1. Resumen, paso a paso, de lo que hicimos

En esta sección explicamos, con palabras sencillas, qué hace ahora nuestro código después del cambio que hicimos para dejar de imponer que las diferencias $\Delta^d t_t$ sean constantes fuera del entrenamiento.

1.1. Paso 1: Tenemos una serie y usamos log–precios

- Para cada activo (ticker) tenemos una serie de precios

$$P_1, P_2, \ldots, P_N.$$

- En lugar de trabajar con los precios directos, trabajamos con los *log–precios*

$$Z_t := \log P_t, \qquad t = 1, \ldots, N.$$

  Esto hace que los cambios porcentuales (retornos) se vuelvan diferencias simples y la escala sea más estable.

1.2. Paso 2: Partimos la serie en train, validación y test

Dividimos el tiempo en tres bloques consecutivos:

$$\underbrace{1, 2, \ldots, N_{\text{train}}}_{\text{entrenamiento}} \quad | \quad \underbrace{N_{\text{train}} + 1, \ldots, N_{\text{train}} + N_{\text{val}}}_{\text{validación}} \quad | \quad \underbrace{N_{\text{train}} + N_{\text{val}} + 1, \ldots, N}_{\text{test}}.$$

- En **train** ajustamos el modelo de Guerrero.
- En **validación** elegimos qué tan suave debe ser la tendencia (el parámetro s).
- En **test** medimos el error final del pronóstico.

1.3. Paso 3: Ajuste de Guerrero en el segmento de entrenamiento

En el tramo de entrenamiento tenemos los datos

$$Z_1, Z_2, \ldots, Z_{N_{\text{train}}}.$$

Queremos encontrar una *tendencia suave*

$$t_1, t_2, \ldots, t_{N_{\text{train}}}$$

que:

1. Esté cerca de los datos, es decir, que

$$(Z_t - t_t)^2$$

sea pequeño para la mayoría de los t en train.

2. No cambie demasiado rápido: se penalizan las *diferencias de orden d*

$$\Delta^d t_t,$$

que son combinaciones lineales del tipo

$$\Delta^d t_t \approx t_{t+d} - \binom{d}{1} t_{t+d-1} + \cdots + (-1)^d t_t.$$

El modelo de Guerrero para train se puede escribir, de forma simple, como:

$$\min_{\{t_t\},\, m} \left[\sum_{t=1}^{N_{\text{train}}} (Z_t - t_t)^2 \; + \; \lambda \sum_t (\Delta^d t_t - m)^2 \right],$$

donde:

- $\lambda > 0$ controla cuánta suavidad queremos.
- m es un escalar que representa la media de las diferencias de orden d.
- No imponemos que $\Delta^d t_t = m$ exactamente; solo penalizamos que se aleje de m.

El resultado de este paso es:

$$\hat{t}_1, \ldots, \hat{t}_{N_{\text{train}}} \quad \text{y} \quad \hat{m},$$

que son la tendencia suavizada en train y la media estimada de las diferencias.

1.4. Paso 4: El problema original con el pronóstico

Antes del cambio, para poder *extender* la tendencia hacia validación y test, hicimos lo siguiente:

- Fijábamos la tendencia ajustada en train:

$$p_t = \hat{t}_t, \qquad t = 1, \ldots, N_{\text{train}}.$$

- Fuera del train imponíamos

$$\Delta^d p_t = \hat{m} \quad \text{para } t > N_{\text{train}}.$$

Esto significa que, en el pronóstico, obligábamos a que la diferencia de orden d fuera *exactamente constante*. Matemáticamente, eso hace que la trayectoria p_t (train + forecast) sea un **polinomio de grado** d en el tiempo.

Problema:

- Guerrero nunca dice que $\Delta^d t_t$ tenga que ser constante, solo que se penaliza el alejamiento de un m.
- Al extrapolar imponiendo $\Delta^d t_t = \hat{m}$ estamos introduciendo una suposición extra: que fuera del train la serie sigue un polinomio exacto, lo cual puede ser demasiado rígido.

1.5. Paso 5: Qué cambio hicimos ahora

La idea nueva fue:

> Mantener la tendencia de Guerrero tal como está en train, pero cambiar por completo la forma en que la extendemos a validación y test, sin imponer que $\Delta^d t_t$ sea constante.

Entonces, ahora:

1. **En train:** todo sigue igual. Para cada valor de suavidad s:
   - usamos `fit_for_s` para obtener la tendencia de Guerrero

     $$\hat{t}_1, \ldots, \hat{t}_{N_{\text{train}}},$$

   - y la media de diferencias $\hat{m}$ (que se usa solo dentro del solver, no en el pronóstico).
2. **En val + test:** en lugar de imponer $\Delta^d p_t = \hat{m}$, buscamos una extensión

   $$t_{N_{\text{train}}+1}, \ldots, t_N$$

   que sea lo más suave posible, en este sentido:

   $$\min_{\{t_t\}_{t>N_{\text{train}}}} \sum_r (\Delta^d t_r)^2,$$

sujetando a que

$$t_1, \dots, t_{N_{\text{train}}} = \hat{t}_1, \dots, \hat{t}_{N_{\text{train}}}.$$

Es decir:

- mantenemos fijos los puntos de entrenamiento,
- elegimos los puntos futuros para que la serie completa (train + futuro) tenga las diferencias de orden d lo más pequeñas posibles.

Intuitivamente:

- Piensa en la serie t_t como una cuerda discreta.
- En train ya conocemos la forma de la cuerda.
- Para los puntos futuros, la cuerda elige la forma más suave posible que continúa la parte conocida, sin forzar una recta, ni una parabola ni un polinomio de grado exacto.

1.6. Paso 6: Cómo se hace esto en el código (idea sencilla)

En el código definimos una función `extend_trend_minimum_roughness` que hace exactamente esto:

1. Crea la matriz de diferencias de orden d sobre todo el horizonte:
$$K_{\text{full}} \in \mathbb{R}^{(N-d)\times N}.$$
2. Separa la serie t_{full} en:
$$t_{\text{full}} = \underbrace{t_{\text{train}}}_{\text{conocido}} \quad \| \quad \underbrace{u}_{\text{desconocido (futuro)}}.$$
3. Escribe la rugosidad como:
$$\|K_{\text{full}} t_{\text{full}}\|^2 = \|K_k\, t_{\text{train}} + K_u\, u\|^2,$$
donde K_k son las columnas de train y K_u las de futuro.
4. Minimizar la rugosidad respecto a u lleva a un sistema lineal:
$$(K_u^\top K_u)\, u = -K_u^\top K_k\, t_{\text{train}},$$
que resolvemos con `np.linalg.lstsq`. No hay λ aquí porque solo nos importa el minimizador, no la escala.

El resultado final es un vector

$$t_1, \dots, t_{N_{\text{train}}}, t_{N_{\text{train}}+1}, \dots, t_N$$

donde:

- Los primeros N_{train} son exactamente los de Guerrero.
- Los restantes $N - N_{\text{train}}$ son la extensión de *mínima rugosidad* que no fuerza $\Delta^d t_t$ a ser constante.

1.7. Paso 7: RMSE y gráficas con la nueva tendencia

Una vez que tenemos la nueva tendencia extendida

$$t_1, \dots, t_N,$$

hacemos lo de siempre:

- Calculamos errores en cada segmento:

$$\begin{aligned} e_t^{\text{train}} &= t_t - Z_t \quad (t \leq N_{\text{train}}), \\ e_t^{\text{val}} &= t_t - Z_t \quad (t \in \text{validación}), \\ e_t^{\text{test}} &= t_t - Z_t \quad (t \in \text{test}). \end{aligned}$$

- Calculamos RMSE simples y ponderadas en train, val, train+val y test, como ya lo teníamos definido.
- Dibujamos:
  - Z_t (serie original).
  - La tendencia de Guerrero en train.
  - La tendencia extendida suave (train + futuro) con línea punteada.

1.8. Resumen en una frase

Antes: extrapolábamos como polinomio de grado d imponiendo $\Delta^d t_t = \hat{m}$ fuera de train.

Ahora: dejamos intacto Guerrero en train y, para val+test, construimos la extensión más suave posible (minimizando la rugosidad $\sum(\Delta^d t_t)^2$), sin imponer que las diferencias sean constantes.