Clase 031 — Pandas: series de tiempo, resampling, rolling

Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 3 § 3.12 Working with Time Series. · Duración estimada: 90 min.

Clase 031 — Pandas: series de tiempo, resampling, rolling

> *Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 3 § 3.12 Working with Time Series. · Duración estimada: 90 min.*

Objetivo

Que el alumno trabaje con datos temporales correctamente: parsear fechas, indexar por DatetimeIndex, hacer resampling (cambiar la frecuencia) y rolling (ventanas móviles para tendencias).

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Parsear strings de fecha con pd.to_datetime(..., format=..., errors=...).
2. Indexar por DatetimeIndex y slicear con strings de fecha (df.loc['2024-01':'2024-03']).
3. Resamplear a otra frecuencia: df.resample('M').sum(), 'W', 'D', 'H'.
4. Aplicar ventanas móviles con rolling(window).mean() para suavizar tendencias.
5. Manejar zonas horarias con tz_localize y tz_convert.

Temas

#	Tema	Por qué importa
1	pd.to_datetime con errors='coerce'	Parseo robusto.
2	DatetimeIndex y slicing por fecha	Sintaxis natural: '2024-01':'2024-03'.
3	Resampling: 'D', 'W', 'M', 'Q', 'Y', 'H'	Cambiar frecuencia + agregar.
4	Rolling windows	Suavizado, medias móviles.
5	shift y diff	Diferencias entre periodos, lag features.
6	Timezones: localize → convert	Cuando los datos tienen TZ.

Definiciones y características

Timestamp / DatetimeIndex

: Tipo nativo de pandas para fechas-horas. Vectorizado. Permite indexar con strings: df.loc['2024-01':'2024-03'].

pd.to_datetime

: Conversor robusto string→Timestamp. errors='coerce' convierte fallos a NaT (Not a Time) en vez de excepción.

Resampling

: Cambiar la frecuencia de una serie: diaria→mensual, horaria→semanal. Requiere agregación (sum, mean, last, ohlc). Códigos: 'D', 'W', 'ME' (month-end), 'h', 'min'.

Rolling window

: Ventana móvil: para cada punto, aplicar función a los últimos N (.rolling(N).mean()). Suaviza tendencias,

calcula medias móviles.

shift / diff / pct_change

: shift(N): desplaza N posiciones (NaN al inicio). diff(N): s - s.shift(N) (cambio absoluto). pct_change(): cambio relativo.

tz_localize vs tz_convert

: localize asigna TZ a fecha naive (no convierte). convert convierte de una TZ a otra (mantiene el instante). Patrón: localize primero, convert después.

Dataset / recursos

Sintético: serie diaria de 2 años de ventas. Sin descarga.

Ejercicios

1. Parseo robusto. Lista de fechas con formatos mixtos ('2024-01-15', '15/02/2024', 'foo'). Parsea con errors='coerce'. Reporta NaT.

2. Slice por fecha. Con índice datetime, selecciona Q1 2024 con df.loc['2024-01':'2024-03'].

3. Resample diaria → mensual. Suma ventas por mes con df.resample('M').sum().

4. Rolling 7-day mean. Calcula media móvil de 7 días sobre ventas diarias. Plotea junto a la serie original.

5. shift para lag feature. Crea columna ventas_lag_1 con shift(1). Útil para features de ML.

Homework verificable

Notebook con serie sintética de 2 años: (a) parseo robusto; (b) slice por trimestre; (c) resample a mensual con sum y mean; (d) rolling 7/30 días con plot; (e) diff y pct_change para variación.

Criterio de aceptación: Plots muestran tendencia clara. Resample correcto (#meses esperado).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
pd.to_datetime falla con ValueError	Formato no estándar. Fix: format='%d/%m/%Y
df.loc['2024-13'] lanza KeyError	Mes inválido. Fix: verifica formato — pand
resample('M') warning sobre deprecation	Pandas 2.2+ prefiere 'ME' (month-end) o 'M
Resta de fechas da Timedelta en lugar de n	Es correcto — usa .dt.days, .dt.seconds pa
Operaciones con TZ mezcladas: Cannot compa	Una fecha tiene timezone, la otra no. Fix:

Preguntas frecuentes

¿Cuándo DatetimeIndex?

Cuando vas a hacer resample, slicing por fechas, o cualquier operación temporal. set_index('fecha') después de parsear.

¿resample o groupby(date.dt.month)?

resample es nativo y maneja gaps + zonas horarias mejor. groupby para agrupaciones no temporales ("por mes ignorando año").

¿Rolling cuánto Nuevoiniciar?

Default: NaN en los primeros N-1 (no hay datos suficientes para la ventana). Si quieres usar lo que haya: .rolling(N, min_periods=1).

¿Cómo manejo zonas horarias en producción?

Regla: guarda todo en UTC, convierte solo para mostrar. df['fecha'] = pd.to_datetime(...).dt.tz_localize('UTC') al ingerir, tz_convert('America/Santiago') al exhibir.

¿pd.date_range o pd.timestamp_range?

date_range — timestamp_range no existe. pd.date_range('2024-01-01', '2024-12-31', freq='D') para 365 fechas diarias.

Referencias

- VanderPlas, cap. 3 § 3.12.
- pandas Time Series user guide

Siguiente clase

Clase 032 — Pandas: eval y query

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

rng = np.random.default_rng(42)
fechas = pd.date_range('2024-01-01', '2025-12-31', freq='D')
ventas = pd.Series(
    rng.normal(1000, 200, len(fechas)).cumsum().clip(min=0).astype(int),
    index=fechas,
    name='ventas',
)
print(ventas.head())
```

Archivos complementarios

- notebook.ipynb