

Sèries Temporals i Predicció

Práctica 2

Extracción de la tendencia y la estacionalidad

La descomposición de series temporales es un procedimiento que transforma una serie en múltiples series. Generalmente, en tres componentes:

- **Estacional:** Patrones que se repiten en un período fijo de tiempo, por ejemplo, un sitio web puede recibir más visitas durante los fines de semana: esto producirá datos con una estacionalidad de 7 días. No todas las series tienen componentes estacionales.
- **Tendencia:** Se trata de movimientos crecientes o decrecientes sostenidos en el tiempo, por ejemplo, un sitio web que se está volviendo cada vez más popular, debería mostrar una tendencia creciente.
- **Aleatorio:** También conocido como ruido. Es lo que queda de la serie original después de quitar las componentes estacionales y la tendencia.

Cuando una serie tiene un componente estacional, además de una tendencia, ya hemos visto que la función `decompose` nos permite aislarlas, y obtener la serie original como suma o producto (según sea el caso) de una tendencia, una componente estacional y un ruido.

A continuación veremos cómo hacerlo paso a paso, para entender y reproducir el procedimiento que hay dentro de la función.

1 Descomposición aditiva

Si la variación estacional parece constante (por ejemplo, no crece cuando crecen los valores de la serie) entonces usaremos una descomposición aditiva

$$X_t = m_t + s_t + Y_t$$

El conjunto de datos `ausbeer` de la librería `fpp`, que corresponde a la producción trimestral de cerveza en Australia (en megalitros) entre 1956:Q1 y 2008:Q3, exhibe una componente estacional que no parece cambiar con la magnitud de las observaciones. Nos quedaremos con un trozo más corto de la serie, el comprendido entre el Q3 de 1957 y el Q2 de 1973 para evitar el aparente cambio de tendencia (que observamos en el gráfico).

Hagamos la descomposición paso a paso:

1.1 Obtener y mirar los datos

```
install.packages("fpp"); library(fpp) #data
data(ausbeer)
plot.ts(ausbeer)
```

```
ausbeerrec = tail(head(ausbeer, 17*4+2),17*4-4)
# hay muchas otras formas de hacer esto, experimentad!
head(ausrec);tail(ausrec)
plot.ts(ausbeerrec, col="navy",lwd=2)
```

1.2 Detectar el Trend (tendencia)

Para detectar la tendencia subyacente, suavizaremos la serie usando un promedio móvil centrado, para ésto es importante usar como orden el de la estacionalidad (si la serie la tiene, si no, no es tan relevante y se pueden usar promedios móviles de cualquier orden). En este caso, usaremos de orden 4 ya que los datos son trimestrales (hay 4 observaciones por año).

```
trend_beer = ma(ausbeerrec, order = 4, centre = T)
plot(as.ts(ausbeerrec), col="navy",lwd=2)
lines(trend_beer, col="gold",lwd=3)
plot.ts(trend_beer,col="gold",lwd=3)
```

Ejercicio: Probar diferentes órdenes , por ejemplo 2 y 10 para la media móvil y agragarlas al gráfico. Observar qué diferencia presentan con la de orden 4.

1.3 Detrend: quitar la tendencia

Al quitar la tendencia de la serie, en este caso, restándola, obtendremos una nueva serie que muestra más claramente la componente estacional.

```
detrend_beer = ausbeerrec - trend_beer
plot.ts(detrend_beer))
```

1.4 Promediar la estacionalidad

A partir de la serie *detrended* es fácil calcular la estacionalidad promedio. Para hacerlo, ponemos la serie en una matriz, de manera que cada columna contenga los elementos correspondientes a cada período (mismo día, mes, trimestre, etc). Finalmente, promediamos cada columna.

```
m_beer = t(matrix(data = detrend_beer, nrow = 4))
seasonal_beer = colMeans(m_beer, na.rm = T)
plot.ts(rep(seasonal_beer,16))
```

1.5 Examinar el ruido restante

Al quitarle a la serie original la tendencia y la estacionalidad, nos quedamos con ruido aleatorio. Sobre éste último propondremos modelos para poder capturar las dependencias temporales.

```
random_beer = ausbeerrec - trend_beer - seasonal_beer
plot.ts(random_beer)
```

2 Reconstrucción de la serie original

Para recuperar la señal original, basta con sumar las componentes que acabamos de extraer:

$$X_t = m_t + s_t + Y_y$$

Perderemos algunos de los puntos al principio y al final de la serie, porque los promedios móviles consumen datos antes de producir promedios..

```
recomposed_beer = trend_beer+seasonal_beer+random_beer
plot.ts(recomposed_beer,col="orchid",lwd=2)
points(ausbeerrec,col="orchid4") #observaciones originales
```

Finalmente, hacemos la descomposición usando las funciones de R, `decompose` y `stl`. (Mirar el `help` de ambas funciones).

```
# Usando la funcion decompose: (frequency = 4 obs/año)
ts_beer = ts(ausbeerrec, frequency = 4)
decompose_beer = decompose(ts_beer, "additive")

plot.ts(decompose_beer$seasonal)
plot.ts(decompose_beer$trend)
plot.ts(decompose_beer$random)
plot(decompose_beer)

### Descomposicion usando la funcion stl (ver el help!)

ts_beer = ts(ausbeerrec, frequency = 4)
stl_beer = stl(ts_beer, "periodic")
names(stl_beer)
seasonal_stl_beer=stl_beer$time.series[,1]
trend_stl_beer = stl_beer$time.series[,2]
random_stl_beer=stl_beer$time.series[,3]
```

```
plot(ts_beer)
plot.ts(seasonal_stl_beer)
plot(trend_stl_beer)
plot(random_stl_beer)
plot(stl_beer)
```

3 Descomposición multiplicativa

Los datos mensuales de pasajeros internacionales AirPassengers de Box y Jenkins son un buen ejemplo para la descomposición multiplicativa. Cuantos más pasajeros hay, más efecto estacional se observa.

Los pasos para hacer la descomposición son muy similares al caso aditivo:

3.1 Obtener y mirar los datos

```
data(AirPassengers)
timeserie_air = AirPassengers
plot.ts(timeserie_air)
```

3.2 Detectar el Trend (tendencia)

El proceso es idéntico que para el modelo aditivo. La estacionalidad es anual, y como en este caso tenemos observaciones mensuales, elegimos una ventana de tamaño 12 para el promedio móvil.

```
trend_air = ma(timeserie_air, order = 12, centre = T)
plot.ts(timeserie_air)
lines(trend_air,col="darkcyan",lwd=2)
plot.ts(trend_air)
```

3.3 Detrend: quitar la tendencia

Para quitar la tendencia de la serie en este caso, dividimos.

```
detrend_air = timeserie_air / trend_air
plot.ts(detrend_air)
```

3.4 Promediar la estacionalidad

En este caso usaremos 12 columnas en la matriz, una por cada mes.

```
m_air = t(matrix(data = detrend_air, nrow = 12))
seasonal_air = colMeans(m_air, na.rm = T)
plot.ts(rep(seasonal_air,12))
```

3.5 Examinar el ruido restante

Quitar a la serie original la tendencia y la estacionalidad, en este caso, dividiendo, para obtener el ruido.

```
random_air = timeserie_air / (trend_air * seasonal_air)
plot.ts(random_air)
```

También podemos quitar solamente la estacionalidad, y en ese caso obtenemos la llamada serie **ajustada por estacionalidad**, observar que en este caso, en 1960 se observa una anomalía que no era detectable a ojo a partir de los datos originales.

```
sadjust_air = timeserie_air / seasonal_air
plot.ts(sadjust_air, col = "orchid", lwd = 2)
```

4 El suavizado Exponencial (Holt-Winters)

Si la serie tiene período d ,

$$\hat{X}_{t+h} = a_t + hb_t + s_{t-d+1+(h-1)\bmod(d)}$$

donde

$$a_t = \alpha(X_t - s_{t-p}) + (1 - \alpha)(a_{t-1} + b_{t-1})$$

$$b_t = \beta(a_t - a_{t-1}) + (1 - \beta)b_{t-1}$$

$$s_t = \gamma(X_t - a_t) + (1 - \gamma)s_{t-d}$$

El modelo multiplicativo se obtiene en forma similar, haciendo los cambios obvios.

```
hw.beer=HoltWinters(ausbeer, alpha = NULL, beta = NULL, gamma = NULL,
                    seasonal = "additive")
p.beer=predict(hw.beer, n.ahead = 5, prediction.interval = TRUE,
               level = 0.95)
plot(hw.beer,p.beer)
```

5 Otros usos de las medias móviles.

En finanzas, el análisis técnico consiste en producir diversas descripciones de las series de precios de los activos financieros, $\{P_t, t = 0, 1, \dots\}$.

Una estrategia para compra o vender es la siguiente: se ajusta a la serie de precios de cierre un promedio móvil de corto plazo $MA(n)_t$ (ventana de tamaño n) y otro de largo plazo $MA(m)_t$, con una ventana de tamaño $m > n$, y se mira el proceso $X_t = MA(n)_t - MA(m)_t$, y si en un instante t_i cambia de signo, se procede así:

- Si $X_{t_i-1} < 0$ y $X_{t_i} > 0$, comprar en el instante $t_i + 1$ al precio de apertura del día siguiente.
- $X_{t_i-1} > 0$ y $X_{t_i} < 0$, vender.

Se suele usar $n = 20$ y $m = 50$.

```
##Technical analysis with quantmod
library(quantmod)
getSymbols("AAPL")
adj=AAPL$AAPL.Adjusted["2017-01-11::2018-09-20"]
plot.ts(adj,type="l",lwd=2)
lines(ma(adj, order = 50, centre = T),col="orchid")
lines(ma(adj, order = 20, centre = T),col="gold")

### Un grafico mas profesional:
chartSeries(AAPL, subset="2017-01-11::2018-09-20",
            theme=chartTheme('white',up.col='green',dn.col='red'),
            TA=c(addBBands(n=20,sd=2),addSMA(n=50,col="blue")))
```

Referencia: Computational Finance, A. Arratia, <http://computationalfinance.lsi.upc.edu/>, RLab 6.1.