eAsia Summer School. Time-series regression in Public Health.

Practical 1. Introduction to the Time Series Design.

In this practical, students will exercise some explanatory analyses and data visualizations for daily time-series data. Next, students will learn basic steps of the time-series regression analysis using R. Finally, we will shortly introduce how to explore exposure-response relationship between the outcome variable and a given environmental exposure. Students will encounter some given questions. Please try to answer them, which could help you better understand the exercise. Let's get started. Enjoy!

First, install and load the necessary libraries and functions to do this practical.

```
# Remove all previous objects.
rm(list = ls())

# Install packages.
install.packages("tsModel", "splines", "dlnm", "gnm")

# Load packages.
library("tsModel"); library("splines"); library("dlnm"); library("gnm")

# Load functions.
source("qAIC.R"); source("findmin.R")
```

In this practical, you will analyze a time-series dataset for the city of London, between 2002-2006. The dataset includes daily mortality counts, mean temperature (ºC), relative humidity (%), ozone (μg/m^3), and particulate matter with aerodynamic diameter <10 μg/m^3 (PM10).

```
# Load and inspect the dataset.
data <- read.csv('london.csv')
names(data)
head(data)

# Formatting date.
data$date <- as.Date(data$date)
```

1. Describing and visualizing time-series data

Let us make daily time-series plots for death counts and temperature, respectively.

```
par(mex=0.8,mfrow=c(2,1))

# Deaths.
plot(data$date, data$all, type="l", col= "blue", ylab="Num. of Deaths"
       , xlab="Date")
```

```
# Temperature.
plot(data$date, data$tmean, type="l", col= "red", ylab="Temperature (ºC)"
       , xlab="Date")

layout(1)
```

To examine the time components, the time-series data can be decomposed in frequency components. The timescales usually includes a single cycle over the entire series, 2-14 cycles, and 15 or more cycles which correspond roughly to the long-term trends, seasonal trends, and higher frequency short-term trends.

```
par(mex=0.8,mfrow=c(2,3))

# Deaths.
all.dc <- tsdecomp(data$all, c(1,2,15,1825))
plot(data$date, all.dc[,1], xlab="Date", ylab="Long-term trend", main="Deaths")
plot(data$date, all.dc[,2], xlab="Date", ylab="Seasonal")
plot(data$date, all.dc[,3], xlab="Date",  ylab="Residual")

# Temperature.
tmean.dc <- tsdecomp(data$tmean, c(1,2,15,1825))
plot(data$date, tmean.dc[,1], xlab="Date", ylab="Long-term trend",
       main="Temperature (ºC)")
plot(data$date, tmean.dc[,2], xlab="Date", ylab="Seasonal")
plot(data$date, tmean.dc[,3], xlab="Date", ylab="Residual")

layout(1)
```

However, you can also examine the time components using boxplots by year, month and day-of-week.

```
par(mex=0.8,mfrow=c(2,3))

# Deaths.
boxplot(all~year, data=data, ylab="Num. of Deaths", xlab="Year")
boxplot(all~month, data=data,ylab="Num. of Deaths", xlab="Year")
boxplot(all~dow, data=data, ylab="Num. of Deaths", xlab="Day-of-week")

# Temperature.
boxplot(tmean~year, data=data, ylab="Temperature (ºC)", xlab="Year")
boxplot(tmean~month, data=data, ylab="Temperature (ºC)", xlab="Month")
boxplot(tmean~dow, data=data, ylab="Temperature (ºC)", xlab="Day-of-week")

layout(1)
```

Questions. What type of time components do you observe in the daily mortality and mean temperature time-series? Are the time components of both time-series related?

---

2. Adjusting for time components in time-series regression

First, check the overdispersion and autocorrelation of the daily mortality time series.

```
# Check overdispersion parameter.
model0 <- glm(all ~ 1, data=data, family=quasipoisson)
summary(model0)

# Autocorrelation plot.
pacf(data$all, lag=30, ylim=c(0,1))
```

Questions. How much overdispersion shows the daily mortality data? Do you think the mortality counts are distributed independently?

2.1. Time-stratified models

A simple approach to adjust for time components in a regression model is by using indicator variables for year and month.

```
par(mex=0.8,mfrow=c(3,1))

# Model with year.
model1a <- glm(all ~ factor(year), data=data, family=quasipoisson)
summary(model1a)
pred1a <- predict(model1a, type="response")

plot(data$date, data$all, ylim=c(100,300), pch=19, cex=0.5, col=grey(0.6),
     ylab="Num. of all", xlab="Date")
lines(data$date, pred1a, lwd=5, col="blue")

# Model with month.
model1b <- glm(all ~ factor(month), data=data, family=quasipoisson)
summary(model1b)
pred1b <- predict(model1b, type="response")

plot(data$date, data$all, ylim=c(100,300), pch=19, cex=0.5, col=grey(0.6),
     ylab="Num. of all", xlab="Date")
lines(data$date, pred1b, lwd=5, col="blue")

# Model with year and month.
model1c <- glm(all ~ factor(year)+factor(month), data=data, family=quasipoisson)
summary(model1c)
pred1c <- predict(model1c, type="response")

plot(data$date, data$all, ylim=c(100,300), pch=19, cex=0.5, col=grey(0.6),
     ylab="Num. of all", xlab="Date")
lines(data$date, pred1c, lwd=5, col="blue")

layout(1)
```

Questions. How do you interpret the model parameters of these Poisson regression models? How much the overdispersion has improved? Which of these time-stratified models do you think best fits the time-components of the daily mortality data?

2.2. Periodic functions

Now let us use periodic functions to fit the time components for annual, semi-annual, quarterly periodicities.

```
par(mex=0.8,mfrow=c(3,1))

# Model with one sine-cosine pairs.
data$time <- seq(nrow(data))
fourier <- harmonic(data$time, nfreq=1, period=365.25)

model2a <- glm(all ~ fourier + time, data=data, family=quasipoisson)
summary(model2a)
pred2a <- predict(model2a, type="response")

plot(data$date, data$all, ylim=c(100,300), pch=19, cex=0.5, col=grey(0.6)
      , ylab="Num. of all", xlab="Date")
lines(data$date, pred2a, lwd=5, col="blue")

# Model with two sine-cosine pairs.
fourier <- harmonic(data$time, nfreq=2, period=365.25)

model2b <- glm(all ~ fourier + time, data=data, family=quasipoisson)
summary(model2b)
pred2b <- predict(model2b, type="response")

plot(data$date, data$all, ylim=c(100,300), pch=19, cex=0.5, col=grey(0.6)
      , ylab="Num. of all", xlab="Date")
lines(data$date, pred2b, lwd=5, col="blue")

# Model with four sine-cosine pairs.
fourier <- harmonic(data$time, nfreq=4, period=365.25)
model2c <- glm(all ~ fourier + time, data=data, family=quasipoisson)
summary(model2c)

pred2c <- predict(model2c, type="response")
plot(data$date, data$all, ylim=c(100,300), pch=19, cex=0.5, col=grey(0.6)
      , ylab="Num. of all", xlab="Date")
lines(data$date, pred2c, lwd=5, col="blue")

layout(1)
```

Questions. How much the overdispersion has improved? Which of these periodic functions models do you think best fits the time-components of the daily mortality data?

2.3. Spline functions

Finally, we use spline functions. Let us start by fitting a natural cubic spline with 1 degree of freedom (df) per year, and next use 6 and 12 df per year.

```
par(mex=0.8,mfrow=c(3,1))

# Model with natural cubit splits with 1 df/year.
numyears <- length(unique(data$year))
spl <- ns(data$time, df=1*numyears)

model3a <- glm(all ~ spl , data=data, family=quasipoisson)
summary(model3a)
pred3a <- predict(model3a, type="response")

plot(data$date, data$all, ylim=c(100,300), pch=19, cex=0.5, col=grey(0.6),
     ylab="Num. of all", xlab="Date")
lines(data$date, pred3a, lwd=5, col="blue")

# Model with natural cubit splits with 6 df/year.
spl <- ns(data$time, df=6*numyears)

model3b <- glm(all ~ spl , data=data, family=quasipoisson)
summary(model3b)
pred3b <- predict(model3b, type="response")

plot(data$date, data$all, ylim=c(100,300), pch=19, cex=0.5, col=grey(0.6),
     ylab="Num. of all", xlab="Date")
lines(data$date, pred3b, lwd=5, col="blue")

# Model with natural cubit splits with 12 df/year.
spl <- ns(data$time, df=12*numyears)

model3c <- glm(all ~ spl , data=data, family=quasipoisson)
summary(model3c)
pred3c <- predict(model3c, type="response")

plot(data$date, data$all, ylim=c(100,300), pch=19, cex=0.5, col=grey(0.6),
     ylab="Num. of all", xlab="Date")
lines(data$date, pred3c, lwd=5, col="blue")

layout(1))
```

Questions. How much the overdispersion has improved fitting the time components with splines? Which of these spline models do you think best fits the time-components of the daily mortality data? Feel free to explore alternative choices for the degrees of freedom per year and fit the spline model by yourself.

---

3. Comparing modeling strategies

Compare the model fit using time stratified models with calendar variables, periodic functions, and flexible functions to adjust for the time components.

```
par(mex=0.8,mfrow=c(3,1))

# Time-stratified model.
qaic0 <- qAIC(model0, type="dev")
disp0 <- sqrt(summary(model1b)$dispersion)
res0 <- residuals(model0, type="response")
pacf(res0, na.action=na.omit, ylim=c(0,1),
      main=paste("Time stratified model | ", "qAIC=" , round(qaic0,1) , "
      , overdisp=" , round(disp0,2)))

# Time-stratified model.
qaic1 <- qAIC(model1c, type="dev")
disp1 <- sqrt(summary(model1b)$dispersion)
res1 <- residuals(model1b, type="response")
pacf(res1, na.action=na.omit, ylim=c(0,1),
      main=paste("Time stratified model | ", "qAIC=" , round(qaic1,1) , "
      , overdisp=" , round(disp1,2)))

# Periodic functions.
qaic2 <- qAIC(model2c, type="dev")
disp2 <- sqrt(summary(model2c)$dispersion)
res2 <- residuals(model2c, type="response")
pacf(res2, na.action=na.omit, ylim=c(0,1),
      main=paste("Periodic functions | ", "qAIC=" , round(qaic2,1) , "
      , overdisp=" , round(disp2,2)))

# Spline functions.
qaic3 <- qAIC(model3b, type="dev")
disp3 <- sqrt(summary(model3b)$dispersion)
res3 <- residuals(model3b, type="response")
pacf(res3, na.action=na.omit, ylim=c(0,1),
      main=paste("Splines | ", "qAIC=" , round(qaic3,1) , "
      , overdisp=" , round(disp3,2)))

# End of figure.
layout(1)
```

Question. Which model best fits the time components of the daily mortality counts?

---

4. Exposure-response function

Once a core model has been identified, you can explore the shape of the exposure-response between the outcome and environmental exposure variables. Let us first assume a linear association between mean temperature and daily mortality.

```
# Linear association.
tmean.b <- onebasis(data$tmean, fun="lin")
model <- glm(all ~ spl + factor(dow) + tmean.b, data=data, family=quasipoisson)
summary(model)
pred.lin <- crosspred(tmean.b, model, by=1)
plot(pred.lin)

# Linear association centered at MMT.
pred.lin <- crosspred(tmean.b, model, by=1, cen=min(data$tmean))
plot(pred.lin)
```

We now explore a nonlinear association, using a natural cubic spline with 4 df.

```
# Non-linear association.
tmean.b <- onebasis(data$tmean, fun="ns", df=4)
model <- glm(all ~ spl + factor(dow) + tmean.b, data=data, family=quasipoisson)
summary(model)
pred.nl <- crosspred(tmean.b, model, by=1)
plot(pred.nl)

# Non-linear association centered at MMT.
mmt <- findmin(tmean.b, model)
pred.nl <- crosspred(tmean.b, model, by=1, cen=mmt)
plot(pred.nl)
```

Questions. How do you interpret the exposure-response functions from these regression models? At what value are the exposure-response functions centered and why is it necessary to re-center them at the Minimum Mortality Temperature (MMT)? Which exposure-response do you think best describe the temperature-mortality association? Feel free to explore alternative choices for the degrees of freedom to fit the natural cubic spline. Can you discuss the limitations of the estimated temperature-mortality association?

---

5. Comparing time-series and case-crossover

Finally, let us compare the temperature-mortality association estimated using a time-series regression model and a time-stratified case-crossover design.

```
# Generate time-stratified strata.
data$month <- as.factor(data$month)
data$year  <- as.factor(data$year)
data$dow   <- as.factor(data$dow)
data$stratum <- with(data, as.factor(year:month:dow))

# Fit fixed-effects conditional quasi-Poisson regression.
model <- gnm(all ~ tmean.b, data=data, family=quasipoisson, eliminate=stratum)
mmt <- findmin(tmean.b, model)
pred.cc <- crosspred(tmean.b, model, by=1, cen=mmt)
plot(pred.cc)
```

```
par(mex=0.8,mfrow=c(1,2))

# Time-series.
plot(pred.nl, main="Time-series")

# Case-crossover.
plot(pred.cc, main="Case-crossover")

layout(1)
```

Questions. Do you think the functions are similar? Can you discuss the advantages and limitations both approaches?
