

MÓDULO: *Análisis de Datos con Python*



Data Analysis
with Python

PRÁCTICA GUIADA DE ANÁLISIS EXPLORATORIO DE DATOS (AED) CON PYTHON

Contexto: Esta práctica está diseñada para alumnado que inicia el módulo de análisis de datos en Python. Se trabajará paso a paso con un conjunto de datos reales sobre alojamientos turísticos en Barcelona, descargado de la plataforma Inside Airbnb.

Indicaciones

CREA UNA CARPETA: [PRACTICA0](#)

En ella descarga el archivo de datos .csv

En la misma carpeta deberás crear tus scripts .py

Para ejecutarlos, recuerda acceder a dicha carpeta desde el terminal: `cd practica0`

Una vez dentro ejecuta el script y comenta que hace cada uno de los comandos.

Ejemplo de ejecución del script `Importacion.py`:

```
(base) C:\Users\Usuario>cd practica0
(base) C:\Users\Usuario\Practica0>python Importacion.py
```

Contenido

0. Configuración e Importación de Librerías	2
1. Importación del Conjunto de Datos	2
2. Análisis Descriptivo Inicial	2
3. Visualización de Distribuciones.....	4
4. Ajuste de Tipos de Datos	4
5. Detección y Tratamiento de Datos Ausentes	4
6. Detección de Valores Atípicos	4
7. Análisis de Correlación	5
8. Generación de Informe Automatizado.....	5
Pasos para instalar la librería que nos falta de manera “manual”	5
Paso 1: Activar el entorno de Anaconda	5
Paso 2: Instalar ydata_profiling en el entorno base	7
Paso 3: Instalar ydata_profiling con pip	8
Paso 4: Verificar instalación.....	8
Extra: Guardar entorno como plantilla (environment.yml).....	9
9. Tarea Final	9

Deberás realizar los siguientes apartados

0. Configuración e Importación de Librerías

```
# Librerías fundamentales para análisis de datos
import pandas as pd          # Manipulación de datos en formato tabla
import numpy as np           # Cálculos numéricos y estadísticos
import matplotlib.pyplot as plt # Visualización básica (gráficos)
import seaborn as sns        # Visualización avanzada y estética
import os                    # Operaciones con archivos y rutas
```

1. Importación del Conjunto de Datos

<https://insideairbnb.com/get-the-data/>

Por ejemplo:

12 de junio de 2025 ([Explorar](#))

Pais/Ciudad	Nombre del archivo	Descripción
Barcelona	listados.csv.gz	Datos detallados de listados
Barcelona	calendario.csv.gz	Datos detallados del calendario
Barcelona	reseñas.csv.gz	Datos de revisión detallados
Barcelona	listados.csv	Información resumida y métricas de listados en Barcelona (útil para visualizaciones).
Barcelona	reseñas.csv	Resumen Revisar datos e ID de listado (para facilitar análisis basados en el tiempo y visualizaciones vinculadas a un listado).
Barcelona	barrios.csv	Lista de barrios para filtro geográfico. Procedente de archivos SIG de la ciudad o de código abierto.
Barcelona	barrios.geojson	Archivo GeoJSON de barrios de la ciudad.

Cargar el archivo listings.csv descargado de Inside Airbnb

Asegúrate de que el archivo esté en la misma carpeta que este archivo .py

```
# Cargar el archivo listings.csv descargado de Inside Airbnb
# Asegúrate de que el archivo esté en la misma carpeta que tu archivo .py
df = pd.read_csv('listings.csv')
```

2. Análisis Descriptivo Inicial

```
# Vista preliminar de los datos
print(df.head(2))
print("="*100)

# Información general del DataFrame
print(df.info())
print("="*100)

# Estadísticas descriptivas de las variables numéricas
print(df.describe())
print("="*100)
```

Por ejemplo, para la vista preliminar de datos,

```
# Vista preliminar de los datos
print(df.head(2))
print("="*100)
```

una vez ejecutado el script, se mostrará:

```
(base) C:\Users\Usuario>cd practica0
(base) C:\Users\Usuario\Practica0>python Importacion.py
      id      name  ...  number_of_reviews_ltm  license
0  18674  Huge flat for 8 people close to Sagrada Familia  ...      6  HUTB-002062
1  23197  Forum CCIB DeLuxe, Spacious, Large Balcony, relax  ...     10  HUTB005057

[2 rows x 18 columns]
=====
```

Explicación de lo que hace dicho código:

```
print(df.head(2))
```

```
print("="*100)
```

1. **print(df.head(2))**

- Muestra por pantalla las **primeras dos filas** del DataFrame df.
- Es útil para **inspeccionar rápidamente** cómo están organizados los datos, qué columnas hay, qué tipo de valores aparecen, y si la carga del archivo fue correcta.
- El número 2 puede cambiarse por cualquier cantidad de filas que quieras visualizar.

2. **print("="*100)**

- Imprime una línea compuesta por 100 signos de igual (=).
- Se usa como **separador visual** para que los resultados en la consola estén más ordenados y legibles.
- Es especialmente útil cuando se muestran varios bloques de información seguidos (como info(), describe(), etc.).

IMPORTANTE: De este modo deberás explicar tu práctica, realizando capturas del resultado de la ejecución de tu script y explicando detalladamente que realizan cada una de las funciones.

3. Visualización de Distribuciones

```
# Selección de columnas numéricas
columnas_numericas = df.select_dtypes(include=[np.number]).columns

# Visualización con histogramas
n = len(columnas_numericas)
nrows = 3
ncols = min(n, 3)

fig, axes = plt.subplots(nrows=nrows, ncols=ncols, figsize=(15, 5*nrows))
fig.suptitle('Distribución de Variables Numéricas', fontsize=16)
axes = axes.flatten()

for i, col in enumerate(columnas_numericas[:len(axes)]):
    ax = axes[i]
    df[col].hist(ax=ax, bins=50, edgecolor='black')
    ax.set_title(f'Distribución de {col}')
    ax.set_xlabel(col)
    ax.set_ylabel('Frecuencia')

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

4. Ajuste de Tipos de Datos

```
# Conversión de variables categóricas
df['neighbourhood'] = df['neighbourhood'].astype('category')
df['room_type'] = df['room_type'].astype('category')
```

- 'neighbourhood': representa el barrio donde se encuentra el alojamiento. Convertirlo a tipo **category** permite ahorrar memoria y facilita análisis agrupados.
- 'room_type': indica el tipo de alojamiento (por ejemplo, habitación privada o vivienda completa). También se convierte a **category**.

5. Detección y Tratamiento de Datos Ausentes

```
# Conteo de valores nulos
print("Valores perdidos por columna:")
print(df.isna().sum())

# Porcentaje de valores perdidos
print("Porcentaje de valores perdidos:")
print(df.isna().mean().round(2))

# Eliminación de columnas con más del 50% de valores nulos
df = df.loc[:, df.isna().mean() < 0.5]

# Imputación de valores numéricos con la media
columnas_numericas_limpias = df.select_dtypes(include=[np.number]).columns
df[columnas_numericas_limpias] = df[columnas_numericas_limpias].fillna(df[columnas_numericas_limpias].mean())
```

6. Detección de Valores Atípicos

Ejemplo con la variable price:

```
# Cálculo de límites IQR
Q1 = df['price'].quantile(0.25)
Q3 = df['price'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Visualización con boxplot
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['price'])
plt.title('Boxplot de precios con outliers')
plt.show()

# Filtrado de datos sin outliers

df_sin_outliers = df[(df['price'] >= lower_bound) & (df['price'] <= upper_bound)]
```

7. Análisis de Correlación

```
# Matriz de correlación
correlacion = df.select_dtypes(include=[np.number]).corr()

# Mapa de calor
plt.figure(figsize=(10, 8))
sns.heatmap(correlacion, annot=True, cmap='coolwarm', center=0, square=True, linewidths=.5, cbar_kws={"shrink": .5})
plt.title('Matriz de correlaciones entre variables numéricas')
plt.show()
```

8. Generación de Informe Automatizado

```
# 8. Generación de Informe Automatizado

# Generación de informe con YData Profiling
from ydata_profiling import ProfileReport

reporte = ProfileReport(df, title='Informe EDA Airbnb Barcelona')
reporte.to_file('reporte_airbnb_barcelona.html')
```

Nota: Puede ocurrir que nos aparezca el siguiente error:

```
ModuleNotFoundError: No module named 'ydata_profiling'
```

Esto significa que la librería ydata_profiling no está instalada en tu entorno de Anaconda. Es una herramienta muy útil para generar informes automáticos de análisis exploratorio, pero hay que instalarla manualmente.

Pasos para instalar la librería que nos falta de manera “manual”

Paso 1: Activar el entorno de Anaconda

Abre Anaconda Prompt y activa el entorno donde quieres instalar la librería. Por ejemplo:

```
conda activate mi_entorno
```

(Sustituye mi_entorno por el nombre real de tu entorno.)

¿Cómo conozco el nombre real de mie entorno en caso de duda?

Ver entornos disponibles en Anaconda

Opción 1: Usar Anaconda Prompt

Abre **Anaconda Prompt** y escribe:

```
conda env list
```

Esto mostrará una lista como:

```
# conda environments:
#
base          * C:\Users\TuUsuario\anaconda3
mi_entorno    C:\Users\TuUsuario\anaconda3\envs\mi_entorno
analisis_datos C:\Users\TuUsuario\anaconda3\envs\analisis_datos
```

- El entorno marcado con * es el que está activo.

Puedes activar otro entorno con:

```
conda activate nombre_del_entorno
```

En mi caso:

```
Microsoft Windows [Versión 10.0.26100.6584]
(c) Microsoft Corporation. Todos los derechos reservados

(base) C:\Users\Usuario>conda env list

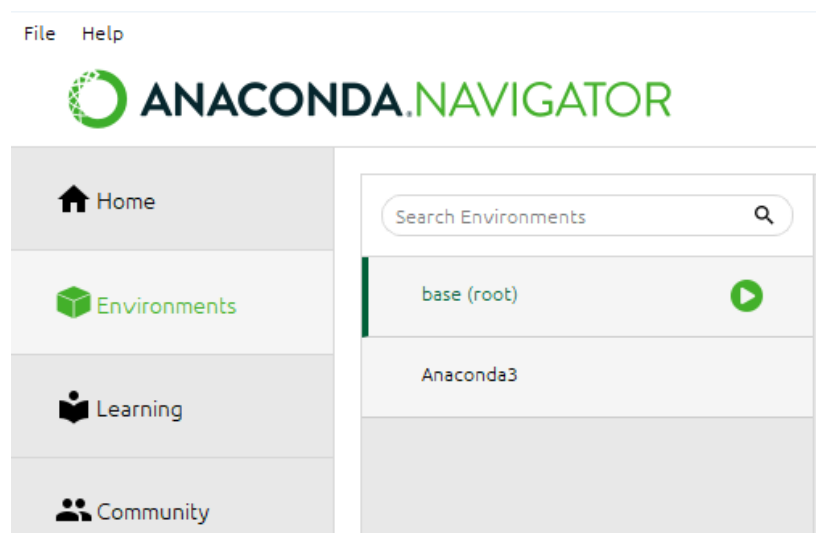
# conda environments:
#
base          C:\Users\Usuario\Anaconda3
              * C:\Users\Usuario\anaconda3

(base) C:\Users\Usuario>
```

Esto significa que estoy trabajando directamente en el entorno base, que se llama simplemente base.

Opción 2: Desde Anaconda Navigator

1. Abre **Anaconda Navigator**.
2. En la parte izquierda, haz clic en **Environments**.
3. Verás todos los entornos disponibles con sus nombres.
4. Puedes seleccionar y activar el que necesites desde ahí.



Por lo tanto, en mi caso deberé activar mi entorno de la siguiente manera:

```
conda activate base
```

En tu caso, hecho esto en tu entorno, verás que el prompt cambia a algo como:

```
(base) C:\Users\Usuario>
```

Paso 2: Instalar ydata_profiling en el entorno base

Una vez activado, instala la librería con:

```
pip install ydata-profiling
```

Si prefieres mantener tus proyectos organizados, puede ser útil crear un entorno nuevo llamado, por ejemplo, analisis_datos, con todas las dependencias necesarias para nuestro módulo.

Crear entorno Conda: analisis_datos

Paso 1: Crear el entorno

Abre **Anaconda Prompt** y ejecuta:

```
conda create -n analisis_datos python=3.10
```

Esto crea un entorno llamado analisis_datos con Python 3.10 (compatible con ydata_profiling y pandas).

Si lo has realizado correctamente indica cómo activar o desactivar el entorno:

```
Downloading and Extracting Packages:
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate analisis_datos
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:\Users\Usuario>
```

Paso 2: Activar el entorno

```
conda activate analisis_datos
```

```
(base) C:\Users\Usuario>conda activate analisis_datos
(analisis_datos) C:\Users\Usuario>
```

Paso 3: Instalar dependencias clave

```
conda install jupyter pandas matplotlib seaborn scikit-learn
```

Estas son las herramientas básicas para análisis de datos y visualización.

Puede tardar un poco la descarga e instalación, pero si todo está correcto aparecerá:

```
Downloading and Extracting Packages:
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(analisis_datos) C:\Users\Usuario>
```



```
pip install ydata-profiling
```

Esta librería no está en los canales Conda, por eso se instala con pip.

Paso 4: Verificar instalación

Abre Jupyter Notebook con:

jupyter notebook

Y prueba este código:

```
# 8. Generación de Informe Automatizado

# Cargar el archivo listings.csv descargado de Inside Airbnb
# Asegúrate de que el archivo esté en la misma carpeta que este script

import pandas as pd
from ydata_profiling import ProfileReport

# Leer el archivo CSV
df = pd.read_csv("listings.csv") # Asegúrate de que el archivo exista en la carpeta actual

# Generar el informe
profile = ProfileReport(df, title="Informe EDA Airbnb Barcelona", explorative=True)

# Exportar el informe a HTML
profile.to_file("reporte_airbnb_barcelona.html")
```

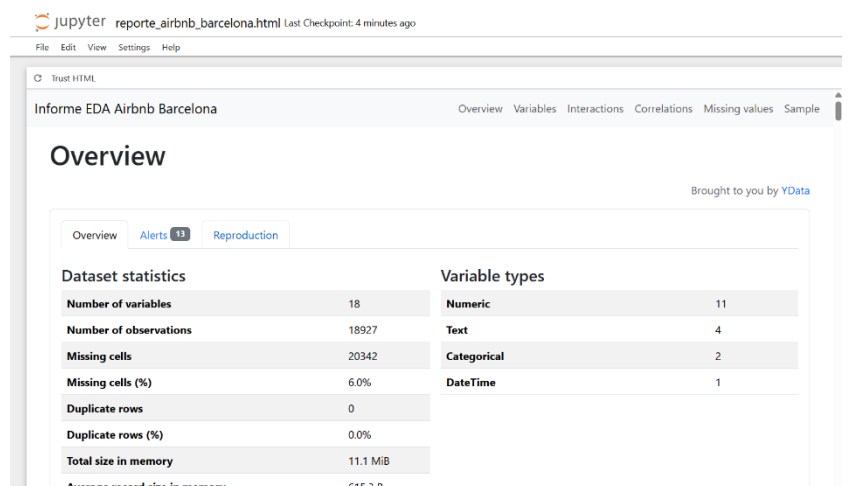
Para ejecutarlo.

```
(analisis_datos) C:\Users\Usuario\Practica0>python informe2.py
```

El resultado de la ejecución es el informe:

```
(analisis_datos) C:\Users\Usuario\Practica0>python informe2.py  
Upgrade to ydata-sdk  
Improve your data and profiling with ydata-sdk, featuring data quality scoring, redundancy detection, outlier identifica-  
tion, text validation, and synthetic data generation.  
Register at https://ydata.ai/register  
100%|██████████████████████████████████████████████████████████████████████████| 18/18 [00:00<00:00, 27.30it/s]  
Summarize dataset: 100%|███████████████████████████████████████████████████████████████████████████| 149/149 [00:40<00:00, 3.72it/s, Completed]  
Generate report structure: 100%|███████████████████████████████████████████████████████████████████████████| 1/1 [00:21<00:00, 21.94s/it]  
Render HTML: 100%|███████████████████████████████████████████████████████████████████████████████████| 1/1 [00:02<00:00, 2.18s/it]  
Export report to file: 100%|███████████████████████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, 11.99it/s]  
  
(analisis_datos) C:\Users\Usuario\Practica0>
```

El archivo `reporte_airbnb_barcelona.html` se ha guardado en la misma carpeta donde está tu script (`practica0`). Puedes abrirlo haciendo doble clic sobre él o arrastrándolo a un navegador.



TODO ESTO LO VEREMOS CON DETALLE EN
NUESTRO MÓDULO, PASO A PASO Y
DESPACIO!!

Extra: Guardar entorno como plantilla (environment.yml)

Si quieres replicar este entorno en otros equipos, puedo generar el archivo environment.yml con todas las dependencias incluidas.

Crear archivo environment.yml desde tu entorno analisis_datos

Paso 1: Activar el entorno

Abre **Anaconda Prompt** y ejecuta:

```
conda activate analisis_datos
```

Paso 2: Exportar el entorno a un archivo .yml

```
conda env export > environment.yml
```

Esto crea el archivo environment.yml en la carpeta actual (por ejemplo, C:\Users\Usuario si no has cambiado de directorio).

Paso 3: Mover el archivo si lo necesitas

Puedes mover el archivo a una carpeta de proyecto, como FP_Analisis_Datos, para mantenerlo organizado:

```
move environment.yml practica0\
```

¿Cómo usar este archivo en otro equipo?

En otro ordenador con Anaconda instalado, basta con ejecutar:

```
conda env create -f environment.yml
```

Esto creará un entorno idéntico al tuyo, con todas las librerías necesarias para ejecutar tus prácticas.

9. Tarea Final

Una vez completada la práctica, cada estudiante deberá:

- Subir un tutorial en PDF con los pasos que ha seguido.
- Incluir capturas de pantalla de los resultados obtenidos. Es mejor realizar uno a uno cada script.
- Explicar brevemente qué has aprendido en cada apartado.