# SÉRIES TEMPORAIS E ECONOMETRIA - LISTA 4

27 de novembro de 2025

EMAp - FGV

**Questão 1.** Considere o Modelo de Nível Local (Local Level Model), também conhecido como Random Walk plus Noise, definido pelas equações:

$$y_t = \mu_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma_\epsilon^2)$$

$$\mu_{t+1} = \mu_t + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2)$$

onde erros $\epsilon_t$ e $\eta_t$ são independentes entre si.

a) Identifique a Equação de Observação e a Equação de Estado. Qual é o vetor de estado?

b) Explique o comportamento da série $y_t$ quando $q = \sigma_\eta^2/\sigma_\epsilon^2 \to 0$ e quando $q \to \infty$.

c) Mostre que $\Delta y_t$ segue um processo MA(1).

**Solução.**

a)
- Equação de Observação: $y_t = \mu_t + \epsilon_t$. Relaciona a variável observada ($y_t$) com o estado não observado.
- Equação de Estado: $\mu_{t+1} = \mu_t + \eta_t$. Descreve a dinâmica do estado ao longo do tempo.
- Vetor de Estado: Neste caso univariado, o estado é escalar: $\alpha_t = \mu_t$.

b) A razão sinal-ruído $q$ controla a variabilidade do nível:
- Se $q \to 0$ (ou seja, $\sigma_\eta^2 \to 0$), então $\mu_{t+1} \approx \mu_t$. O nível torna-se constante ($\mu_t = \mu$). O modelo colapsa para $y_t = \mu + \epsilon_t$, que é o modelo de nível constante estimado por OLS (média global).
- Se $q \to \infty$ (ruído de observação desprezível), $y_t \approx \mu_t$, e como $\mu_t$ é um Random Walk, a série original se comporta puramente como um passeio aleatório.

c) Tomando a primeira diferença:

$$\Delta y_t = y_t - y_{t-1} = (\mu_t + \epsilon_t) - (\mu_{t-1} + \epsilon_{t-1})$$

Sabemos que $\mu_t - \mu_{t-1} = \eta_{t-1}$. Logo:

$$\Delta y_t = \eta_{t-1} + \epsilon_t - \epsilon_{t-1}$$

O lado direito é uma soma de ruídos brancos defasados (média zero). A autocovariância na defasagem 1 é não nula ($\text{Cov}(\Delta y_t, \Delta y_{t-1}) = -\sigma_\epsilon^2$), e zero para defasagens maiores que 1. Por definição, isso caracteriza um processo MA(1).

**Questão 2.** Considere um processo AR(2) estacionário $y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \epsilon_t$. Defina o vetor de estado como $\alpha_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix}$. Encontre as matrizes $Z$, $T$ e $R$ para a representação de espaço de estados.

**Solução.** Queremos escrever na forma:

$$y_t = Z\alpha_t \quad \text{e} \quad \alpha_{t+1} = T\alpha_t + R\epsilon_{t+1}$$

1. Equação de Observação: Como $\alpha_t = [y_t, y_{t-1}]'$, para recuperar $y_t$ precisamos selecionar o primeiro elemento:

$$Z = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

2. Equação de Estado: Precisamos expressar $\alpha_{t+1} = [y_{t+1}, y_t]'$ em função de $\alpha_t$.
   - Linha 1: $y_{t+1} = \phi_1 y_t + \phi_2 y_{t-1} + \epsilon_{t+1}$ (Definição do AR(2)).
   - Linha 2: $y_t = 1 \cdot y_t + 0 \cdot y_{t-1}$ (Identidade).

   Em forma matricial:
$$\begin{bmatrix} y_{t+1} \\ y_t \end{bmatrix} = \underbrace{\begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix}}_{T} \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{R} \epsilon_{t+1}$$
   Portanto, $Z = [1\ 0]$, $T = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix}$ e $R = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

**Questão 3.** Escreva as equações de Predição (Time Update) do Filtro de Kalman que levam de $(a_{t|t}, P_{t|t})$ para $(a_{t+1}, P_{t+1})$ considerando o sistema $\alpha_{t+1} = T\alpha_t + \eta_t$ e $y_t = Z\alpha_t + \epsilon_t$.

**Solução.** Dado o estimador filtrado (atualizado) no tempo $t$, denotado por $a_{t|t} = \mathbb{E}[\alpha_t|Y_t]$ e sua covariância $P_{t|t}$, o passo de predição projeta o estado para $t+1$ usando a dinâmica do modelo:

1. Predição do Estado (Média):
$$a_{t+1} = \mathbb{E}[T\alpha_t + \eta_t \,|\, Y_t] = T\mathbb{E}[\alpha_t \,|\, Y_t] + 0$$
$$a_{t+1} = Ta_{t|t}$$
2. Predição da Covariância:
$$P_{t+1} = var(T\alpha_t + \eta_t \,|\, Y_t) = Tvar(\alpha_t \,|\, Y_t)T' + var(\eta_t)$$
$$P_{t+1} = TP_{t|t}T' + Q$$
   Essas equações preparam o filtro para receber a observação $y_{t+1}$ no próximo passo.

**Questão 4.** Defina Predição, Filtragem e Suavização em termos da esperança condicional $\mathbb{E}[\alpha_t \,|\, Y_s]$. Por que a variância do estimador suavizado é menor?

**Solução.** Seja $n$ o tamanho total da amostra.

- Predição: Estimação de $\alpha_t$ dado $Y_s$ com $s < t$ (geralmente $s = t-1$). $\hat{\alpha}_{t|t-1} = \mathbb{E}[\alpha_t \,|\, Y_{t-1}]$.
- Filtragem: Estimação de $\alpha_t$ dado $Y_s$ com $s = t$ (em tempo real). $\hat{\alpha}_{t|t} = \mathbb{E}[\alpha_t \,|\, Y_t]$.
- Suavização: Estimação de $\alpha_t$ dado $Y_s$ com $s > t$ (geralmente $s = n$, usando toda a amostra). $\hat{\alpha}_{t|n} = \mathbb{E}[\alpha_t \,|\, Y_n]$.

A variância do estimador suavizado ($P_{t|n}$) é menor ou igual à variância do filtrado ($P_{t|t}$) porque o suavizador utiliza mais informação (observações passadas $y_{1:t}$ E futuras $y_{t+1:n}$) para estimar o estado no instante $t$. O filtro utiliza apenas o passado e o presente. Matematicamente, $P_{t|n} \leqslant P_{t|t}$.

**Questão 5.** Utilize o dataset `Nile` no R. Ajuste um Modelo de Nível Local, reporte as variâncias e plote os resultados (série, filtrado, suavizado).

**Solução.** Abaixo segue o código R comentado e a interpretação dos resultados.

```r
# Carregar dados
data("Nile")

# a) Ajuste do Modelo de Nível Local (Gaussian Random Walk + Noise)
# Usamos StructTS, que usa Máxima Verossimilhança para estimar as variâncias
```

```r
fit <- StructTS(Nile, type = "level")
print(fit)

# b) Variâncias Estimadas
# fit£coef contém as variâncias estimadas
var_epsilon <- fit$coef["epsilon"] # Variância da Observação (Irregular)
var_eta <- fit$coef["level"]       # Variância do Estado (Level)

cat("Var(Obs):", var_epsilon, "\nVar(Estado):", var_eta, "\n")

# Se Var(Estado) for pequena em relação a Var(Obs), o nível é suave.
# Se Var(Estado) for grande, o nível reage rápido aos dados.

# c) Filtragem e Suavização
# O StructTS já faz a filtragem/suavização internamente, mas podemos usar
# o tsSmooth para extrair os componentes suavizados.
# Para ver o filtrado explicitamente,
# podemos usar KalmanRun ou pacotes como dlm.
# Aqui focaremos no resultado do StructTS (Suavizado).

smoothed_level <- tsSmooth(fit)

plot(Nile, type="p", pch=20, col="grey",
     main="Fluxo do Rio Nilo: Dados vs Modelo")
lines(fitted(fit), col="blue", lwd=2) # Nível Filtrado
lines(smoothed_level, col="red", lwd=2) # Nível Suavizado

legend("bottomleft", legend=c("Dados", "Filtrado", "Suavizado"),
       col=c("grey", "blue", "red"), lty=c(NA, 1, 1), pch=c(20, NA, NA))
```

Interpretação: O gráfico mostra uma queda abrupta no nível do rio por volta de 1899. O filtro de Kalman (linha azul) demora alguns períodos para se ajustar completamente ao novo nível ("lag"), enquanto o suavizador (linha vermelha), por usar dados futuros, ajusta a quebra de nível de forma mais precisa e centrada no momento da ocorrência (construção da barragem).