Modelización de regresión de series temporales con R.
XIV Summer School MESIO UPC-UB.

Práctica Sesiones 1 y 2: Conceptos básicos

Aurelio Tobías, Dominic Royé

En esta sesión práctica, primero vamos **gestionar archivos y depurar los datos** necesarios para examinar el efecto de la temperatura sobre la mortalidad. Realizaremos una **visualización de los datos de series temporales** a fin de entender adecuadamente su estructura temporal. Por último, utilizaremos **regresión de Possion** ajustando la estructura temporal de los datos. Lo haremos a modo de demostración, siguiendo unos pasos establecidos y ejecutando el código de **R** que se proporciona para cada paso. El alumno deberá contestar las preguntas que se plantean.

---

1. Manejo de datos y archivos

En primer lugar, vamos a cargar las **librerías y funciones** necesarias para el manejo de datos.

```
# librerias manejo de datos
library(tidyverse)
library(lubridate)
library(fs)
library(tsibble)
library(feasts)

# librerias series temporales
library(tsModel)
library(splines)
library(dlnm)

# cargar funciones R
source("QAICM.R")
source("findmin.R")
```

La datos que vamos a utilizar son series temporales de unidad diaria de la ciudad de **València** para el periodo **1994-2006**. Los archivos de datos en formato original contienen la mortalidad diaria **(mor_valencia.csv)**, variables meteorológicas **(meteo_valencia.csv)**, y contaminación atmosférica **(ozono_valencia.csv)**.

Importamos los datos en **R** y los fusionamos en un *data frame*.

```r
# importar y unir archivos de datos
data_valencia <- dir_ls("Datos", regexp = "csv$") %>%
                    map(read.csv2) %>%
                    reduce(left_join, by = "fecha")
```

A continuación, convertimos la variable **fecha** en el formato apropiado para fechas en **R** y comprobamos si hay **datos duplicados** para algún día o **lagunas** en los datos.

```r
# formato fecha
data_valencia <- mutate(data_valencia,
                  fecha = mdy(fecha)) %>%
                  as_tsibble(index = "fecha")

# duplicados
is_duplicated(data_valencia, index = fecha)

# lagunas
has_gaps(data_valencia)
```

Generamos nuevas **variables de calendario**: año, mes, día del mes, día de la semana, y tendencia.

```r
# generar variables de calendario
data_valencia <- mutate(data_valencia,
                        yr = as.factor(year(fecha)),
                        mo = as.factor(month(fecha, label = TRUE)),
                        dy = day(fecha),
                        wd = as.factor(wday(fecha, label = TRUE, abbr = FALSE)),
                        wk = if_else(wd %in% c("domingo", "sábado"), 1, 0),
                        trend = row_number())
```

Por último, **exportamos** los datos a un archivo **.csv* y los **guardamos** en formato **.RData*

```r
# exportar *.csv
write_csv2(data_valencia, "data_valencia_clean.csv")

# guardar *.RData
save(data_valencia, file = "data_valencia_clean.RData")
```

2. Visualización de datos de series temporales

Veamos el **gráfico temporal** de la mortalidad diaria por causas naturales **(rm)**.

```r
# grafico serie temporal
ggplot(data_valencia,
       aes(fecha, tm)) +
    geom_line(colour = 'blue') +
  scale_x_date(date_breaks = "2 years", date_labels = "%Y") +
```

```
  scale_y_continuous(breaks = seq(0, 50, 5)) +
  labs(x = "Fecha", y = "Num. muertes") +
  theme_minimal()
```

Utilizamos **diagramas de cajas** para describir la **tendencia** a largo plazo (años), **estacionalidad** (meses) y **variaciones a corto plazo** (día de la semana).

```
# box plot años
ggplot(data_valencia,
       aes(yr, tm))  +
  geom_boxplot()

# box plot meses
ggplot(data_valencia,
       aes(mo, tm))  +
  geom_boxplot()

# box plot dia de la semana
ggplot(data_valencia,
       aes(wd, tm))  +
  geom_boxplot()
```

Por último, evaluamos la **descomposición anual** de la mortalidad diaria.

```
# descomposicion
data_valencia %>%
  model(STL(tm ~ season(period = 365.25))) %>%
  components() %>%
  autoplot()
```

Pregunta 1. ¿Qué componentes de tendencia y estacionalidad observas en la mortalidad diaria por causas naturales?

Ahora vamos a representar la temperatura media diaria **(tmean)**.

```
# serie temporal
ggplot(data_valencia,
    aes(fecha, tmean)) +
    geom_line(colour = 'red') +
  scale_x_date(date_breaks = "2 years", date_labels = "%Y") +
  scale_y_continuous(breaks = seq(0, 35, 5)) +
  labs(x = "Fecha", y = "Temperatura (ºC)") +
  theme_minimal()
```

Utilizamos de nuevo **diagramas de cajas** para describir la tendencia a largo plazo (años), estacionalidad (meses), y variaciones a corto plazo (día de la semana).

```
# box plot años
ggplot(data_valencia,
```

```
       aes(yr, tmean))  +
  geom_boxplot()

# blox plot meses
ggplot(data_valencia,
       aes(mo, tmean))  +
  geom_boxplot()

# box plot dia de la semana
ggplot(data_valencia,
       aes(wd, tmean))  +
  geom_boxplot()
```

Pregunta 2. ¿Qué estructura temporal observas en la temperatura media diaria?

---

3. Ajuste de la estructura temporal

En primer lugar veamos el grado de **sobredispersión y autocorrelación** que tiene la serie de mortalidad diaria por causas naturales.

```
# modelo
model0 <- glm(tm ~ 1, data=data_valencia, family=quasipoisson)
summary(model0)

# autocorrelacion
acf(data_valencia$tm, lag=30)
```

Pregunta 3. ¿Cuanta sobredisperión tienen la mortalidad diaria y crees que se distribuye independientemente?

Empezamos el proceso de modelización temporal ajustando la variable año **(yr)** en el modelo de regresión.

```
# modelo
model1 <- glm(tm ~ factor(yr), data=data_valencia, family=quasipoisson)
summary(model1)

# prediccion
pred1 <- predict(model1, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred1, lwd=2, col="blue")
```

```
# autocorrelacion residual
res1 <- residuals(model1, type="response")
acf(res1, lag=30)
```

Pregunta 3. ¿Cómo interpretas los parámetros del modelo de regresión de Poisson?

Pregunta 4. ¿Presenta el modelo sobredispersión y autocorrelación residual?

Pregunta 5. ¿Crees que el año produce un buen ajuste de la estructura temporal de la mortalidad diaria?

A continuación, ajustamos por año y mes **(mo)**.

```
# modelo
model2 <- glm(tm ~ factor(yr) + factor(mo), data=data_valencia,
family=quasipoisson)
summary(model2)

# prediccion
pred2 <- predict(model2, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred2, lwd=2, col="blue")

# autocorrelacion residual
res2 <- residuals(model2, type="response")
acf(res2, lag=30)
```

Pregunta 6. ¿Ha mejorado el ajuste de la estructura temporal de la mortalidad diaria?

Ahora vamos a utilizar **funciones periódicas** para ajustar la tendencia y estacionalidad. Empezamos ajustando una **periodicidad anual** en el modelo de regresión.

```
# funcion periodica anual
fourier1 <- harmonic(data_valencia$fecha,nfreq=1,period=365.25)

# modelo
model3 <- glm(tm ~ trend + fourier1 , data=data_valencia, family=quasipoisson)
summary(model3)

# prediccion
pred3 <- predict(model3, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
```

```
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred3, lwd=2, col="seagreen")

# autocorrelacion residual
res3 <- residuals(model3, type="response")
acf(res3, lag=30)
```

A continuación, consideramos periodicidades **semestral, cuatrimestral y trimestal**.

```
# funcion periodica semestral
fourier2 <- harmonic(data_valencia$fecha,nfreq=2,period=365.25)

# modelo
model4 <- glm(tm ~ trend + fourier2 , data=data_valencia, family=quasipoisson)
summary(model4)

# prediccion
pred4 <- predict(model4, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred4, lwd=2, col="seagreen")

# autocorrelacion residual
res4 <- residuals(model4, type="response")
acf(res4, lag=30)
```

```
# funcion periodica cuatrimestral
fourier3 <- harmonic(data_valencia$fecha,nfreq=3,period=365.25)

# modelo
model5 <- glm(tm ~ trend + fourier3 , data=data_valencia, family=quasipoisson)
summary(model5)

# prediccion
pred5 <- predict(model5, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred5, lwd=2, col="seagreen")

# autocorrelacion residual
res5 <- residuals(model5, type="response")
acf(res5, lag=30)
```

```
# funcion periodica trimestral
fourier4 <- harmonic(data_valencia$fecha,nfreq=4,period=365.25)

# modelo
model6 <- glm(tm ~ trend + fourier4 , data=data_valencia, family=quasipoisson)
summary(model6)

# prediccion
pred6 <- predict(model6, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
```

```
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred6, lwd=2, col="seagreen")

# autocorrelacion residual
res6 <- residuals(model6, type="response")
acf(res6, lag=30)
```

Evaluamos la bondad de ajuste a través del Criterio de Información de Akaike **(AIC)** adaptado para regresión con **quasipoisson**.

```
# comparar modelos con AIC
QAICM(model3, "dev")
QAICM(model4, "dev")
QAICM(model5, "dev")
QAICM(model6, "dev")
```

Pregunta 7. ¿Qué modelo muestra una mejor bondad de ajuste (AIC), sobredispersión y autocorrelación residual?

Por último, utilizamos **funciones flexibles** (splines). Empezaremos utilizando **1 grado de libertad** por año.

```
# funcion flexible 1 df/ano
spl <- ns(data_valencia$trend,df=12)

# modelo
model7 <- glm(tm ~ spl , data=data_valencia, family=quasipoisson)
summary(model7)

# prediccion
pred7 <- predict(model7, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred7, lwd=2, col="red")

# autocorrelacion residual
res7<- residuals(model7, type="response")
acf(res7, lag=30)
```

A continuación utilizamos *3, 5 y 7 grados de libertad* por año.

```
# funcion flexible 3 df/ano
spl <- ns(data_valencia$trend, df=36)

# modelo
model8 <- glm(tm ~ spl , data=data_valencia, family=quasipoisson)
summary(model8)

# prediccion
pred8 <- predict(model8, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
```

```
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred8, lwd=2, col="red")

# autocorrelacion residual
res8 <- residuals(model8, type="response")
acf(res8, lag=30)
```

```
# funcion flexible 5 df/ano
spl <- ns(data_valencia$trend, df=60)

# modelo
model9 <- glm(tm ~ spl , data=data_valencia, family=quasipoisson)
summary(model9)

# prediccion
pred9 <- predict(model9, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred9, lwd=2, col="red")

# autocorrelacion residual
res9 <- residuals(model9, type="response")
acf(res9, lag=30)
```

```
# funcion flexible 7 df/ano
spl <- ns(data_valencia$trend, df=84)

# modelo
model10 <- glm(tm ~ spl , data=data_valencia, family=quasipoisson)
summary(model10)

# prediccion
pred10 <- predict(model10, type="response")
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred10, lwd=2, col="red"))

# autocorrelacion residual
res10 <- residuals(model10, type="response")
acf(res10, lag=30)
```

Evaluamos la bondad de ajuste a través del **AIC**.

```
# comparar modelos con AIC
QAICM(model7, "dev")
QAICM(model8, "dev")
QAICM(model9, "dev")
QAICM(model10, "dev")
```

Pregunta 8. ¿Cuantos grados de libertad crees más apropiados para ajustar la estructura temporal de la mortalidad diaria?

A continuación comparamos el ajuste de los modelos utilizando variables de calendario, funciones periódicas y funciones flexibles.

```
plot(data_valencia$fecha, data_valencia$tm, pch=19, cex=0.2, col=grey(0.6),
  ylab="Num. muertes", xlab="Fecha")
lines(data_valencia$fecha, pred2, lwd=2, col="blue")
lines(data_valencia$fecha, pred6, lwd=2, col="seagreen")
lines(data_valencia$fecha, pred10, lwd=2, col="red")
```

```
# comparar modelos con AIC
QAICM(model2, "dev")
QAICM(model3, "dev")
QAICM(model10, "dev")
```

Pregunta 9. ¿Qué modelo crees que ajusta mejor la estructura temporal de la mortalidad diaria?

---

4. Asociación entre temperatura y mortalidad

Una vez definido el **modelo basal** vamos a estudiar el efecto de la **temperatura**. Empezamos evaluando la **asociación lineal** entre temperatura y mortalidad.

```
# ajuste lienal de temperatura
btmean1 <- onebasis(data_valencia$tmean, df=1)
model.tmean1 <- glm(tm ~ spl + btmean1, data=data_valencia, family=quasipoisson)
min1 <- min(data_valencia$tmean)
pred1 <- crosspred(btmean1, model.tmean1, cen=min1)
plot(pred1, xlab="Temperatura (ºC)", ylab="Riesgo de mortalidad")
```

Ahora evaluamos una posible **asociación no lineal**. Utilizamos 2, 3 y 4 grados de libertad para la función flexible.

```
# ajuste no-lienal de temperatura con df=2
btmean2 <- onebasis(data_valencia$tmean, df=2)
model.tmean2 <- glm(tm ~ spl + btmean2, data=data_valencia, family=quasipoisson)
min2 <- findmin(btmean2, model.tmean2)
pred2 <- crosspred(btmean2, model.tmean2, cen=min2)
plot(pred2, xlab="Temperatura (ºC)", ylab="Riesgo de mortalidad")
```

```
# ajuste no-lienal de temperatura con df=3
btmean3 <- onebasis(data_valencia$tmean, df=3)
model.tmean3 <- glm(tm ~ spl + btmean3, data=data_valencia, family=quasipoisson)
min3 <- findmin(btmean3, model.tmean3)
pred3 <- crosspred(btmean3, model.tmean3, cen=min3)
plot(pred3, xlab="Temperatura (ºC)", ylab="Riesgo de mortalidad")
```

```r
# ajuste no-lienal de temperatura con df=4
btmean4 <- onebasis(data_valencia$tmean, df=4)
model.tmean4 <- glm(tm ~ spl + btmean4, data=data_valencia, family=quasipoisson)
min4 <- findmin(btmean4, model.tmean4)
pred4 <- crosspred(btmean4, model.tmean4, cen=min4)
plot(pred4, xlab="Temperatura (ºC)", ylab="Riesgo de mortalidad")
```

Vamos a comparar el ajuste de los modelos.

```r
# comparar modelos con AIC
QAICM(model.tmean1, "dev")
QAICM(model.tmean2, "dev")
QAICM(model.tmean3, "dev")
QAICM(model.tmean4, "dev")
```

Pregunta 10. ¿Qué **forma funcional**, y con cuantos **grados de libertad**, se define mejor la asociación entre temperatura y mortalidad?

Ahora vamos a evaluar la relación entre **ozono** y mortalidad, comparando la asociación lineal y no lineal (con 2 grados de libertad).

```r
# ajuste lienal de ozono con df=1
bozono1 <- onebasis(data_valencia$o38, df=1)
model.ozono1 <- glm(tm ~ spl + bozono1, data=data_valencia, family=quasipoisson)
min <- min(data_valencia$o38, na.rm=TRUE)
pred <- crosspred(bozono1, model.ozono1, cen=min)
plot(pred, xlab="Ozono (8h)", ylab="Riesgo de mortalidad")
```

```r
# Ajuste no lineal de ozono con df=2
bozono2 <- onebasis(data_valencia$o38, df=2)
model.ozono2 <- glm(tm ~ spl + bozono2, data=data_valencia, family=quasipoisson)
pred <- crosspred(bozono2, model.ozono2, cen=min)
plot(pred, xlab="Ozono (8h)", ylab="Riesgo de mortalidad")
```

```r
# comparar modelos con AIC
QAICM(model.ozono1, "dev")
QAICM(model.ozono2, "dev")
```

Pregunta 11. ¿Qué **forma funcional** resume mejor la relación entre ozono y mortalidad?

Pregunta 12. ¿Qué **limitaciones** tiene el análisis que hemos realizado para evaluar las asociaciones entre temperatura, ozono y mortalidad diaria?
