Clase 025 — Pandas: datos faltantes

Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 3 § 3.5 Handling Missing Data. · Duración estimada: 75 min.

Clase 025 — Pandas: datos faltantes

> *Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 3 § 3.5 Handling Missing Data. · Duración estimada: 75 min.*

Objetivo

Que el alumno detecte, cuantifique y maneje datos faltantes con criterio. Eliminar es la opción fácil pero suele ser incorrecta: cuándo eliminar, cuándo imputar (media, mediana, forward-fill), y cuándo el faltante es señal que merece su propia columna.

Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Detectar NaN con isna(), notna() y cuantificar por columna/fila.
2. Eliminar filas/columnas con NaN usando dropna con how/thresh/subset.
3. Imputar con fillna: valor escalar, media/mediana, forward/backward fill, interpolación.
4. Distinguir NaN vs None vs pd.NA y por qué importan los dtypes nullable (Int64, boolean).
5. Decidir entre eliminar/imputar/dejar — y crear columna was_missing cuando el faltante es informativo.

Temas

#	Tema	Por qué importa
1	Tipos de missing en pandas: NaN, None, NaT	Cada uno tiene caso de uso.
2	Detección: isna, notna, isna().sum()	First-look obligatorio.
3	dropna: how='any'/'all', thresh, subset	Eliminar con precisión.
4	fillna: escalar, dict, ffill, bfill, inter	Imputar según contexto.
5	Dtypes nullable: Int64, Float64, boolean	El nuevo missing nativo.
6	was_missing como feature	Cuando el missing es señal.

Definiciones y características

NaN (Not a Number)

: Float especial IEEE 754 que representa missing en columnas numéricas. Característica: NO es igual a sí mismo (np.nan == np.nan → False). Usa pd.isna() para detectarlo.

None vs NaN

: None es objeto Python (en columnas object). NaN es float (en columnas numéricas). Pandas trata ambos como missing pero internamente son distintos.

pd.NA

: Sentinel "missing universal" (pandas 1.0+) compatible con dtypes nullable (Int64, boolean, string). Comportamiento más consistente que NaN en operaciones.

MCAR / MAR / MNAR

: MCAR (Missing Completely At Random): missing es aleatorio. MAR (Missing At Random): missing depende de variables observadas. MNAR (Missing Not At Random): depende del propio valor missing (peor caso).

Imputación

: Reemplazar missing con un valor estimado. Estrategias: media/mediana/moda global, por grupo (groupby.transform), forward-fill (series temporales), KNN, regresión, MICE.

was_missing flag

: Columna booleana adicional que registra qué filas tenían missing antes de imputar. Permite al modelo usar el hecho de que faltaba como feature (a veces es informativo).

Dataset / recursos

Palmer Penguins (tiene NaN reales en sex y mediciones). Sin descarga adicional si ya está en clases anteriores.

Ejercicios

1. Cuantifica. Carga penguins, reporta % de NaN por columna y por fila.

2. Eliminar filas con cualquier NaN. df.dropna(how='any'). Compara shape antes/después.

3. Eliminar solo filas con NaN en sex. df.dropna(subset=['sex']). Más selectivo.

4. Imputar. Rellena bill_length_mm con la mediana por especie (groupby + transform). Justifica por qué la mediana es mejor que la media aquí.

5. Forward fill en series temporales. Crea una Series con NaN intercalados. Aplica ffill, bfill, interpolate. Compara.

Homework verificable

Notebook con penguins: (a) reporte completo de missing (% por col, % por fila, filas más incompletas); (b) 3 estrategias: drop all, drop subset, imputar por grupo; (c) columna bill_was_missing y demuestra que el flag puede mejorar un modelo simple; (d) demo de dtypes nullable Int64.

Criterio de aceptación: Imputación por grupo no introduce sesgo; el flag was_missing añade información.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
if df['col'] == np.nan no funciona	NaN no es igual a nada (incluso a sí mismo
df.dropna() borra todo	Sin parámetros, borra fila con CUALQUIER N
Imputar con media global introduce sesgo	Si los grupos tienen medias muy distintas,
Una columna int se volvió float tras leer	Tiene NaN → promoción automática (NumPy in
fillna(0) mata el flag de "missing"	Pierdes la información de que faltaba. Fix

Preguntas frecuentes

¿Eliminar, imputar o flag?

Depende: <1% missing aleatorio → drop. 5-30% MAR → imputar (mediana por grupo). >50% → eliminar columna o tratar como categoría 'missing'. Si missing es informativo → flag + imputar.

¿Mediana o media para imputar?

Mediana es más robusta a outliers (recomendada por default). Media si la distribución es ~normal y sin outliers. Para categóricas: moda.

¿ffill siempre bueno para series temporales?

Bueno cuando los valores cambian poco entre observaciones (precios, temperaturas). Malo si los gaps son largos o el dato es volátil. Considera interpolate(method='time') para mejor resultado.

¿Cómo sé si la imputación afecta mi modelo?

Compara métricas: (a) baseline drop, (b) imputación X, (c) imputación + flag. Si el flag mejora el modelo, el missing era informativo (caso MNAR).

¿KNN/MICE para imputación en pandas?

No nativo. Usa sklearn.impute.KNNImputer o IterativeImputer (= MICE). Más caro pero mejor que mediana en datasets con correlaciones fuertes.

Referencias

- VanderPlas, cap. 3 § 3.5.
- pandas — Missing data user guide
- pandas nullable Integer dtypes

Siguiente clase

Clase 026 — Pandas: MultiIndex

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
import numpy as np
import pandas as pd
rng = np.random.default_rng(42)
```

Archivos complementarios

- notebook.ipynb