

Anexo VIII

Código de FitARMA

```
# Librerías auxiliares
library(ggplot2)
library(zoo)
library(forecast)
library(tseries)

# Instalamos y cargamos FitARMA
install.packages("FitARMA")
library(FitARMA)

# Cargamos los datos desde la API de datamarket (necesaria conexión a internet)
library(rdatamarket)
accidentes <- as.ts(dmseries("http://data.is/1yFXOBi"))
# Cargamos los datos desde csv (en caso de problemas con internet o la API)
datos <- read.csv("monthly-traffic-fatalities-in-on.csv")
accidentes <- ts(datos, start = c(1960, 1), frequency = 12)

# Función genérica para plotting de test vs pred a través de ggplot2
combine <- function(test, pred) {
  require(ggplot2)
  p <- ggplot() +
    geom_line(aes(x = index(as.zoo(test)),
                  y = coredata(as.zoo(test)), colour = "Test")) +
    geom_line(aes(x = index(as.zoo(acc.test)),
                  y = pred, colour = "Prediccion")) +
    scale_color_manual(name = "", values = c("Test" = "black",
                                             "Prediccion" = "red"),
                      labels = c("Test", "Predicción")) +
    ylab("Número de accidentes de tráfico") +
    xlab("Año 1974") + scale_x_continuous(breaks = c(), labels = c())
  p
}

# Ajustamos estacionalmente la serie y volvemos a dividir en dos conjuntos
decomposition <- decompose(accidentes, type = "additive")
accidentes.adj <- seasadj(decomposition) # Forecast
acc.train.adj <- window(accidentes.adj, start = c(1960,1), end = c(1973,12))
acc.test.adj <- window(accidentes.adj, start = c(1974,1))

# Diferenciamos
accidentes.dif.adj <- diff(accidentes.adj)
acc.train.dif.adj <- window(accidentes.dif.adj,
                           start = c(1960,2), end = c(1973,12))
acc.test.dif.adj <- window(accidentes.dif.adj, start = c(1974,1))

# ACF y PACF
autoplot(Acf(acc.train.dif.adj))
autoplot(Pacf(acc.train.dif.adj))

# ARMA(2,1) (FitARMA)
model.1 <- FitARMA(acc.train.dif.adj, order = c(2,0,1))
summary(model.1) # AIC = 939.5 Loglikelihood = -465.75
coef(model.1)
model.1$racf
autoplot(Acf(residuals(model.1)))
model.1$LjungBoxQ
```

```

# ARMA(2,1) (FitARMA) (MeanMLEQ = TRUE)
model.2 <- FitARMA(acc.train.dif.adj, order = c(2,0,1), MeanMLEQ = TRUE)
summary(model.2) # AIC = 938.5 Loglikelihood = -465.27
coef(model.2)
autoplot(Acf(residuals(model.2)))
model.1$LjungBoxQ
mod <- ggplot() +
  geom_line(aes(x = index(as.zoo(acc.train.dif.adj)),
                y = coredata(as.zoo(acc.train.dif.adj)), colour = "Original")) +
  geom_line(aes(x = index(as.zoo(fitted(model.2))),
                y = coredata(as.zoo(fitted(model.2))), colour = "Ajustada")) +
  scale_color_manual(name = "", values = c("Original" = "black",
                                           "Ajustada" = "blue"),
                    labels = c("Ajustada", "Original")) +
  xlab("Año") + ylab("Primeras diferencias")
mod

# Ajustando pApprox
pApprox.metrics <- c()
pApprox.values <- seq(10, 80, 10)

for (i in pApprox.values) {
  model <- FitARMA(acc.train.dif.adj, order = c(2,0,1),
                  MeanMLEQ = TRUE, pApprox = i)
  pApprox.metrics <- c(pApprox.metrics, model$loglikelihood)
}
pApp <- data.frame(pApprox = pApprox.values,
                  LogLikelihood = pApprox.metrics)

ajuste.papprox <- ggplot() +
  geom_line(data = pApp, aes(x = pApprox, y = LogLikelihood)) +
  xlab("pApprox") + ylab("Logaritmo de la verosimilitud") +
  scale_x_continuous(breaks = pApprox.values)
ajuste.papprox

# ARMA(2,1) (FitARMA) (MeanMLEQ = TRUE)
model.3 <- FitARMA(acc.train.dif.adj, order = c(2,0,1),
                  MeanMLEQ = TRUE, pApprox = 80)
summary(model.3) # AIC = 937 Loglikelihood = -464.52
coef(model.3)
autoplot(Acf(residuals(model.2)))
model.1$LjungBoxQ

# Comparando tiempos para ARMAs
p <- 2; q <- 1
timing.forecast <- c()
timing.tseries <- c()
timing.fitarma <- c()

for (i in 1:50) {
  timing.fitarma <- c(timing.fitarma,
                    as.numeric(system.time(FitARMA(acc.train.dif.adj,
                                                    order = c(p,0,q)))[3]))

  timing.forecast <- c(timing.forecast,
                    as.numeric(system.time(Arima(acc.train.dif.adj,
                                                    order = c(p,0,q)))[3]))

  timing.tseries <- c(timing.tseries,

```

```

        as.numeric(system.time(arma(acc.train.dif.adj,
                                   order = c(p, q)))[3]))
}

timing.plot <- ggplot() +
  geom_line(aes(x = 1:length(timing.forecast),
                y = timing.forecast, colour = "Forecast")) +
  geom_line(aes(x = 1:length(timing.tseries),
                y = timing.tseries, colour = "TSeries")) +
  geom_line(aes(x = 1:length(timing.fitarma),
                y = timing.fitarma, colour = "FitARMA")) +
  xlab("Simulación") + ylab("Tiempo de ejecución en segundos") +
  scale_color_manual(name = "Leyenda", values = c("TSeries" = "red",
                                                  "Forecast" = "blue",
                                                  "FitARMA" = "green"),
                    labels = c("FitARMA", "Forecast", "TSeries"))

timing.plot

```