:2006),lty=3,col=4)
abline(v=c(2006:2007),lty=3,col=4)
abline(v=c(2007:2008),lty=3,col=4)
abline(v=c(2008:2009),lty=3,col=4)
abline(v=c(2009:2010),lty=3,col=4)
abline(v=c(2010:2011),lty=3,col=4)
abline(v=c(2011:2012),lty=3,col=4)

##CONVERSIÓN DE LA SERIE A ESTACIONARIA

#Buscamos el mejor método para estabilizar la varianza
library(car)
help("boxCox")
boxCox(lm(nivel~1),lambda = seq(-3, -1,0.1)) #lambda=-2
#Transformación para estabilizar la varianza
lnivel<-1/(nivel)^2
plot(lnivel, xlab="Años", ylab="Nivel en m del rio", main="Serie transformada")

#Diferenciación de la serie para eliminar la estacionalidad
niveldif<-diff(lnivel,lag=365)
plot(niveldif)
```

```

#Contraste de hipótesis serie estacionaria
library(tseries)
library(forecast)
kpss.test(niveldif) #p-value=0.01 no pasa el test

#Diferenciamos una vez para eliminar la tendencia
niveldifdif<-diff(niveldif)
plot(niveldifdif)

#Contraste de hipótesis serie estacionaria
kpss.test(niveldifdif) #p-value=0.1 no se rechaza que la serie sea estacionaria

#Renombramos nuestra serie estacionaria
serie<-niveldifdif
plot(serie)

## SELECCION PRELIMINAR DEL MODELO

#Selección de los rangos de p y q

pacf(serie,na.action=na.pass) #14
acf(serie,na.action=na.pass) #6

#Modelo ARMA

for (i in c(0:14))
{
  for (j in c(0:6))
  {
    serieARMA<-Arima(serie, order=c(i,0,j))
    cat('AIC del modelo: ',i,j,serieARMA$aic,fill=TRUE)
  }
}

#Menor AIC ARMA(6,5)
ARMA65<-Arima(serie, order=c(6,0,5))
ARMA65
ARMA65$coef
ARMA65$sigma2
ARMA65$var.coef
ARMA65$loglik
ARMA65$aic

##VALIDACIÓN DEL MODELO
#Cálculo de los residuos
residuos<-ARMA65$resid[is.na(ARMA65$resid)==F]
var(residuos)

# No tienen varianza 1, los transformamos para que sea así
resstan<-residuos/(sum(residuos**2)/length(residuos))**0.5
var(resstan) #varianza de los residuos estandarizados; sale muy próxima a 1

# Ver si los residuos son de media nula y varianza constante

```

```

plot(as.ts(resstan)) #no hay indicio claro de que la serie no sea estacionaria
abline(h=0, col='green',lty=1)
abline(h=2, col='blue',lty=2)
abline(h=-2, col='blue',lty=2)

# Ver si los residuos son incorrelados

#ACF de la serie esiduos
acf(resstan)

#Test de Ljung-Box
Box.test(resstan,type='Ljung')

#Estas pruebas se aplica a veces a los residuos de un ARMA(p,q),en cuyo caso
# las referencias sugieren fitdf = p+q,por supuesto siempre que lag > fitdf.

for (i in c(1:12))
{
a<-Box.test(resstan,type='Ljung',lag=i) #valor sin ajustar los grados de libertad
print(a$p.value)
a<-Box.test(resstan,type='Ljung',lag=i,fitdf=4)
print(a$p.value)

cat(fill=T)
}
tsdiag(ARMA65)

##PREDICCIÓN
# Predicciones de la serie estacionaria

predARMA65<-predict(ARMA65,30)
predARMA65
plot(forecast(ARMA65,30))

#Predicción con estimación componente estacional por armónicos

predic2<- Arima(nivel, order=c(6,1,5), xreg=fourier(nivel, K=1), lambda=-2)
pre<-forecast(predic2, h=30, xreg=fourierf(nivel, K=1, h=30))
pre
plot(forecast(predic2, h=30, xreg=fourierf(nivel, K=1, h=30)))

```
