Anexo III

Código de Forecast

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

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

# Instalando forecast desde el repositorio oficial en GitHub
#(se recomienda instalar del CRAN)
install.packages("githubinstall")
library(githubinstall)
# Necesario instalar RTools (en caso de no tenerlo se instala por defecto)
githubinstall("forecast") # robjhyndman/forecast  forecast package for R
library(forecast)

# 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))

# Plotting
plot(accidentes)

# Plotting con ggplot del training y el test
p <- ggplot() +
  geom_line(aes(x = index(as.zoo(acc.train)), y = coredata(as.zoo(acc.train)),
                colour = "Training")) +
  geom_line(aes(x = index(as.zoo(acc.test)), y = coredata(as.zoo(acc.test)),
                colour = "Test")) +
  scale_color_manual(name = "", values = c("Training" = "blue", "Test" = "red"),
                     labels = c("Test","Training")) +
  xlab("Año") + ylab("Número de accidentes de\ntráfico")
p

# 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
}

# Frecuencia estacional de nuestros datos
findfrequency(x = acc.train)
```

```
monthdays(x = acc.train)
seasonplot(x = acc.train, s = 12)
seasonplot(x = acc.train, s = findfrequency(acc.train))

# Descomposición de una serie temporal e independencia de los residuos (STATS PACKAGE)
acc.train.decomp <- decompose(x = acc.train, type = "multiplicative")
seasonal(object = acc.train.decomp)
trendcycle(object = acc.train.decomp)
remainder(object = acc.train.decomp)
Box.test(x = acc.train.decomp$random, type = "Ljung-Box") # Son ruido blanco

# Desestacionalización a través de objectos decompose
acc.train.seasadj <- seasadj(acc.train.decomp)
plot(acc.train.seasadj)
ndiffs(acc.train.seasadj, test = "kpss") # 1
ndiffs(acc.train.seasadj, test = "adf") # 0
ndiffs(acc.train.seasadj, test = "pp") # 0

# Visualizando la tendencia
tend <- ma(acc.train, order = 12)
autoplot(tend)

# Prediciendo nuevos valores con la media
model <- meanf(acc.train, h = 12)
autoplot(model, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
model$mean[1] == mean(acc.train) # TRUE
combine(model$mean, acc.test)
# Medimos la precisión del modelo
accuracy(model, acc.test) # 36.28770
# Revisamos los residuos
Box.test(model$residuals, lag = 12, type = "Ljung-Box")
checkresiduals(model) # No son ruido blanco

#  Modelo ingenuo en el que se predice con el valor anterior observado
model <- naive(acc.train, h = 12)
autoplot(model, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(model$mean, acc.test)
accuracy(model, acc.test) # 33.83333
checkresiduals(model) # No son ruido blanco

#  Modelo ingenuo en el que se predice con el valor anterior observado con constante != 0
model <- rwf(acc.train, h = 12, drift = TRUE)
autoplot(model, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(model$mean, acc.test)
accuracy(model, acc.test) # 33.27645
checkresiduals(model) # No son ruido blanco

# Modelo ingenuo que tiene en cuenta la estacionalidad
model <- snaive(acc.train, h = 12)
autoplot(model, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(model$mean, acc.test)
accuracy(model, acc.test) # 22.58333
checkresiduals(model) # No son ruido blanco

# Comparando gráficamente modelos ingenuos
model.mean <- meanf(acc.train, h = 12)
model.naive <- naive(acc.train, h = 12)
model.snaive <- snaive(acc.train, h = 12)
model.rwf <- rwf(acc.train, h = 12, drift = TRUE)
```

```
figura.11 <- ggplot() +
  geom_line(aes(x = index(as.zoo(acc.test)),
              y = coredata(as.zoo(acc.test)), colour = "Test")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
              y = model.mean$mean, colour = "Meanf")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
              y = model.naive$mean, colour = "Naive")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
              y = model.snaive$mean, colour = "Snaive")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
              y = model.rwf$mean, colour = "Rwf")) +
  scale_color_manual(name = "", values = c("Test" = "black",
                                          "Meanf" = "red",
                                          "Naive" = "blue",
                                          "Snaive" = "green",
                                          "Rwf" = "yellow"),
                     labels = c("Meanf","Naive", "Rwf", "Snaive", "Test")) +
  ylab("Número de accidentes de tráfico") + xlab("Año 1974") +
  scale_x_continuous(breaks = c(), labels = c())
figura.11

# Ajuste de un modelo lineal con tendencia y estacionalidad a los datos
model <- tslm(acc.train ~ trend + season)
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test)
accuracy(pred, acc.test) # 19.44597
checkresiduals(model) # No son ruido blanco

# Damped Holt-Winters Mutiplicativo
model <- hw(acc.train, h = 12, damped = TRUE, seasonal = "multiplicative",
            initial = "optimal")
summary(model)
autoplot(model, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(model$mean, acc.test)
accuracy(model, acc.test) # 15.04762
checkresiduals(model) # No son ruido blanco

# Damped Holt-Winters Aditivo
model <- hw(acc.train, h = 12, damped = TRUE, seasonal = "additive",
            initial = "optimal")
summary(model)
autoplot(model, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(model$mean, acc.test)
accuracy(model, acc.test) # 19.67722
checkresiduals(model) # No son ruido blanco

# Comparando modelos Holt-Winters
model.mult <- hw(acc.train, h = 12, damped = TRUE, seasonal = "multiplicative",
                 initial = "optimal")
model.adit <- hw(acc.train, h = 12, damped = TRUE, seasonal = "additive",
                 initial = "optimal")
ajuste <- ggplot() +
  geom_line(aes(x = index(as.zoo(acc.train)),
              y = coredata(as.zoo(acc.train)), colour = "Serie original")) +
  geom_line(aes(x = index(as.zoo(fitted(model.adit))),
              y = coredata(as.zoo(fitted(model.adit))), colour = "H-W Aditivo")) +
  geom_line(aes(x = index(as.zoo(fitted(model.mult))),
              y = coredata(as.zoo(fitted(model.mult))), colour = "H-W Multiplicativo")) +
  scale_color_manual(name = "", values = c("Serie original" = "black",
                                          "H-W Multiplicativo" = "red",
                                          "H-W Aditivo" = "green"),
                     labels = c("H-W Aditivo","H-W Multiplicativo","Serie original")) +
  xlab("Año") + ylab("Número de accidentes de\ntráfico")
ajuste

predicciones <- ggplot() +
  geom_line(aes(x = index(as.zoo(acc.test)), y = coredata(as.zoo(acc.test)),
                colour = "Serie original")) +
  geom_line(aes(x = index(as.zoo(acc.test)), y = model.adit$mean,
                colour = "H-W Aditivo")) +
  geom_line(aes(x = index(as.zoo(acc.test)), y = model.mult$mean,
                colour = "H-W Multiplicativo")) +
  scale_color_manual(name = "", values = c("Serie original" = "black",
                                          "H-W Multiplicativo" = "red",
                                          "H-W Aditivo" = "green"),
                     labels = c("H-W Aditivo","H-W Multiplicativo","Serie original")) +
  xlab("Año") + ylab("Número de accidentes de tráfico") +
  scale_x_continuous(breaks = c(), labels = c())
predicciones

# Red Neuronal por defecto
# (Las medidas de precisión pueden variar debido al caracter aleatorio de las redes neuronales)
model <- nnetar(acc.train)
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año",
         ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test)
accuracy(pred, acc.test) # [22-28]

# Estudio del parámetro de regularización (decay)
install.packages("Metrics")
library(Metrics)
metrics.decay <- c()
for (i in seq(0, 10, 0.1)) {
  model <- nnetar(acc.train, decay = i)
  pred <- forecast(model, h = 12)
  metrics.decay <- c(metrics.decay, mae(pred$mean, acc.test))
}
m.decay <- data.frame(decay = seq(0, 10, 0.1), MAE = metrics.decay)

decay <- ggplot() +
  geom_line(data = m.decay, aes(x = decay, y = MAE)) +
  xlab("Parámetro de regularización (Decay)") + ylab("MAE")
decay

# Red Neuronal personalizada 1
model <- nnetar(acc.train, decay = 9.5)
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año",
         ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test)
accuracy(pred, acc.test) # 14....

# Estudio general de las redes (tarda en ejecutar)
metrics.repeats <- c()
for (i in seq(0, 1000, 100)[-1]) {
  model <- nnetar(acc.train, repeats = i)
  pred <- forecast(model, h = 12)
  metrics.repeats <- c(metrics.repeats, mae(pred$mean, acc.test))
```

```
}
m.repeats <- data.frame(repeats = seq(0, 1000, 100)[-1],
                        MAE = metrics.repeats)

repeats <- ggplot() +
  geom_line(data = m.repeats, aes(x = repeats, y = MAE)) +
  xlab("Número de modelos estimados") + ylab("MAE") +
  scale_x_continuous(breaks = seq(0, 1000, 100)[-1])
repeats

metrics.size <- c()
for (i in 1:10) {
  model <- nnetar(acc.train, size = i)
  pred <- forecast(model, h = 12)
  metrics.size <- c(metrics.size, mae(pred$mean, acc.test))
}
m.size <- data.frame(size = 1:10, MAE = metrics.size)

size <- ggplot() +
  geom_line(data = m.size, aes(x = size, y = MAE)) +
  xlab("Número de neuronas en la capa oculta") + ylab("MAE") +
  scale_x_continuous(breaks = 1:10)
size

metrics.p <- c()
for (i in 1:43) {
  model <- nnetar(acc.train, decay = 9.5, p = i)
  pred <- forecast(model, h = 12)
  metrics.p <- c(metrics.p, mae(pred$mean, acc.test))
}
m.p <- data.frame(p = 1:43, MAE = metrics.p)

p <- ggplot() +
  geom_line(data = m.p, aes(x = p, y = MAE)) +
  xlab("Parámetro p") + ylab("MAE")
p

metrics.PP <- c()
for (i in 1:10) {
  model <- nnetar(acc.train, decay = 9.5, P = i)
  pred <- forecast(model, h = 12)
  metrics.PP <- c(metrics.PP, mae(pred$mean, acc.test))
}
m.PP <- data.frame(P = 1:10, MAE = metrics.PP)

PP <- ggplot() +
  geom_line(data = m.PP, aes(x = P, y = MAE)) +
  xlab("Parámetro P") + ylab("MAE") +
  scale_x_continuous(breaks = 1:10)
PP

# Red Neuronal personalizada 2
model <- nnetar(acc.train, repeats = 25, size = 20, decay = 9.5, p = 20, P = 4)
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año",
         ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test)
accuracy(pred, acc.test) # 13....

# Técnicas ARIMA
```

```
autoplot(acc.train, xlab = "Año",
         ylab = "Número de accidentes de tráfico", main = "")
# Ajustamos la varianza con log y con Box Cox
autoplot(log(acc.train), xlab = "Año",
         ylab = "Log del número de accidentes\nde tráfico", main = "")
acc.train.trans <- BoxCox(acc.train, lambda = BoxCox.lambda(acc.train))
autoplot(acc.train.trans, xlab = "Año",
         ylab = "Transformación Box-Cox de\naccidentes de tráfico",
         main = "")
p <- ggplot() +
  geom_line(aes(x = index(as.zoo(acc.train)),
                y = coredata(as.zoo(log(acc.train))),
                colour = "Training")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
                y = coredata(as.zoo(log(acc.test))),
                colour = "Test")) +
  scale_color_manual(name = "", values = c("Training" = "blue", "Test" = "red"),
                     labels = c("Test","Training")) +
  xlab("Año") + ylab(" Log del número de accidentes de\ntráfico")
p

# ¿Nuestra serie es estacionaria?
ndiffs(acc.train) # 1
nsdiffs(acc.train) # 0

# Diferenciamos la serie (no estacionalmente)
autoplot(diff(acc.train), xlab = "Año", ylab = "Primeras diferencias", main = "")
autoplot(Acf(diff(acc.train))) # MA -> 0 | 2 (12)
autoplot(Pacf(diff(acc.train))) # AR -> 0 | 0 (12)

# Ajustamos un SARIMA(0,1,0)(0,0,2)12
model <- Arima(acc.train, order = c(0,1,0), seasonal = c(0,0,2))
summary(model) # AIC = 1550.38
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test)
accuracy(pred, acc.test) # MAE = 22.19949
checkresiduals(model) # No son ruido blanco

# Diferenciamos estacionalmente (se pierde la estacionalidad)
accidentes.diff12 <- diff(accidentes, 12) # perdemos un año de observaciones
acc.train.diff12 <- window(accidentes.diff12, start = c(1961,1), end = c(1973,12))
acc.test.diff12 <- window(accidentes.diff12, start = c(1974,1))

p <- ggplot() +
  geom_line(aes(x = index(as.zoo(acc.train.diff12)),
                y = coredata(as.zoo(acc.train.diff12)),
                colour = "Training")) +
  geom_line(aes(x = index(as.zoo(acc.test.diff12)),
                y = coredata(as.zoo(acc.test.diff12)),
                colour = "Test")) +
  scale_color_manual(name = "", values = c("Training" = "blue", "Test" = "red"),
                     labels = c("Test","Training")) +
  xlab("Año") + ylab("Primeras diferencias estacionales")
p

autoplot(Acf(acc.train.diff12)) # MA -> 1 | 1 (12)
autoplot(Pacf(acc.train.diff12)) # AR -> 1 | 2 (12)

# Ajustamos un SARIMA(1,0,1)(2,1,1)12 (Sin drift)
```

```
model <- Arima(acc.train, order = c(1,0,1), seasonal = c(2,1,1), include.drift = FALSE)
summary(model) # AIC = 1367.68
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test)
accuracy(pred, acc.test) # MAE = 19.22651
checkresiduals(model) # No son ruido blanco

# Ajustamos un SARIMA(1,0,1)(2,1,1)12 (Con drift)
model <- Arima(acc.train, order = c(1,0,1), seasonal = c(2,1,1), include.drift = TRUE)
summary(model) # AIC = 1359.82
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test)
accuracy(pred, acc.test) # MAE = 17.0999
checkresiduals(model) # Ruido blanco

# Selección automática del modelo (SARIMA(1,0,0)(1,0,0)12)
model <- auto.arima(acc.train, test = "adf")
summary(model) # AIC = 1514.22
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test[1:12])
accuracy(pred, acc.test) # MAE = 14.86938
checkresiduals(model) # No son ruido blanco

# Comparamos modelos con y sin drift
model.sin <- Arima(acc.train, order = c(1,0,1), seasonal = c(2,1,1), include.drift = FALSE)
pred.sin <- forecast(model.sin, h = 12)
model.con <- Arima(acc.train, order = c(1,0,1), seasonal = c(2,1,1), include.drift = TRUE)
pred.con <- forecast(model.con, h = 12)

drift <- ggplot() +
  geom_line(aes(x = index(as.zoo(acc.test)),
             y = coredata(as.zoo(acc.test)), colour = "Test")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
             y = pred.sin$mean, colour = "Sin intercepto")) +
  geom_line(aes(x = index(as.zoo(acc.test)),
             y = pred.con$mean, colour = "Con intercepto")) +
  scale_color_manual(name = "", values = c("Test" = "black", "Sin intercepto" = "red",
                                       "Con intercepto" = "green"),
                    labels = c("Con intercepto", "Sin intercepto", "Test")) +
  ylab("Número de accidentes de tráfico") +
  xlab("Año 1974") + scale_x_continuous(breaks = c(), labels = c())
drift

# Selección automática del modelo (sin ahorrar coste computacional) (SARIMA (2,0,0)(2,0,0)12)
model <- auto.arima(acc.train, test = "adf", max.order = 7,
                    stepwise = FALSE, approximation = FALSE)
summary(model) # AICc = 1499.24
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test)
accuracy(pred, acc.test) # MAE = 14.96979
checkresiduals(model) # No son ruido blanco

# Selección automática del modelo (sin ahorrar coste computacional) (SARIMA (1,1,4)(2,0,0)12)
model <- auto.arima(acc.train, max.order = 7,
                    stepwise = FALSE, approximation = FALSE)
summary(model) # AICc = 1485.29
```

```
pred <- forecast(model, h = 12)
autoplot(pred, xlab = "Año", ylab = "Número de accidentes de tráfico", main = "")
combine(pred$mean, acc.test)
accuracy(pred, acc.test) # MAE = 14.96979
checkresiduals(model) # No son ruido blanco
```