UNIVERSIDAD CATÓLICA DE PEREIRA

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

TÍTULO DE LA ACTIVIDAD

TALLER EDA

NOMBRES ESTUDIANTES

SEBASTIAN OBANDO

Análisis Explorado de Datos (EDA) usando el Conjunto de Datos Iris

Este documento describe detalladamente el proceso de realizar un Análisis Exploratorio de Datos (EDA) con el conjunto de datos 'Iris', que incluye mediciones de tres variedades de flores Iris. Se empleó la plataforma Google Colab y las bibliotecas de Python: pandas, seaborn y matplotlib.

Iniciar el Proyecto

Google Colab o https://github.com/Sebas19981/EDA_Iris_Colab.git

Importar bibliotecas

Importar **librerías**

```
[2]  import pandas as pd
     import seaborn as sns
     import matplotlib.pyplot as plt
```

Se traen las bibliotecas esenciales para la manipulación de datos (pandas), la visualización (seaborn, matplotlib) y el análisis.

Cargar el conjunto de datos Iris

Cargar el dataset Iris

```
[4]  df = sns.load_dataset("iris")
```

```
[5]  print(df.head())        # Ver primeras filas
     print(df.info())        # Info de columnas y tipos
     print(df.describe())    # Estadísticas
```

```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
```

Se incorpora el conjunto de datos Iris, que ya está integrado en seaborn. No es preciso descargarlo de forma manual.

Se examinan las primeras filas, la composición de los datos y las estadísticas descriptivas como la media y la desviación estándar.

Comprobar valores ausentes

Verificar valores faltantes

```
[6]  print(df.isnull().sum())  # Ver cuántos valores nulos hay por columna
```

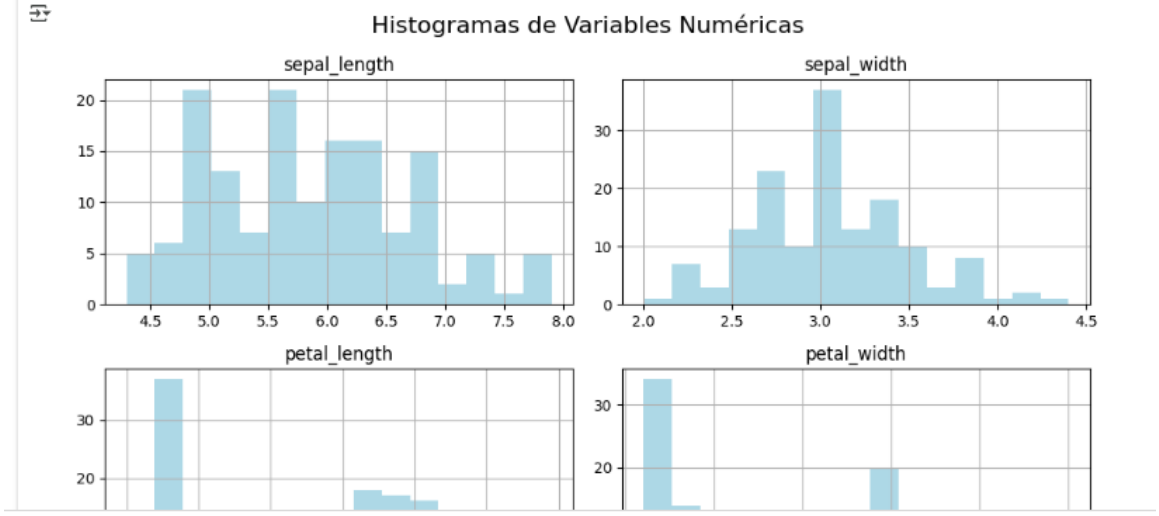
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

Se investiga si existen valores nulos en el conjunto de datos. En este caso, el conjunto de datos Iris está completo, sin valores perdidos.

Generación de histogramas

Histogramas

```
df.hist(figsize=(10, 6), bins=15, color='lightblue')
plt.suptitle("Histogramas de Variables Numéricas", fontsize=16)
plt.tight_layout()
plt.show()
```



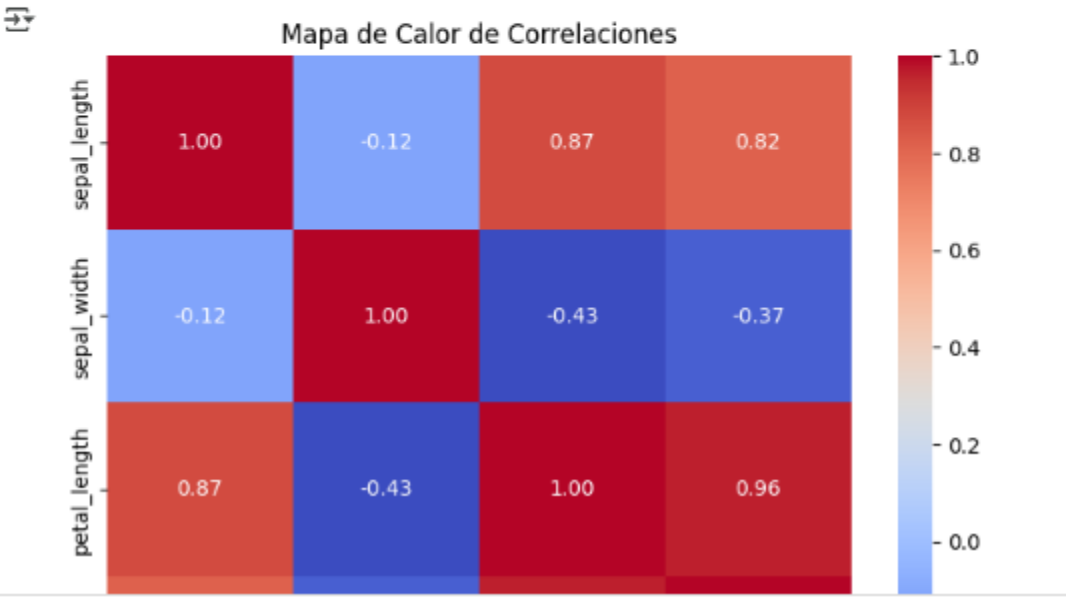
Se elaboran histogramas para cada variable numérica, lo que permite analizar la distribución de los datos.

Mapa de calor

Mapa de calor (Heatmap)

```
[9]  # Seleccionar solo columnas numéricas
     numeric_df = df.select_dtypes(include=["float64", "int64"])

     # Crear mapa de calor
     plt.figure(figsize=(8, 6))
     sns.heatmap(numeric_df.corr(), annot=True, cmap="coolwarm", fmt=".2f")
     plt.title("Mapa de Calor de Correlaciones")
     plt.show()
```

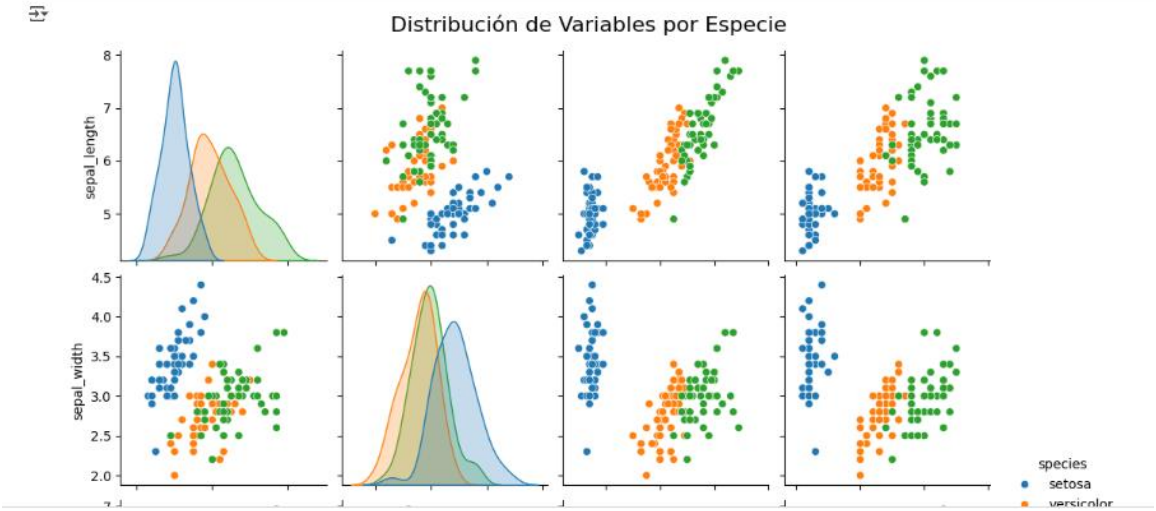


Se confecciona una matriz de correlaciones entre las variables numéricas. Esta matriz se presenta mediante un mapa de calor utilizando seaborn.

Identificación de patrones

Detectar patrones (gráfico de dispersión)

```
[10] sns.pairplot(df, hue="species")
     plt.suptitle("Distribución de Variables por Especie", fontsize=16, y=1.02)
     plt.show()
```



Se utiliza un gráfico de dispersión múltiple (pairplot) para examinar las relaciones entre las variables en función de la especie de flor.