II MALACKATHON 2025 – HITO 3: ANÁLISIS EXPLORATORIO DE DATOS

Dataset: Ingresos hospitalarios por salud mental (RAE-CMBD)
Equipo: aprobados

1️⃣ Análisis descriptivo inicial

🔹 Resumen del dataset

El dataset contiene **21.210 registros** y **111 variables**, correspondientes a episodios de ingreso hospitalario por motivos de salud mental.
Tras el intento de parseo automático de fechas a partir del nombre de las columnas, los tipos detectados fueron:

- **Categóricas (object): 71 columnas**
- **Numéricas (int/float): 29 columnas**
- **Fechas (datetime): 11 columnas**

El volumen de datos y su estructura confirman la presencia de información clínica y administrativa detallada: fechas de ingreso y alta, diagnósticos principales y secundarios (CIE-10), procedimientos, variables derivadas del sistema GRD-APR, duración de la estancia, coste y severidad hospitalaria.

🔹 Metodología y herramientas empleadas

El análisis se realizó con **Python y la librería pandas**, y el pdf con Chat Gpt a partir de los datos encontrados. Se han usando las siguientes operaciones principales:

1. **Carga de datos:** lectura del fichero `SaludMental.xls` con el motor `xlrd`.
2. **Conversión de fechas:** intento automático de detección y parseo (`to_datetime`, `dayfirst=True`).
3. **Clasificación de variables** por tipo (`object`, `numeric`, `datetime`).

4. **Cálculo de estadísticos descriptivos** (`describe()`) para variables numéricas.
5. **Conteo de categorías** para variables cualitativas (`value_counts()`).
6. **Cálculo de valores nulos y porcentaje de nulos** por variable.
7. **Exportación de resultados** en los archivos:
    - `descriptivos_numericos.csv`
    - `resumen_variables.csv`
    - `categorias_preview.json`

🔹 Clasificación de variables (ejemplos)

Tipo	Ejemplos
Fechas	Fecha de nacimiento, Fecha de Ingreso, Fecha de Fin Contacto, Fecha de Intervención, Reingreso, Fecha de Inicio contacto, Ingreso en UCI, Edad en Ingreso, Mes de Ingreso
Numéricas	Edad, Estancia Días, GRD APR, CDM APR, Nivel Severidad APR, Riesgo Mortalidad APR, Coste APR, Días UCI, Régimen Financiación, Procedencia, Continuidad Asistencial
Categóricas	Comunidad Autónoma, Servicio, Sexo, Categoría, Diagnóstico Principal, POA diagnósticos, Procedimientos (códigos CIE-10), Centro Recodificado, País Residencia

---

🔹 Valores nulos o desconocidos

El análisis de la matriz de nulos muestra un alto porcentaje de campos vacíos en variables de procedimientos y derivados del agrupador hospitalario.
Las principales conclusiones fueron:

- **Columnas con 100% de nulos:**
  `CCAA Residencia`, `Fecha de Intervención`, `GDR AP`, `CDM AP`, `Tipo GDR AP`, `Valor Peso Español`, `Tipo GDR APR`, `Valor Peso Americano APR`, `Reingreso`, `GDR IR`, `Tipo GDR IR`, `Tipo PROCESO IR`, `Fecha de Inicio contacto`, `Procedimiento Externo 4-6`.

- **Columnas con ≥95% de nulos:**
  `Procedimientos 11–20`, `Procedimientos Externos 1–3`, `Diagnósticos 14–20` y sus respectivos marcadores `POA`.

- **Días UCI:** Solo **100 registros no nulos (~0,47%)**, con media **3,51 días** y máximo **38 días**.

Estas variables serán evaluadas en la fase de limpieza para decidir si se eliminan, imputan o agrupan.

🔹 Estadísticos descriptivos (variables numéricas clave)

Variable	Media	Desv.	Mín	Mediana	Máx
Edad	43,6 años	14,1	0	44	96
Estancia Días	15,5 días	19,9	0	11	814
GRD APR	751,3	33,6	4	752	952
CDM APR	18,9	0,95	0	19	24
Nivel Severidad APR	1,54	0,58	1	1	4
Riesgo Mortalidad APR	1,06	0,27	1	1	4
Coste APR (€)	5.453 €	1.562 €	1.496 €	5.988 €	70.601 €
Días UCI	3,5 días	5,5	0	2	38
Régimen Financiación	1,07	0,68	1	1	9
Procedencia	22,0	5,1	21	21	90

🔹 Variables categóricas destacadas

- **Comunidad Autónoma:**
  ANDALUCÍA (20.034 registros) y LA RIOJA (1.176).

- **Diagnóstico principal (CIE-10):**
  F20.0 (Esquizofrenia paranoide) – 4.573 casos
  F60.3 (Trastorno límite de la personalidad) – 1.372 casos
  F29, F31.2, F25.0 y F25.9 también entre los más frecuentes.

- **Categoría diagnóstica:**
    - Esquizofrenia y trastornos delirantes → 9.126
    - Trastornos del humor → 5.224
    - Trastornos de la personalidad → 3.248
    - Trastornos neuróticos/estrés → 2.082
    - Uso de sustancias → 744
    - Trastornos infantiles/adolescentes → 642

🔹 Outliers detectados

- **Estancia Días:** valores extremos de hasta **814 días**, indicando posibles estancias prolongadas o errores de registro.
- **Coste APR:** máximo **70.601 €**, significativamente superior al promedio.
- En la siguiente fase se aplicará **detección de outliers mediante el método IQR (1.5×IQR)** y **z-score** para variables numéricas clave (`Edad`, `Estancia Días`, `Coste APR`, `Días UCI`).

2️⃣ Ingeniería de características (avance)

Para futuras fases del proyecto se proponen las siguientes variables derivadas:

Nueva variable	Descripción	Método
Duración estancia	Validación y recalculo: `Fecha Fin − Fecha Ingreso`	Confirmar con `Estancia Días`
Grupo edad	Segmentación: `<18`, `18–35`, `36–60`, `>60`	Clasificación ordinal
Mes/Año de ingreso	Variable temporal para análisis estacional	`EXTRACT(MONTH/YEAR)`

Categoría diagnóstica CIE-10	Primeros 3 caracteres del código diagnóstico	Agrupar F20–F39
Indicador UCI	Variable binaria (1 si Días UCI > 0)	Derivada
Reingreso 30 días	Marcador de reingreso por paciente (actualmente nulo)	Requiere campo paciente ID

🔹 Conclusión

El análisis exploratorio permitió identificar la estructura, tipos de variables y calidad del dataset.
Existen variables muy completas (edad, diagnósticos, estancia, coste, severidad) junto con otras casi vacías (procedimientos y campos administrativos).
Estos hallazgos serán la base para la **limpieza de datos, detección de outliers e ingeniería de características** del siguiente hito.

Este es el código

```
# -*- coding: utf-8 -*-
import pandas as pd
from pathlib import Path

# --- Configuración de salida bonita ---
pd.set_option("display.max_rows", 200)
pd.set_option("display.max_columns", 200)
pd.set_option("display.width", 160)

# --- Cargar XLS (requiere xlrd instalado) ---
# pip install xlrd
df = pd.read_excel("SaludMental.xls", engine="xlrd")

print("\n=== INFO GENERAL ===")
print(df.info())
print("\n=== PRIMERAS FILAS ===")
print(df.head())

# --- Intento de detección/parseo de fechas (por nombre de columna) ---
posibles_fechas = [c for c in df.columns if any(k in c.lower() for k in
["fecha", "fec", "ingres", "alta", "nac", "interv"])]
for c in posibles_fechas:
    try:
        df[c] = pd.to_datetime(df[c], errors="coerce", dayfirst=True)
    except Exception:
        pass
```

```
# Recontar tipos tras convertir fechas
print("\n=== TIPOS DE DATO (tras parseo de fechas) ===")
print(df.dtypes.value_counts())

# --- Clasificación por tipo de dato ---
categoricas = df.select_dtypes(include="object").columns.tolist()
numericas   = df.select_dtypes(include=["int64",
"float64"]).columns.tolist()
fechas      =
df.select_dtypes(include="datetime64[ns]").columns.tolist()

print(f"\nVariables categóricas: {len(categoricas)}")
print(f"Variables numéricas:   {len(numericas)}")
print(f"Variables de fecha:    {len(fechas)}")

print("\nEjemplos de variables por tipo:")
print(" - Categóricas:", categoricas[:10])
print(" - Numéricas:  ", numericas[:10])
print(" - Fechas:     ", fechas[:10])

# --- Estadísticos descriptivos ---
print("\n=== DESCRIPTIVOS NUMÉRICOS ===")
desc_num = df[numericas].describe().T
print(desc_num)

print("\n=== DESCRIPTIVOS CATEGÓRICOS (conteo top 10) ===")
for col in categoricas[:5]:  # muestra 5 como ejemplo (puedes ampliar)
    vc = df[col].value_counts(dropna=False).head(10)
    print(f"\n-- {col} --")
    print(vc)

# --- Nulos y unicidad ---
summary = pd.DataFrame({
    "Tipo de dato": df.dtypes.astype(str),
    "Valores únicos": df.nunique(dropna=True),
    "Nulos": df.isna().sum(),
})
summary["% Nulos"] = (summary["Nulos"] / len(df) * 100).round(2)
summary = summary.sort_values(["% Nulos", "Nulos"], ascending=[False,
False])

print("\n=== RESUMEN POR VARIABLE (primeras 30 por % de nulos) ===")
```

```
print(summary.head(30))

# --- Guardar outputs para tu informe ---
outdir = Path("eda_outputs")
outdir.mkdir(exist_ok=True)

desc_num.to_csv(outdir / "descriptivos_numericos.csv",
encoding="utf-8", index=True)
summary.to_csv(outdir / "resumen_variables.csv", encoding="utf-8",
index=True)

# También útil: lista de categorías por columna (limitado para no
explotar)
cat_preview = {}
for col in categoricas:
    vals = df[col].dropna().unique()
    cat_preview[col] = vals[:20]  # primeras 20 categorías
pd.Series(cat_preview).to_json(outdir / "categorias_preview.json",
force_ascii=False)

print(f"\nArchivos generados en: {outdir.resolve()}")
print(" - descriptivos_numericos.csv")
print(" - resumen_variables.csv")
print(" - categorias_preview.json")
```