



Dashboard

Career Path

Forms

Profile

Change Status

# Introducción al Análisis Exploratorio (EDA) - Día 1

in-progress

40 min

## Learning Objectives

- 1 Comprender la filosofía fundamental del análisis exploratorio de datos como aproximación científica
- 2 Distinguir entre EDA y análisis confirmatorio en la práctica profesional
- 3 Establecer una metodología estructurada para abordar datasets nuevos de manera sistemática

Theory

Practice

Quiz

Evidence

## \*\*Ejercicio\*\*: Primer EDA en dataset de ventas de e-commerce

Ejercicio práctico para aplicar los conceptos aprendidos.

### Carga y configuración inicial:

```
import pandas as pd
import numpy as np

# Crear dataset de ejemplo de e-commerce
np.random.seed(42)
n_orders = 1000

df = pd.DataFrame({
    'order_id': range(1, n_orders + 1),
    'customer_id': np.random.randint(1, 201, n_orders),
    'product_id': np.random.randint(1, 51, n_orders),
    'quantity': np.random.randint(1, 5, n_orders),
    'unit_price': np.round(np.random.uniform(10, 500, n_orders), 2),
    'order_date': pd.date_range('2023-01-01', periods=n_orders, freq='H')[:n_orders],
    'payment_method': np.random.choice(['Credit Card', 'Debit Card', 'PayPal', 'Cash'], n_orders),
    'customer_age': np.random.normal(35, 10, n_orders).clip(18, 80).astype(int),
    'shipping_region': np.random.choice(['North', 'South', 'East', 'West'], n_orders)
```

us English



Sign Out

Toggle theme: Light Theme





Dashboard

Career Path

Forms

Profile

})

```
# Introducir algunos valores faltantes
mask = np.random.random(n_orders) < 0.05
df.loc[mask, 'customer_age'] = np.nan

print("Dataset cargado exitosamente")
```

### Inspección inicial sistemática:

```
# Dimensiones y estructura
print(f"Dataset shape: {df.shape}")
print(f"\nColumnas: {list(df.columns)}")
print(f"\nTipos de datos:\n{df.dtypes}")

# Primeras y últimas filas
print("\nPrimeras 5 filas:")
print(df.head())

print("\nÚltimas 5 filas:")
print(df.tail())
```

### Análisis de calidad de datos:

```
# Valores faltantes
print("Valores faltantes por columna:")
print(df.isnull().sum())

print(f"\nPorcentaje de completitud: {(1 - df.isnull().sum() / len(df)) * 100}"))

# Valores únicos por columna
print("\nValores únicos por columna:")
for col in df.select_dtypes(include=['object']).columns:
    print(f"{col}: {df[col].nunique()} valores únicos")

# Estadísticos básicos para numéricas
print("\nEstadísticos básicos de variables numéricas:")
print(df.select_dtypes(include=[np.number]).describe())
```



[→] Sign Out



### Preguntas exploratorias iniciales:

[Dashboard](#)[Career Path](#)[Forms](#)[Profile](#)

```
# Distribución por región
print("Distribución de pedidos por región:")
print(df['shipping_region'].value_counts())

# Método de pago más popular
print(f"\nMétodo de pago más usado: {df['payment_method'].value_counts().index[0]}")

# Rango de fechas
print(f"\nPeríodo de datos: {df['order_date'].min()} a {df['order_date'].max()}")

# Edad promedio de clientes
edad_promedio = df['customer_age'].mean()
print(f"\nEdad promedio de clientes: {edad_promedio:.1f} años")
```

**Verificación:** Confirma que has identificado las dimensiones del dataset, tipos de datos, valores faltantes, y generado preguntas iniciales basadas en los datos observados.

**Requerimientos:**

- Python con Pandas y NumPy instalados
- Jupyter Notebook para exploración interactiva
- Dataset de ejemplo o datos para analizar
- matplotlib opcional para primeras visualizaciones

[Sign Out](#)