Anexo IX

Código de Opera

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

# Instalamos y cargamos Opera
install.packages("opera")
library(opera)

# 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)

# Training y test
acc.train <- window(accidentes, start = c(1960,1), end = c(1973,12))
acc.test <- window(accidentes, start = c(1974,1))

# 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(test)),
                  y = pred, colour = "Prediccion")) +
    scale_color_manual(name = "", values = c("Test" = "black",
                                             "Prediccion" = "red"),
                       labels = c("Test","Predicción")) +
    xlab("Año 1974") + ylab("Número de accidentes de tráfico") +
    scale_x_continuous(breaks = c(), labels = c())
  p
}

# Ajustamos los modelos a combinar
# Red neuronal
red <- nnetar(acc.train, repeats = 25, size = 20, decay = 9.5, p = 20, P = 4)
pred.red <- forecast(red, h = 12)
accuracy(pred.red, acc.test) # 13.2...
# SARIMA(1,0,0)(1,0,0)12
sarima <- auto.arima(acc.train, test = "adf")
pred.sarima <- forecast(sarima, h = 12)
accuracy(pred.sarima, acc.test) # 14.86938
# Damped HoltWinters Mutiplicativo
pred.hw <- hw(acc.train, h = 12, damped = TRUE, seasonal = "multiplicative",
              initial = "optimal")
accuracy(pred.hw, acc.test) # 15.04762

# Juntamos las predicciones
X <- cbind(red = pred.red$mean, sarima = pred.sarima$mean, hw = pred.hw$mean)

# Gráfico de las predicciones individuales de los modelos
```

```
p <- ggplot() +
  geom_line(aes(x = index(as.zoo(acc.test)),
                y = coredata(as.zoo(acc.test)), colour = "Original")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
                y = X[, 1], colour = "Red Neuronal")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
                y = X[, 2], colour = "SARIMA")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
                y = X[, 3], colour = "Holt-Winters")) +
  scale_color_manual(name = "", values = c("Original" = "black",
                                           "Red Neuronal" = "blue",
                                           "SARIMA" = "red",
                                           "Holt-Winters" = "green"),
                     labels = c("Holt-Winters","Original",
                                "Red Neuronal", "SARIMA")) +
  xlab("Año 1974") + ylab("Número de accidentes de\ntráfico") +
  scale_x_continuous(breaks = c(), labels = c())
p

# Loss
loss(X, as.vector(acc.test), loss.type = "absolute")

# Oracle (Convex y Linear) Offline
oracle.convex <- oracle(Y = acc.test, experts = X,
                        loss.type = "absolute", model = "convex")
plot(oracle.convex)
print(oracle.convex)
accuracy(as.vector(oracle.convex$prediction), acc.test) # 13.1...
combine(as.vector(oracle.convex$prediction), acc.test)

oracle.linear <- oracle(Y = acc.test, experts = X,
                        loss.type = "absolute", model = "linear")
plot(oracle.linear)
print(oracle.linear)
accuracy(as.vector(oracle.linear$prediction), acc.test) # 12.2...
combine(as.vector(oracle.linear$prediction), acc.test)

# Predicciones en línea
mix <- mixture(model = "EWA", loss.type = "absolute",
               coefficients = "Uniform")
model <- mix
for (i in 1:length(as.vector(acc.test))) {
  model <- predict(model, X[i,], as.vector(acc.test)[i])
}
print(model)
model$weights
accuracy(as.vector(model$prediction), acc.test) # 14.08114
combine(as.vector(model$prediction), acc.test)
plot(model, pause = TRUE)
```