Universidad Nacional ARTURO JAURETCHE

DIPLOMATURA SUPERIOR EN INTELIGENCIA ARTIFICIAL

TICAPPS | UNAJ
Programa Tecnologías de la Información y la Comunicación (TICs) en Aplicaciones de Interés Social

PREPROCESAMIENTO DE DATOS
CLASES 2

AGENDA

1. VISUALIZACIÓN

1. VISUALIZACIÓN

1. VISUALIZACIÓN.

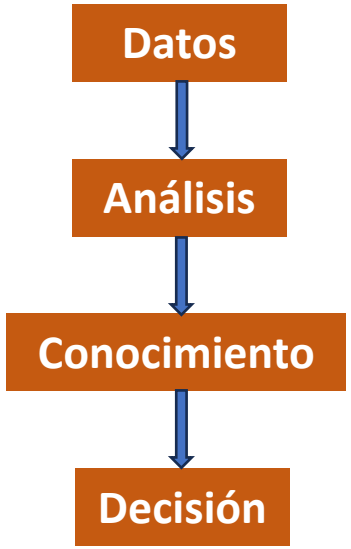
CONTENIDOS:

- CONCEPTOS DE ESTADÍSTICA
- ESTADÍSTICA DESCRIPTIVA
- ESTADÍSTICA DESCRIPTIVA. DESCRIPCIÓN NUMÉRICA
- ESTADÍSTICA DESCRIPTIVA. DESCRIPCIÓN GRÁFICA
- VISUALIZACIONES EN PYTHON
- VISUALIZACIONES CON EL DATASET IRIS

1.1. Conceptos de Estadística

¿Qué es la Estadística?

- **El campo de la Estadística** tiene que ver con la **recopilación, organización, análisis y uso de datos para tomar decisiones** razonables basadas en tal análisis.
- **Al recoger datos relativos a las características de un grupo de individuos** u objetos, suele ser **imposible o poco práctico observar todo el grupo** completo, en especial si es muy grande.
- En general, **en lugar de analizar el conjunto entero** llamado **población** o universo, **se examina una pequeña parte** del mismo, llamada **muestra**.



1.1. Conceptos de Estadística

Definiciones

- Una ***población*** está **formada por la totalidad de las observaciones** en las cuales se tiene cierto interés.
- Un ***individuo*** **es cada uno de los elementos** de la población.
- Una ***muestra*** es un **subconjunto de observaciones** seleccionada de una población.
- Una ***muestra aleatoria*** es **una muestra cuyos elementos son elegidos al azar**.
- Una ***variable aleatoria*** es una **función que asigna un valor numérico al resultado de un experimento** aleatorio.

1.1. Conceptos de Estadística

Clasificación de la Estadística

- La **Estadística** se divide en **dos grandes ramas** según el propósito del análisis:

❑ ***Estadística Descriptiva:***

Se ocupa de **organizar, resumir y presentar los datos de manera comprensible**. Utiliza **tablas, gráficos y medidas numéricas** (como media, mediana, moda, varianza, desviación estándar, etc.) **para describir las características** de un **conjunto de datos**.

❑ ***Estadística Inferencial:***

Permite **sacar conclusiones o generalizaciones sobre una población** a partir de la información obtenida en una muestra. Emplea **probabilidad, estimación y pruebas de hipótesis** para **inferir o predecir comportamientos** o parámetros poblacionales.

1.1. Conceptos de Estadística

¿Cómo se relaciona la Estadística con las aplicaciones en IA?

La **Estadística nos permite**:

- **entender la estructura y variabilidad** de los datos.
- **seleccionar variables relevantes**.
- **evaluar el rendimiento y la validez** de **modelos**.

Toda predicción **o modelo parte de una descripción estadística** sólida.

1.2. Estadística Descriptiva

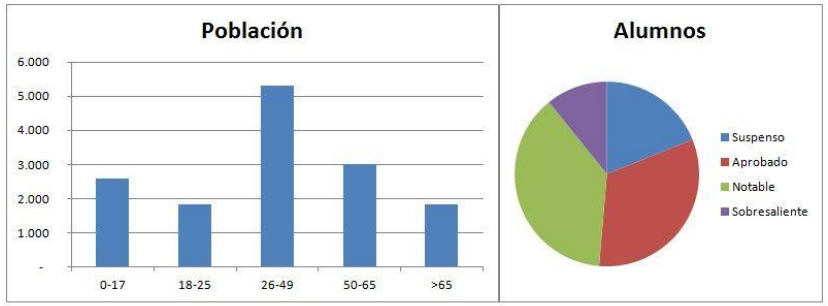
Estadística Descriptiva

- Describe y **organiza los datos mediante representaciones numéricas, gráficas o tablas**.

- En **IA se usa en la exploración de datos**.

- Las **herramientas descriptivas básicas** que se utilizan son:

  - **Descripción numérica.**
  - **Descripción gráfica.**



Edad	Población
0-17	2.592
18-25	1.834
26-49	5.314
50-65	3.012
>65	1.839

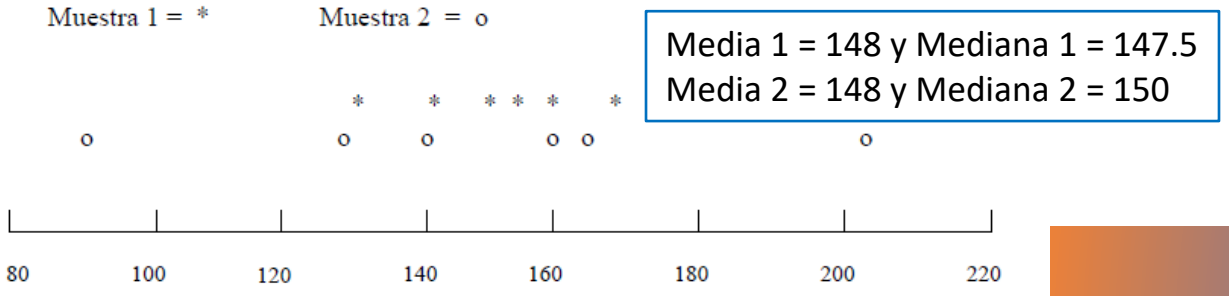
Nota	Alumnos
Suspenso	7
Aprobado	12
Notable	14
Sobresaliente	4

1.3. Estadística Descriptiva - Descripción Numérica

Medidas de posición y de dispersión

Se distinguen **dos tipos de medidas**: **de posición y de dispersión** para **describir la información contenida en una muestra** $x_1, x_2, \ldots, x_n$ de tamaño n.

- Las **medidas de posición proporcionan valores alrededor de los cuales se distribuyen los datos** observados en la muestra.
- Se distinguen **medidas de posición de tendencia central** (media, mediana, moda) y **de tendencia no central** (cuartiles, percentiles).
- Las **medidas de dispersión muestran la variabilidad de los datos**, indicando **la mayor o menor concentración de datos respecto a las medias de centralización**. Entre ellas se encuentran: rango, rango intercuartílico (IQR), desviación estándar.



Media 1 = 148 y Mediana 1 = 147.5
Media 2 = 148 y Mediana 2 = 150

Muestra 1: 130, 150, 145, 158, 165, 140

Muestra 2: 90, 128, 205, 140, 165, 160

1.3. Estadística Descriptiva. Descripción Numérica

Medidas de posición de tendencia central:

- **Media aritmética o media muestral** $(\bar{x})$

$$\bar{x} = \frac{x_1 + \ldots + x_n}{n} = \frac{\sum_{i=1}^{n} x_i}{n}.$$

- ✓ La **media aritmética** es el **valor promedio de la muestra**.
- ✓ No tiene por qué pertenecer al conjunto de posibles valores de la variable.
- ✓ **No es una medida robusta**, es decir, **su valor se ve influenciado** por datos anormalmente altos o bajos (**valores atípicos**).

1.3. Estadística Descriptiva. Descripción Numérica

Medidas de posición de tendencia central:

- **Mediana** $(\tilde{x})$

✓ Si los **datos de la muestra están ordenados** de menor a mayor, la **mediana** es el **valor hasta el cual se encuentran el 50 % de los casos**. Por tanto, la mediana dejará la mitad de las observaciones por debajo de su valor y la otra mitad por encima.

✓ A diferencia de la media, **la mediana es una medida robusta**, ya que su valor se ve **poco afectado** por la presencia **de datos atípicos**.

✓ **Si de una muestra se obtienen la media y la mediana y sus valores difieren sustancialmente, esto será indicativo de la presencia de datos atípicos.**

- **Moda**

✓ La **moda** es la **observación que más se repite en la muestra**. Puede existir más de una moda.

1.3. Estadística Descriptiva. Descripción Numérica

Medidas de posición de tendencia no central:

- **Cuartiles: Si los datos de la muestra están ordenados de menor a mayor**

✓ **Los cuartiles Q1, Q2 y Q3 dividen la muestra en cuatro partes iguales.**

✓ El primer cuartil o cuartil inferior (Q1) es un valor que tiene la cuarta parte (25%) de las observaciones por debajo de él. El segundo cuartil (Q2), tiene la mitad (50%) de las observaciones por debajo de él. El segundo cuartil coincide con la mediana. El tercer cuartil o cuartil superior (Q3), tiene las tres cuartas partes (75%) de las observaciones por debajo de él.

- **Percentiles: Si los datos de la muestra están ordenados de menor a mayor**

✓ **y se los divide en cien partes iguales, los puntos de división reciben el nombre de percentiles**.

✓ Puede verse que $p_{0.25} = q_1$, $p_{0.5} = q_2$, $p_{0.75} = q_3$.

1.3. Estadística Descriptiva. Descripción Numérica

Medidas de dispersión:

- **Rango muestral** $R = \text{máx}\{x_i\} - \text{mín}\{x_i\}$

✓ **Diferencia entre la observación más grande y la más pequeña**. Puede verse afectado por la presencia de datos atípicos. A mayor rango, mayor variabilidad en los datos.

- **Rango intercuartílico (IQR)** $IQR = Q3 - Q1$

✓ **Diferencia entre el cuartil superior e inferior**. El rango intercuartílico es menos sensible a los valores extremos de la muestra que el rango muestral.

- **Desviación estándar muestral**

✓ Es el **promedio de las desviaciones de los datos respecto a la media**.

$$s = +\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

1.3. Estadística Descriptiva. Descripción Numérica

En Python:

Las **principales medidas numéricas** pueden obtenerse con la función **describe** de la librería **pandas**. Esta permite obtener un **resumen estadístico rápido** de un **DataFrame o Serie**.

```
1 datos_mios.describe()
```

	personas.pre	personas.post
count	5.000000	5.000000
mean	11.600000	6.200000
std	7.300685	8.408329
min	5.000000	1.000000
25%	8.000000	2.000000
50%	10.000000	2.000000
75%	11.000000	5.000000
max	24.000000	21.000000

1.4. Estadística Descriptiva. Descripción Gráfica

Visualizaciones

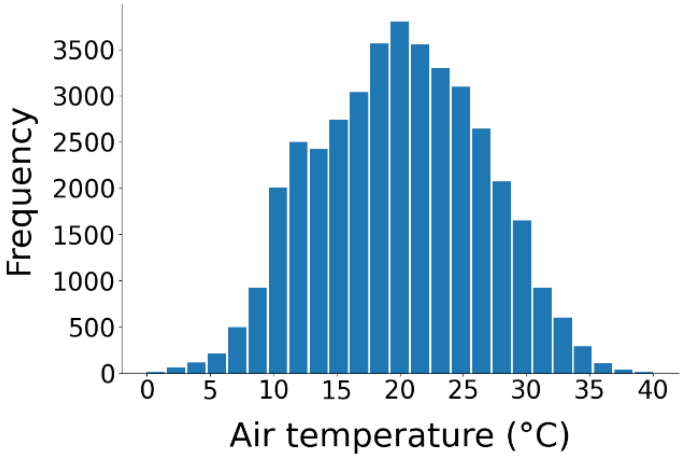
Otra manera de describir la información presente en dat**os es la gráfica**. **Existen muchos tipos de gráfico**s en la Estadística Descriptiva. Los más comunes son:

- **Histogramas**
- **Diagramas de dispersión**
- **Diagramas de cajas y bigotes**

1.4. Estadística Descriptiva. Descripción Gráfica

Histograma:

- Es una **representación de la distribución de los datos**.
- Permite **visualizar dónde están situados los datos** y tener una idea de su distribución.
- Para su construcción se debe **partir el rango de valores de la variable** (valor mínimo y máximo) **en n intervalos** (bins) iguales (no necesariamente). Luego, se debe **contar cuántos datos caen en cada intervalo**.
- **La base de cada rectángulo es el intervalo** de clase y la **altura es el número de muestras** en ese intervalo (frecuencia).

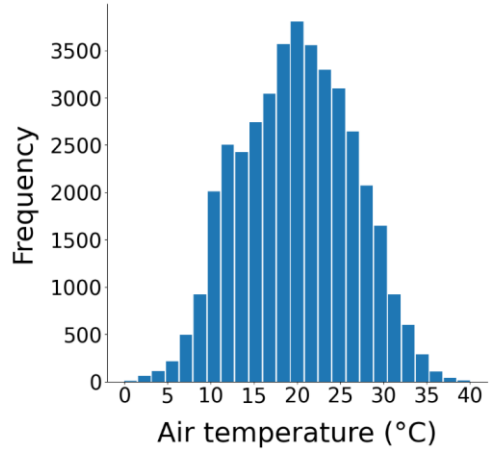


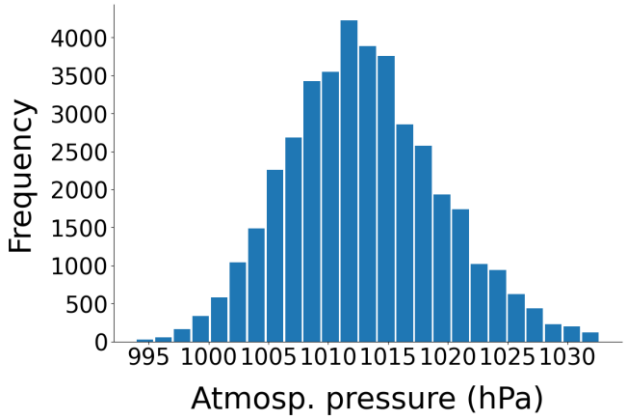
En Python:

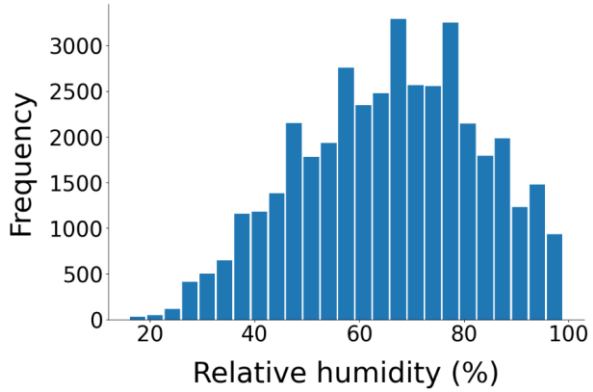
```
data.hist()
```

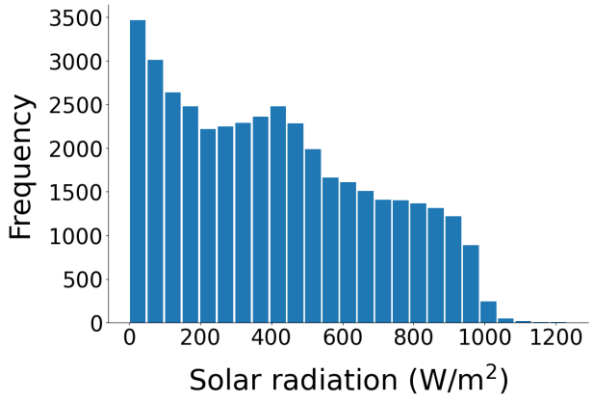
1.4. Estadística Descriptiva. Descripción Gráfica

Histograma:





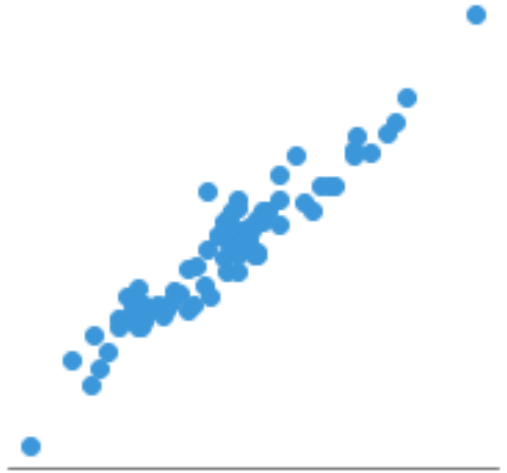




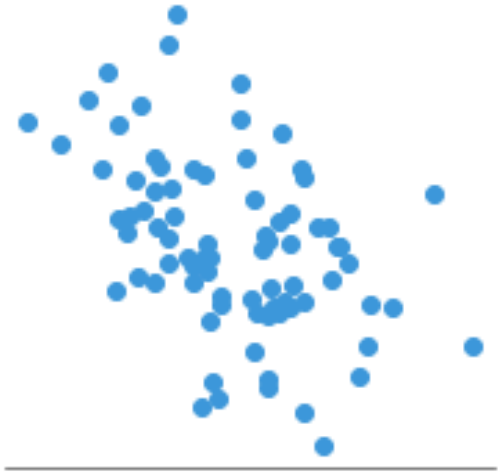
1.4. Estadística Descriptiva. Descripción Gráfica

Diagrama de dispersión (scatter plot):

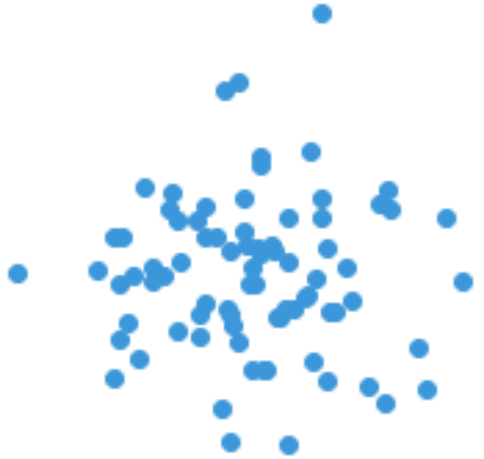
Es una herramienta gráfica que representa la relación entre dos variables numéricas para mostrar si existe una **correlación** entre ellas.



Fuerte, positiva, lineal



Moderada, negativa, lineal



Sin relación

Fuerte, no lineal

En Python:

```
plt.scatter(datosPos['tiempo'],
            datosPos['posicion'], s=20, c='blue')
```

1.4. Estadística Descriptiva. Descripción Gráfica

$$IQR = Q3 - Q1$$

Diagrama de cajas y bigotes (boxplot):

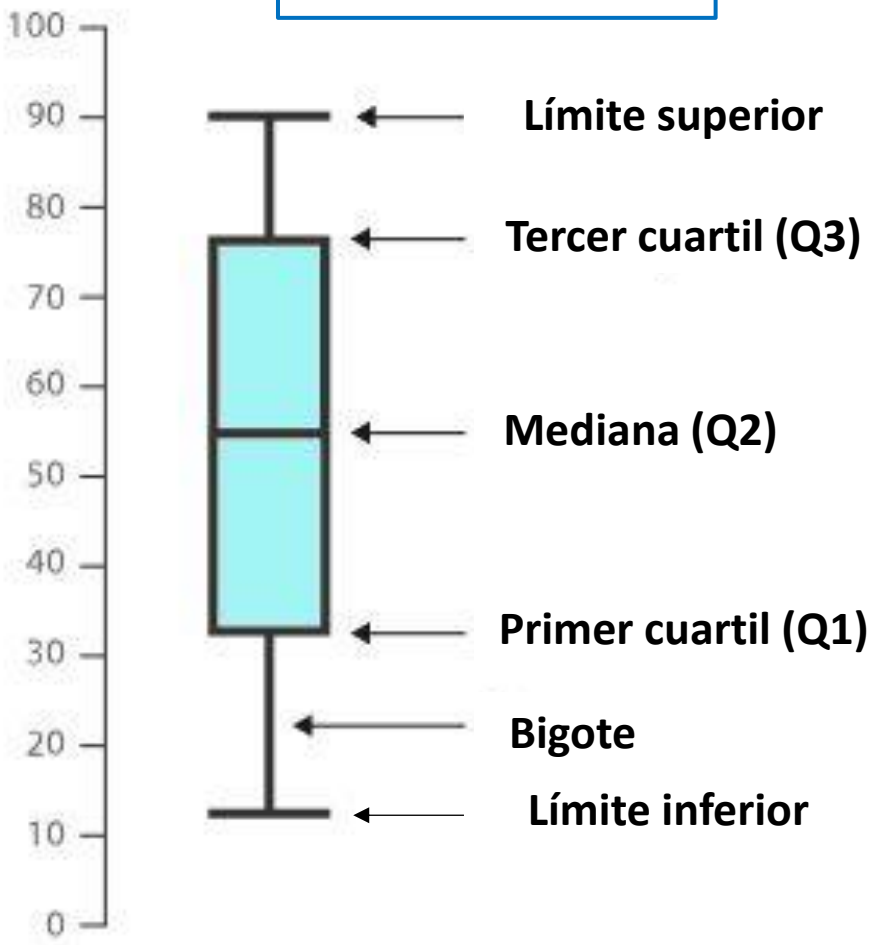
Se construye a partir de las siguientes medidas:

- El **primer y el tercer cuartil, Q1 y Q3**, delimitan la **caja central**.
- La **longitud de la caja** viene dada por el **IQR** (**rango intercuartílico**).
- Los límites inferior y superior se calculan como:

$$\text{Límite inferior} = Q1 - 1.5 \times IQR$$

$$\text{Límite superior} = Q3 + 1.5 \times IQR$$

- La **mediana (Q2) se representa con una línea horizontal en la caja centra**



En Python:

```
iris.boxplot()
```

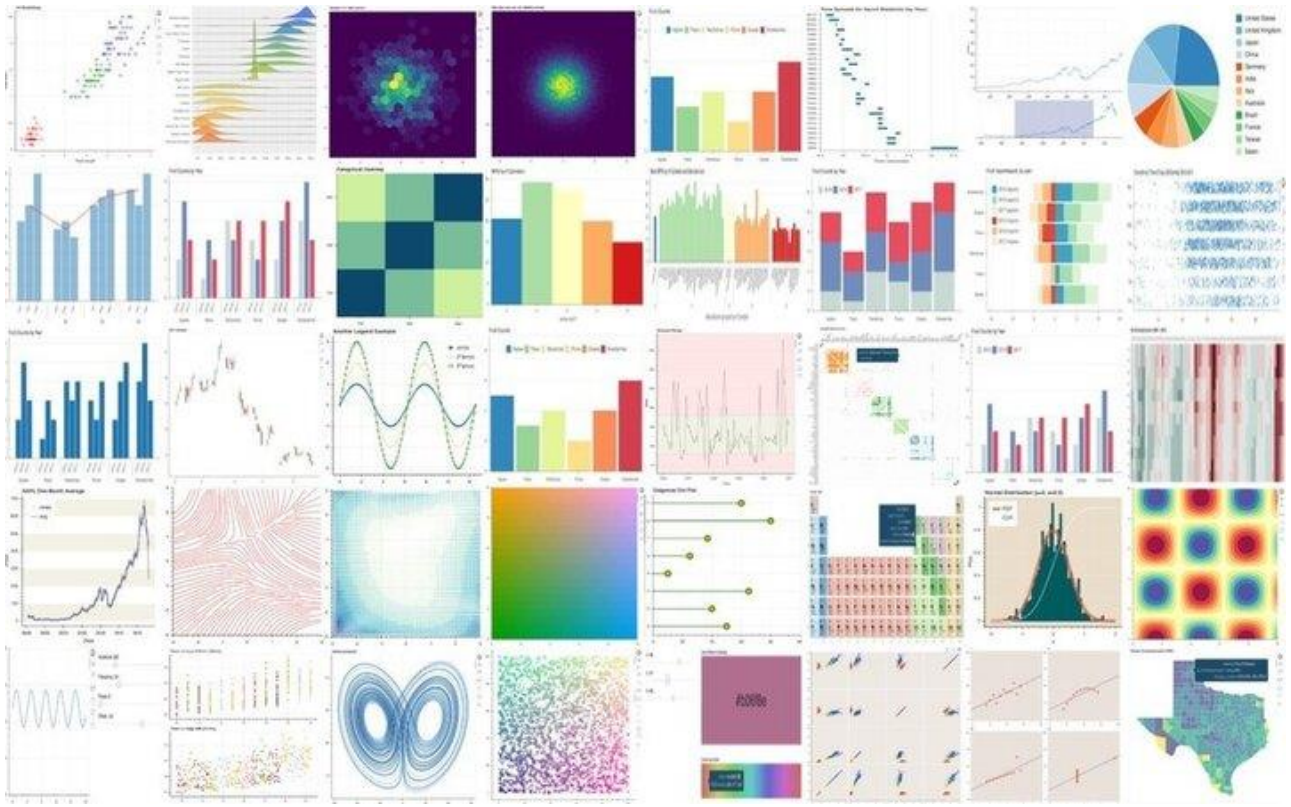
1.5. Visualizaciones en Python

Visualizaciones

Visualizar los datos es una tarea esencial para entender el dominio en el que estamos trabajando.

Algunas librerías de Visualización en Python:

- ❑ **Matplotlib**
- ❑ **Seaborn**



1.5. Visualizaciones en Python

Librería de Visualización Matplotlib

- Es la **librería gráfica más utilizada para realizar gráficos** en Python. **https://matplotlib.org/tutorials**
- Permite **realizar figuras de calidad en una variedad de formatos**.
- Es ideal para **generar gráficos simples con datos rápidamente** y sin mucho esfuerzo.
- También **tiene soporte para visualizaciones más avanzadas** y personalizadas.

```
import matplotlib.pyplot as plt
```

matplotlib

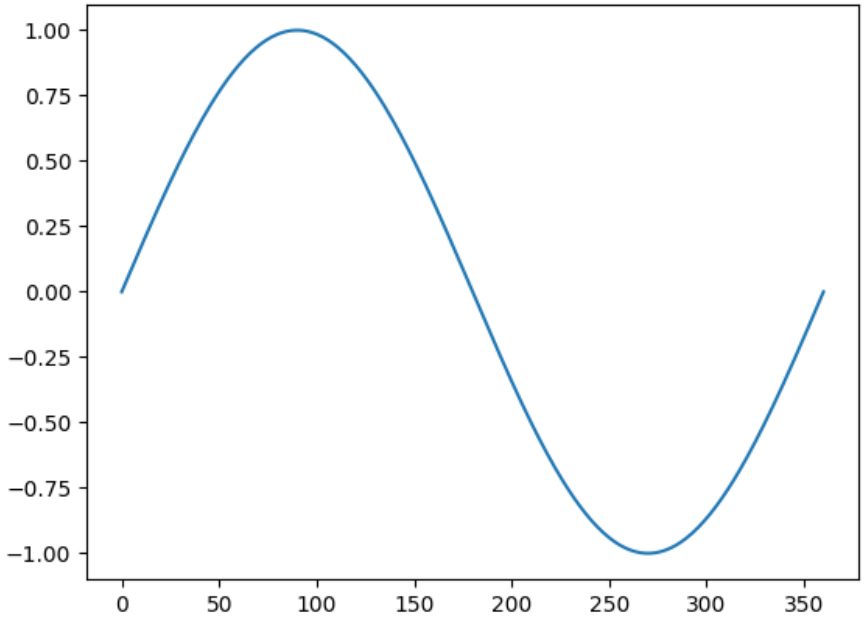
1.5. Visualizaciones en Python

Librería de Visualización Matplotlib

- La función **plot(x,y)** permite realizar un gráfico 2D de la variable **y** en función de **x**.

Ejemplo en Python de la función sen(x)

```
1 # Gráfico de la función sen(x)
2 import numpy as np
3 import matplotlib.pyplot as plt
4 x= np.arange(0, 361)
5 y= np.sin(x * np.pi / 180.)
6 plt.figure()
7 plt.plot(x,y)
```



1.5. Visualizaciones en Python

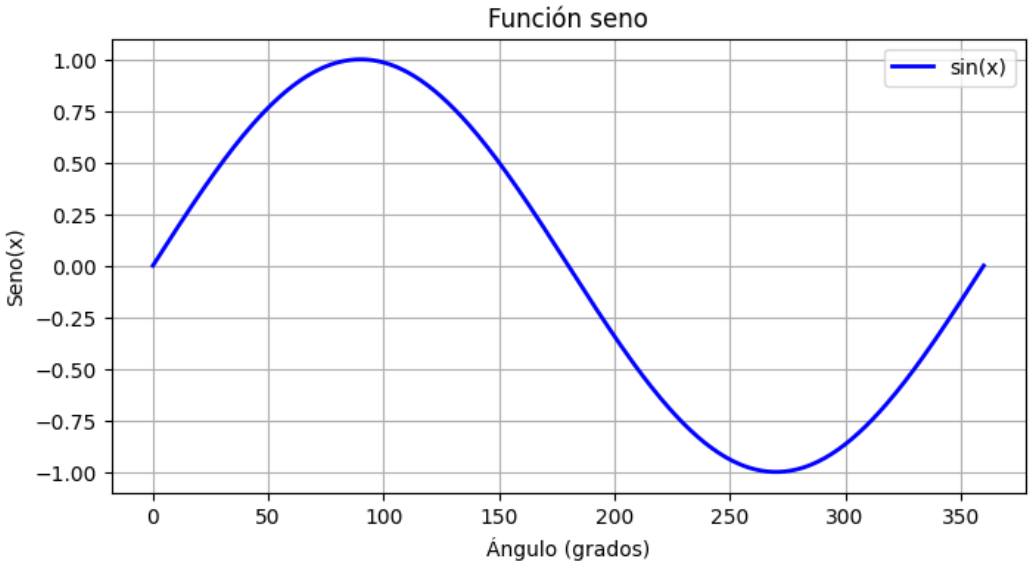
plt.figure() permite crear una nueva figura. Sino, todo lo ejecutado por matplotlib será impreso en la figura activa.

Librería de Visualización Matplotlib

- Matplotlib permite adornar las figuras con mucha información adicional.

Ejemplo en Python de la función sen(x)

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0, 361)  # 0 a 360 inclusive
y = np.sin(x * np.pi / 180)
plt.figure(figsize=(8,4))  # tamaño opcional de la figura
plt.plot(x, y, color='blue',
         linewidth=2, label='sin(x)')
plt.title('Función seno')
plt.xlabel('Ángulo (grados)')
plt.ylabel('Seno(x)')
plt.grid(True)
plt.legend()
plt.show()
```



1.5. Visualizaciones en Python

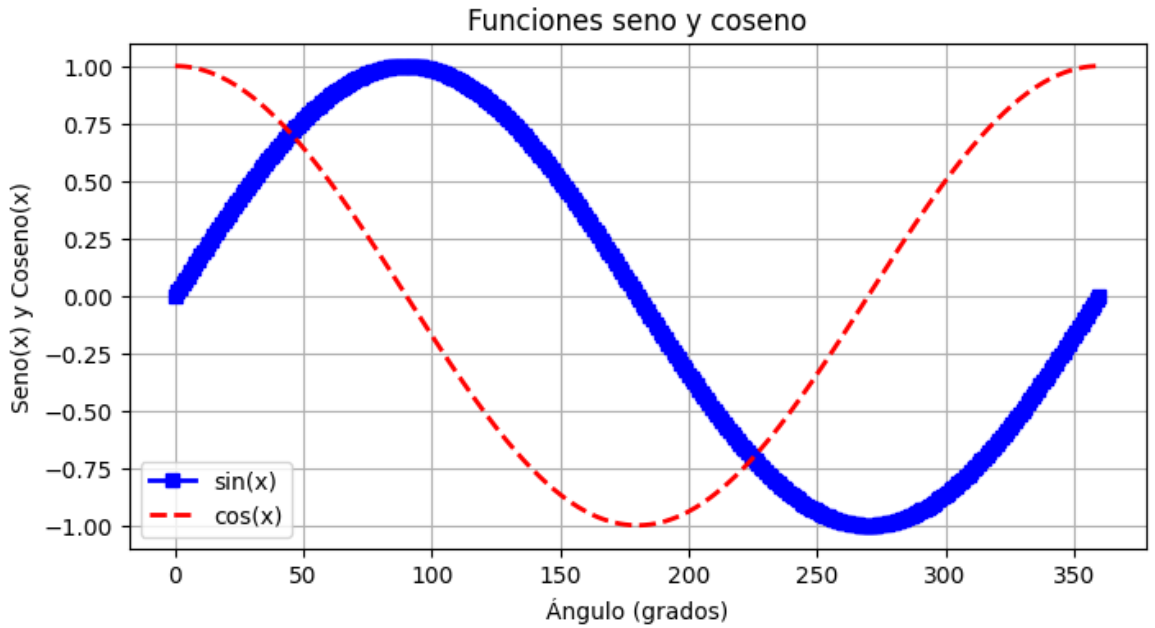
Librerías de Visualización en Python:

❑ Matplotlib

- **Superposición de dos gráficos en la misma figura** (seno y coseno)

Ejemplo en Python de la función sen(x) y cos(x)

```
# Superposición de las funciones sen(x) y cos(x)
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0, 361)  # 0 a 360 inclusive
y= np.sin(x * np.pi / 180.)
plt.figure(figsize=(8,4))  # tamaño opcional de la figura
plt.plot(x, y, color='b', marker='s', #b:blue, s:square
         linewidth=2, label='sin(x)')
y= np.cos(x * np.pi / 180.)
plt.plot(x, y, color='r', linestyle='--',
         #r: red, --: líneas entrecortadas
         linewidth=2, label='cos(x)')
plt.title('Funciones seno y coseno')
plt.xlabel('Ángulo (grados)')
plt.ylabel('Seno(x) y Coseno(x)')
plt.grid(True)
plt.legend()
plt.show()
```



1.5. Visualizaciones en Python

Librerías de Visualización en Python:

❑ Matplotlib

- Organizar **varios subgráficos dentro de una misma figura**.

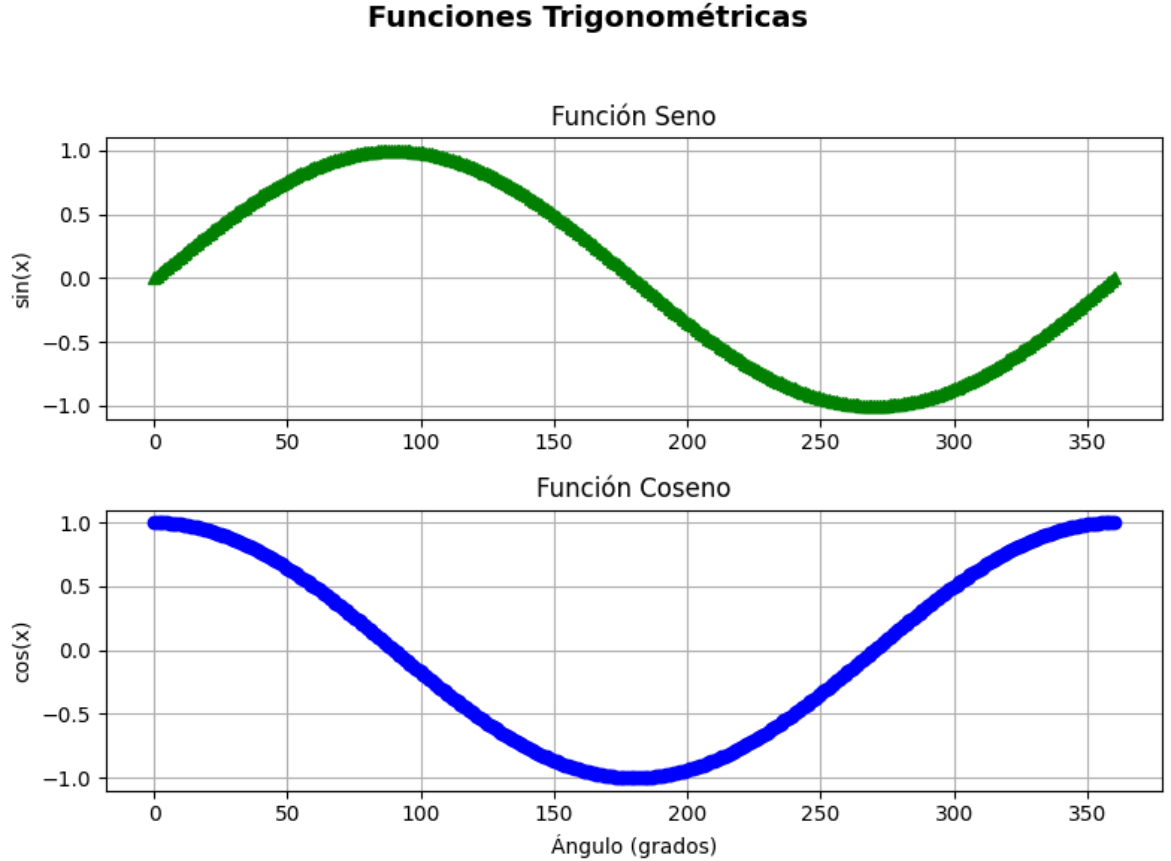
Ejemplo en Python de la función sen(x) y cos(x)

```python
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0, 361)
plt.figure(figsize=(8,6))

# Subplot 1: seno
plt.subplot(211)
y = np.sin(x * np.pi / 180.)
plt.plot(x, y, 'g^')
plt.title('Función Seno')
plt.ylabel('sin(x)')
plt.grid(True)

# Subplot 2: coseno
plt.subplot(212)
y = np.cos(x * np.pi / 180.)
plt.plot(x, y, 'bo')
plt.title('Función Coseno')
plt.xlabel('Ángulo (grados)')
plt.ylabel('cos(x)')
plt.grid(True)

# Título general de la figura
plt.suptitle('Funciones Trigonométricas',
             fontsize=14, fontweight='bold')
plt.tight_layout(rect=[0, 0, 1, 0.95]) # ajusta
                 # los espacios entre subplots
plt.show()
```



1.5. Visualizaciones en Python

Librería de Visualización Seaborn

- Es una **librería de visualización en Python basada en Matplo**tlib. **https://seaborn.pydata.org/**

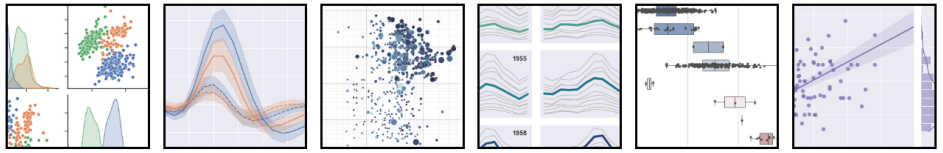
- **Crea gráficos atractivos e informativos de manera sencilla**.

- Facilita principalmente la **creación de gráficos estadísticos, como diagramas de dispersión, diagramas de caja, gráficos de barras, mapas de calor, y gráficos de distribución** (histogramas y estimaciones de densidad del núcleo).

- Ejemplo de visualización del **dataset Iris en la plataforma kaggle**

https://www.kaggle.com/noelano/seaborn-visualization-on-iris-data-set

seaborn



1.6. Visualizaciones con el dataset Iris

Dataset Iris

El **dataset "Iris" está almacenado en un archivo "csv"** (Comma Separated Values). Estos son **archivos de texto plano**, donde **cada registro está delimitado por un Enter**, y **cada columna por una coma.**

El dataset **contiene información sobre 3 especies distintas de flores**. **Contiene 4 atributos:** tamaño y largo del pétalo, tamaño y largo del sépalo.

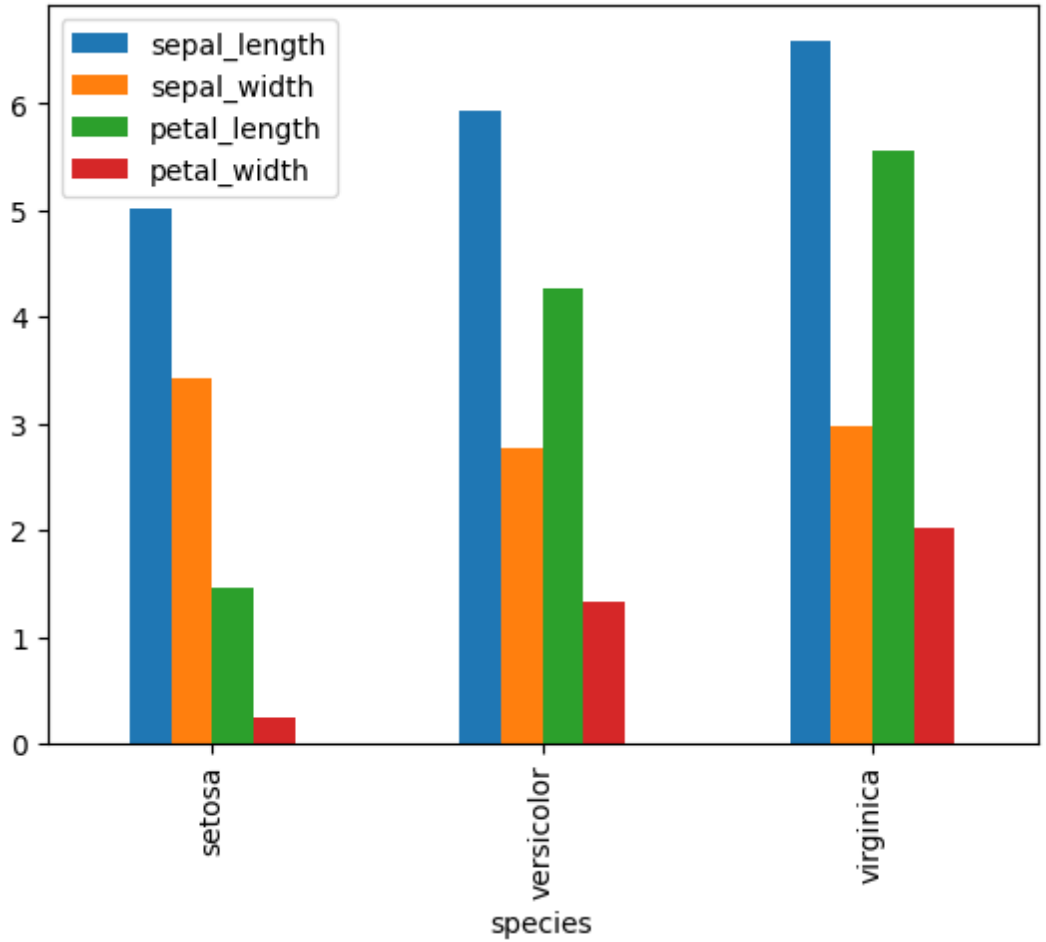
```
sepal_length,sepal_width,petal_length,petal_width,species
5.1,3.5,1.4,0.2,setosa
4.9,3.0,1.4,0.2,setosa
4.7,3.2,1.3,0.2,setosa
4.6,3.1,1.5,0.2,setosa
5.0,3.6,1.4,0.2,setosa
5.4,3.9,1.7,0.4,setosa
```

1.6. Visualizaciones con el dataset Iris

Gráfico de barras

Gráfico de barras de la media de cada atributo diferenciado por clase.

```
# Gráfico de barras de la media de cada atributo,
# diferenciado por clase.
plot_data= iris.groupby('species').mean()
plot_data.plot(kind='bar')
```

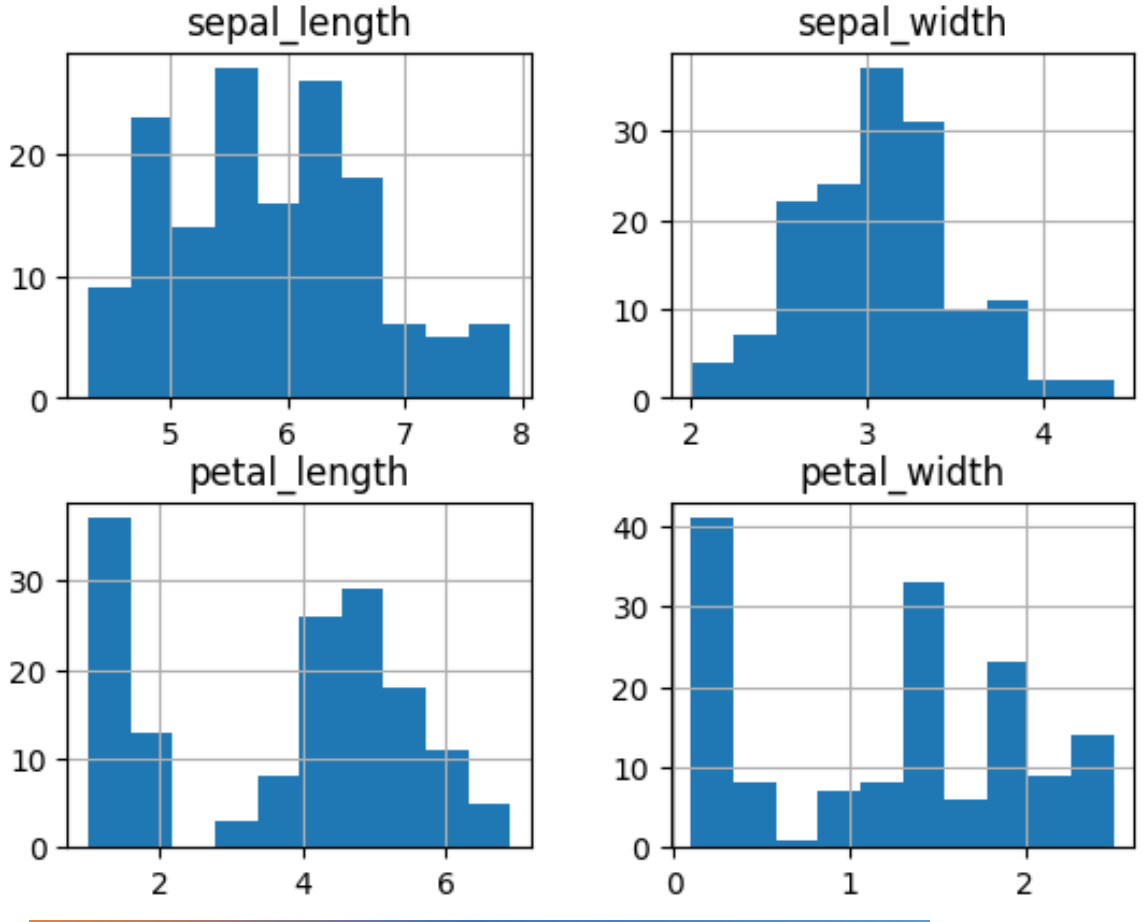


1.6. Visualizaciones con el dataset Iris

Histograma

Histograma para cada columna numérica diferente.

```
# Histograma para cada
# columna numérica diferente
iris.hist()
```

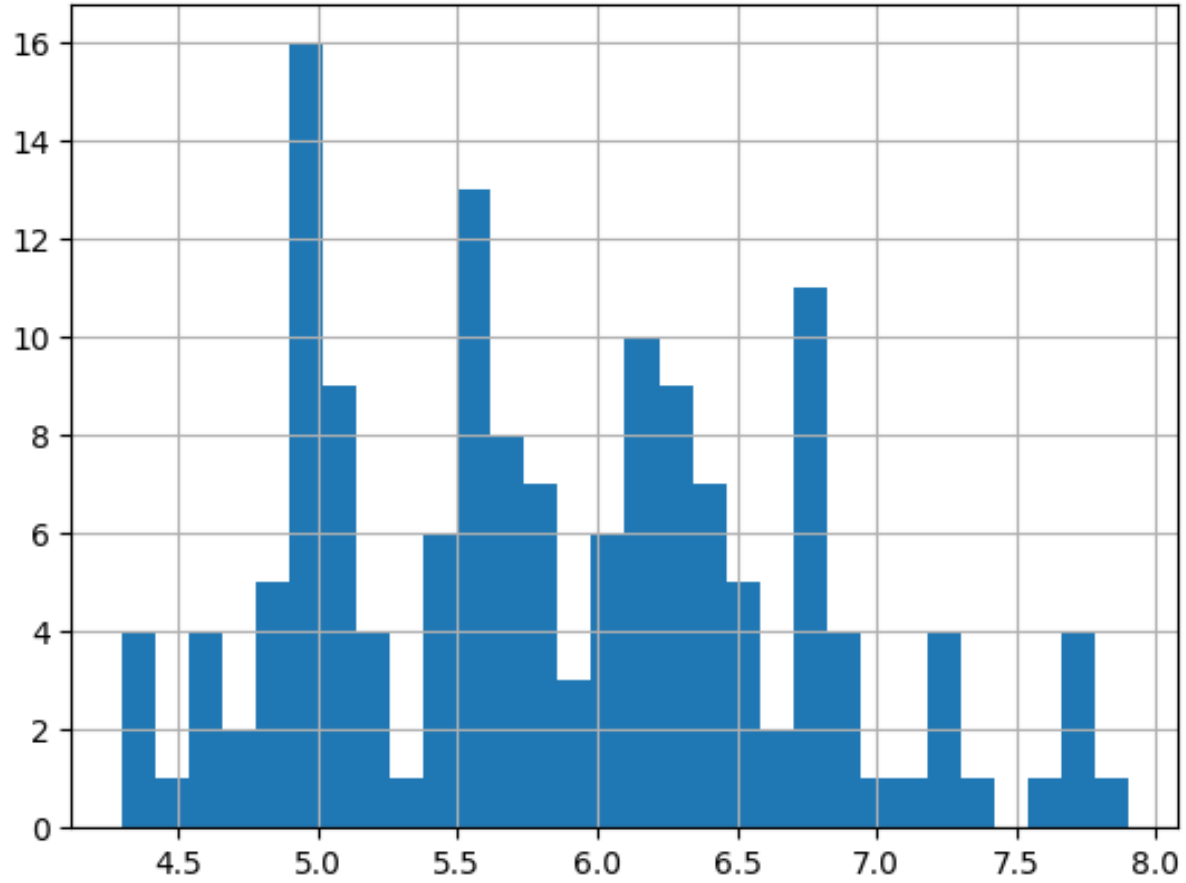


1.6. Visualizaciones con el dataset Iris

Histograma

Histograma para la columna “sepal_length”.

```
# Histograma para la columna “sepal_length”
iris.sepal_length.hist(bins=30)
```

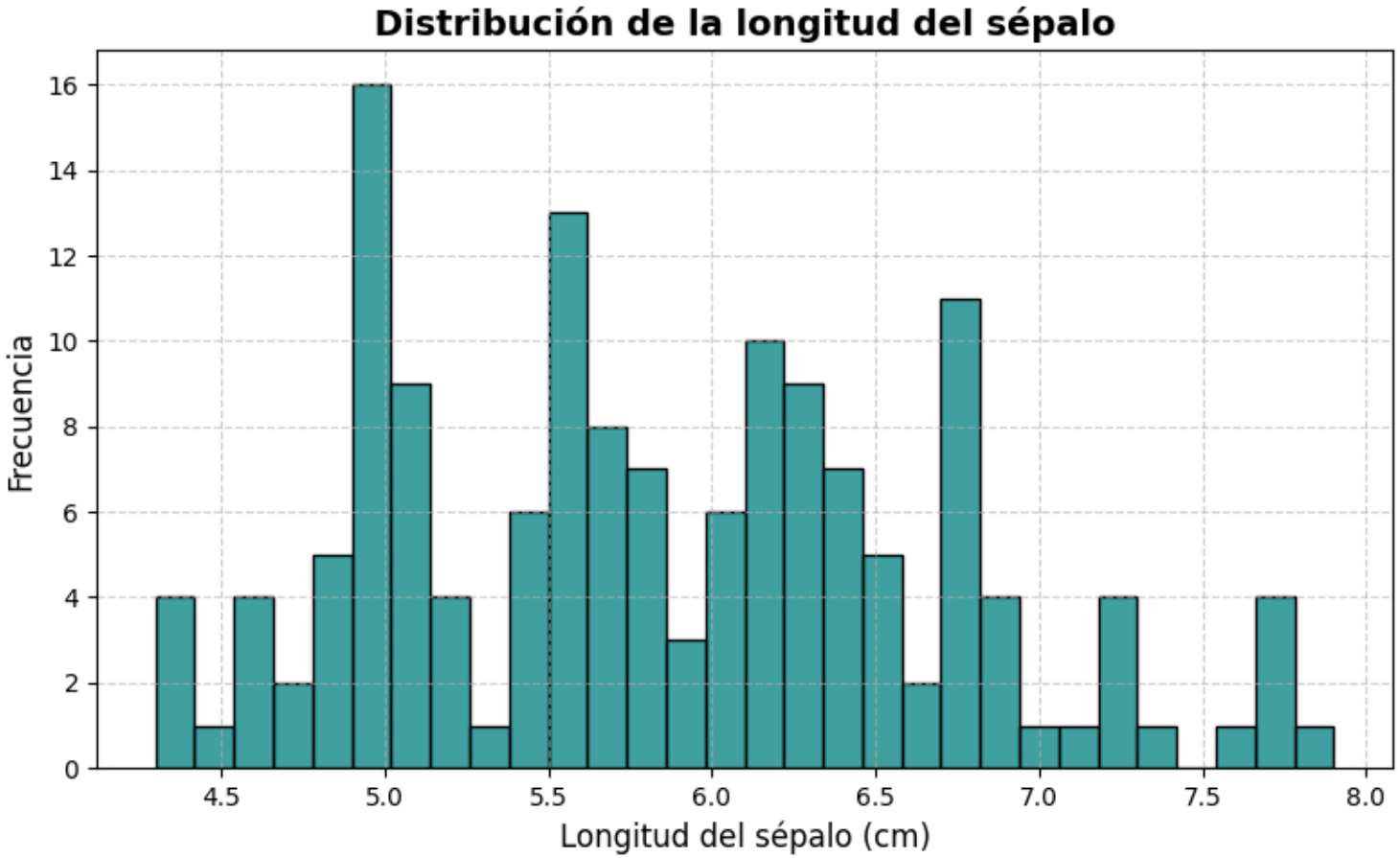


1.6. Visualizaciones con el dataset Iris

Histograma

Histograma para la columna “sepal_length” con la **función histplot de la librería seaborn**.

```
# Histograma para la columna “sepal_length”
# con la función histplot de la librería seaborn
plt.figure(figsize=(8,5))
sns.histplot(iris['sepal_length'], bins=30,
             color='teal', edgecolor='black')
# Títulos y etiquetas
plt.title('Distribución de la longitud del sépalo',
          fontsize=14, fontweight='bold')
plt.xlabel('Longitud del sépalo (cm)', fontsize=12)
plt.ylabel('Frecuencia', fontsize=12)
# Cuadrícula y formato
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



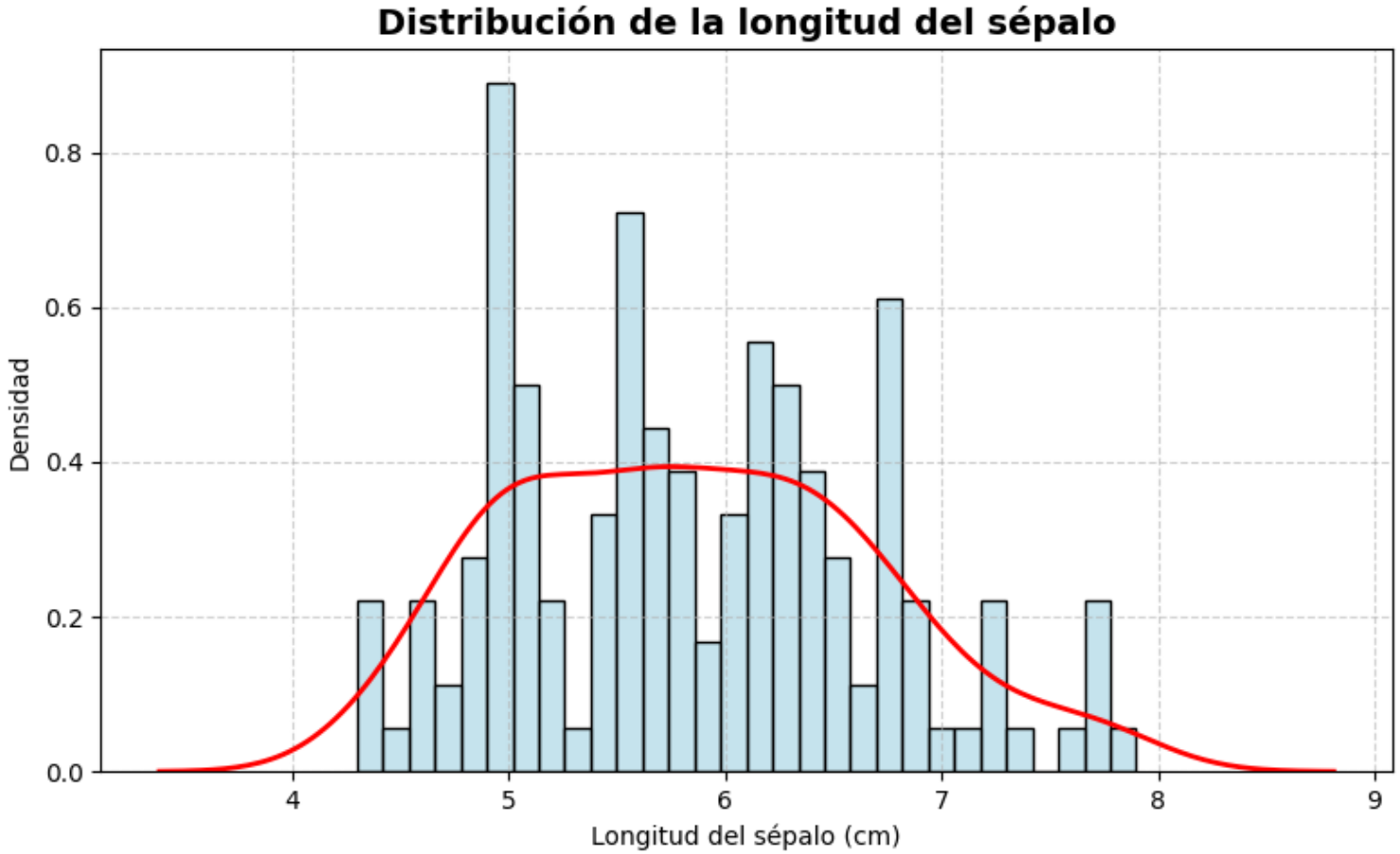
1.6. Visualizaciones con el dataset Iris

La **curva de densidad KDE (Kernel Density Estimation, o Estimación de Densidad por Núcleos)** es una forma suave y continua de representar cómo se distribuyen los datos de una variable numérica.

Histograma más curva de densidad KDE

Histograma para la columna "sepal_length" con la **curva de densidad KDE superpuesta**.

```
# Histograma para la columna "sepal_length"
# con la curva de densidad superpuesta
plt.figure(figsize=(8,5))
# Histograma con curva de densidad
sns.histplot(
    iris['sepal_length'],
    bins=30,
    color='lightblue',
    edgecolor='black',
    stat='density',   # importante para
    # que la escala coincida con la KDE
    alpha=0.7
)
# Curva de densidad (KDE) con color personalizado
sns.kdeplot(
    iris['sepal_length'],
    color='red',
    linewidth=2
)
plt.title('Distribución de la longitud del sépalo',
          fontsize=14, fontweight='bold')
plt.xlabel('Longitud del sépalo (cm)')
plt.ylabel('Densidad')
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



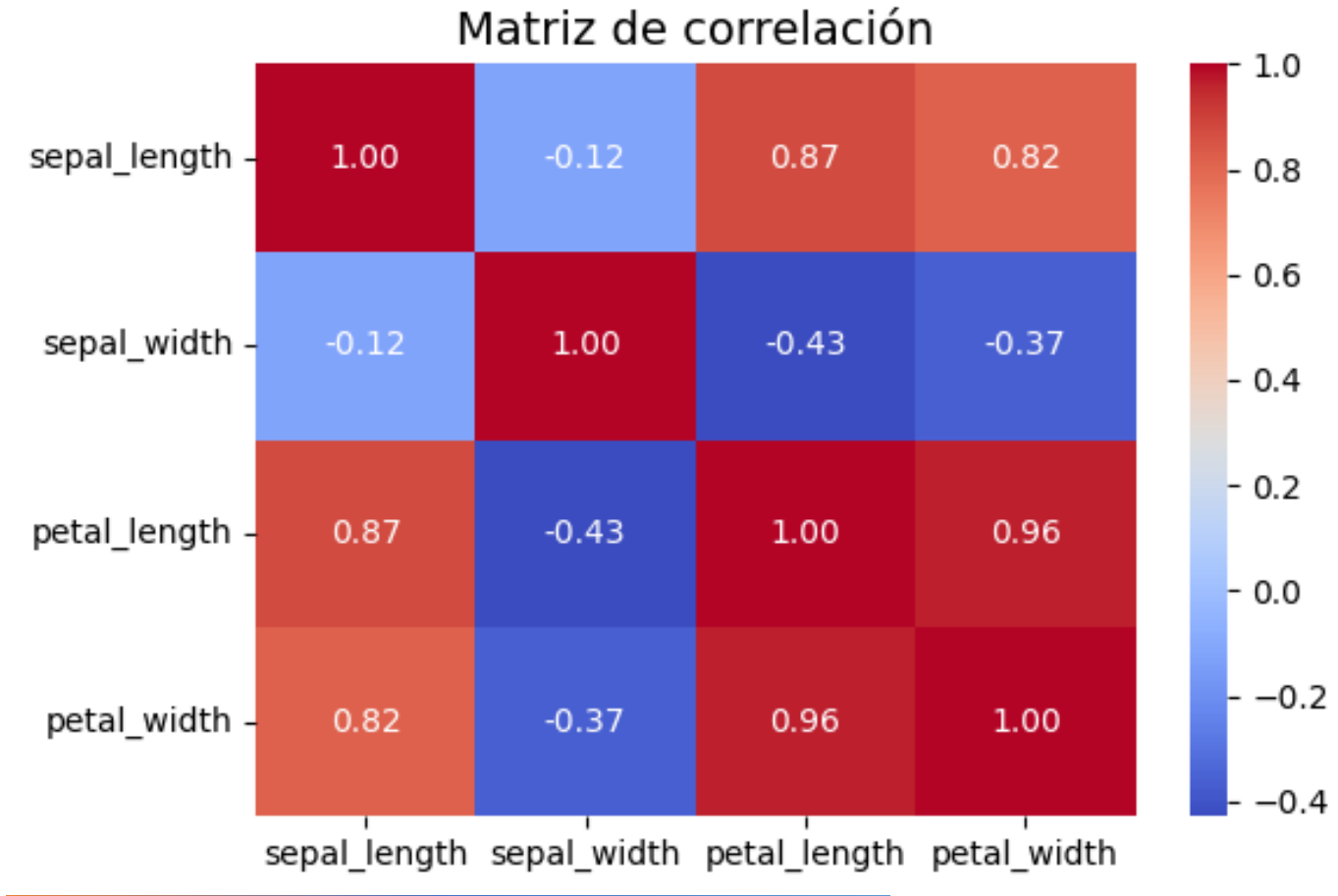
1.6. Visualizaciones con el dataset Iris

Matriz de correlación

Matriz de correlación entre columnas numéricas.

```
# Calcular matriz de correlación
# (solo columnas numéricas)
corr = iris.corr(numeric_only=True)

# Mostrar la matriz
plt.figure(figsize=(6,4))
sns.heatmap(corr, annot=True,
            cmap='coolwarm', fmt=".2f")
plt.title('Matriz de correlación',
          fontsize=14)
plt.tight_layout()
plt.show()
```



1.6. Visualizaciones con el dataset Iris

Scatter permite visualizar una dispersión de puntos en un espacio 2D

Diagrama de dispersión (scatter plot):

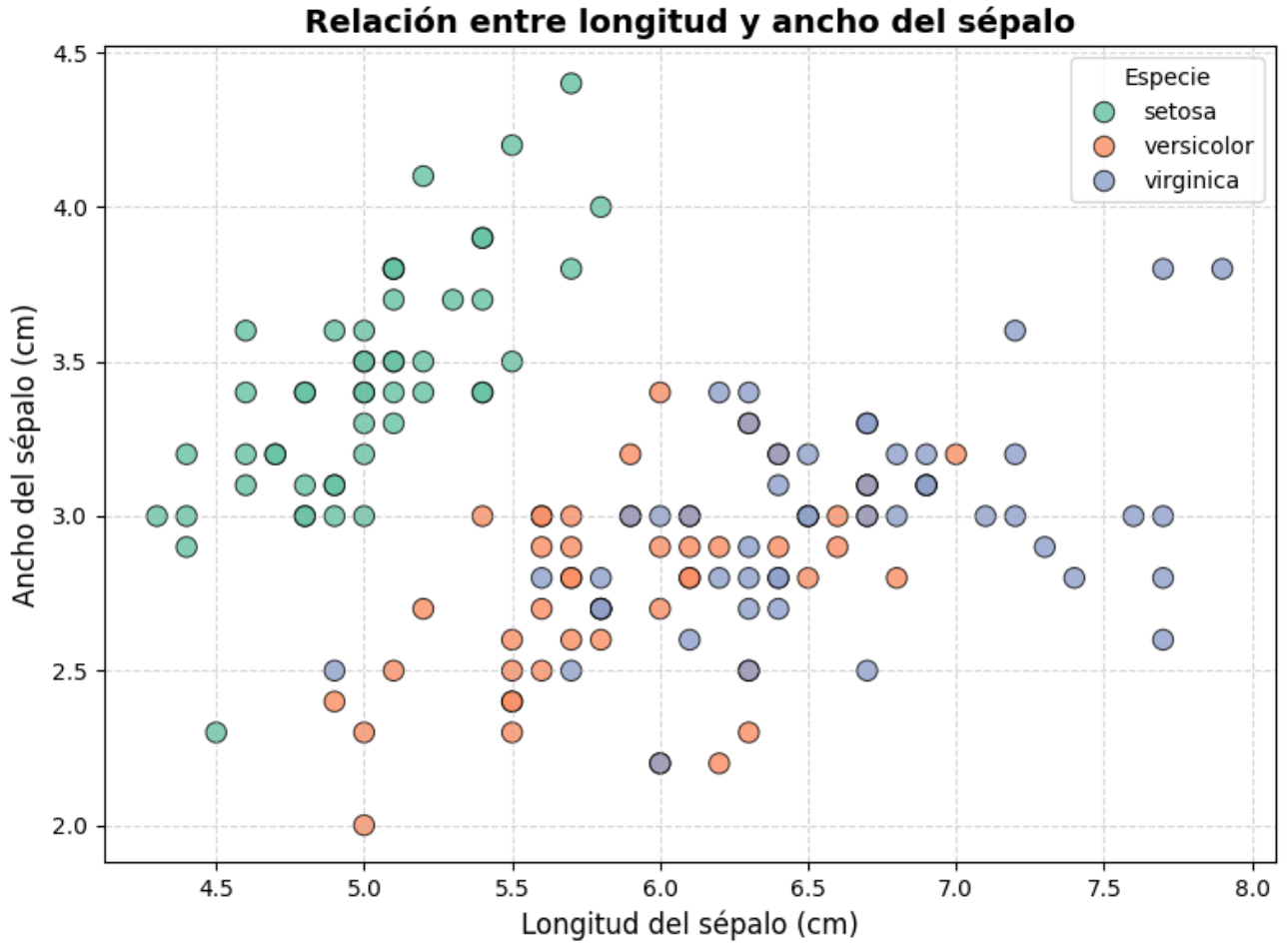
Diagrama de dispersión entre dos de las variables del dataset.

```
# Diagrama de dispersión (scatter plot):
plt.figure(figsize=(8,6))

# Gráfico de dispersión con Seaborn
sns.scatterplot(
    data=iris,
    x='sepal_length',
    y='sepal_width',
    hue='species',
    palette='Set2',
    s=80,
    edgecolor='black',
    alpha=0.8
)

# Títulos y etiquetas
plt.title('Relación entre longitud y ancho del sépalo',
          fontsize=14, fontweight='bold')
plt.xlabel('Longitud del sépalo (cm)', fontsize=12)
plt.ylabel('Ancho del sépalo (cm)', fontsize=12)

# Mejoras visuales
plt.legend(title='Especie')
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```



1.6. Visualizaciones con el dataset Iris

Matriz de dispersión:

Matriz de dispersión en 2D para cada par de columnas numéricas.

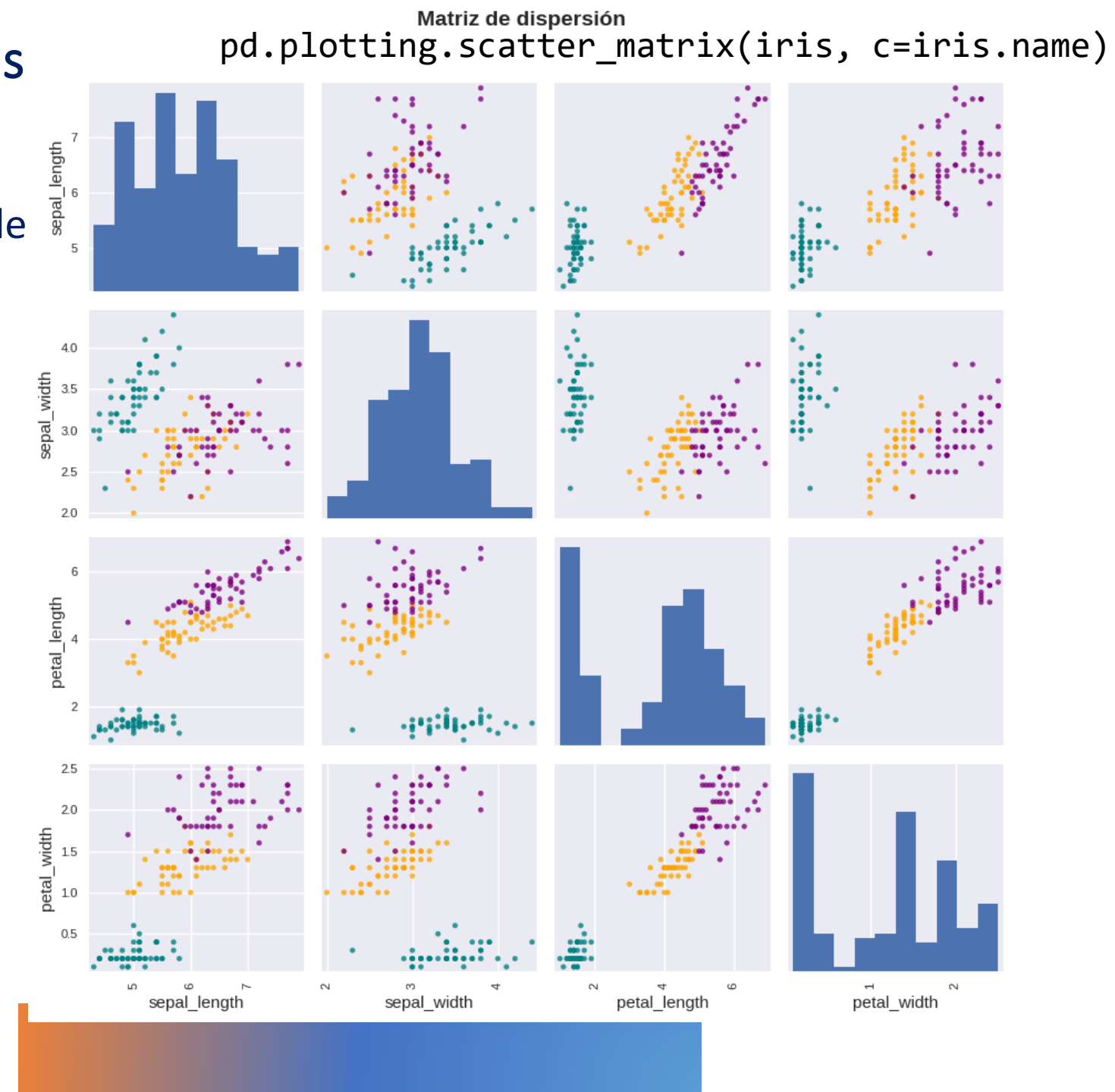
```
# Asignar colores personalizados por especie
colores = {'setosa': 'teal',
           'versicolor': 'orange',
           'virginica': 'purple'}
iris['color'] = iris['species'].map(colores)

# Estilo general
plt.style.use('seaborn-v0_8')

# Matriz de dispersión con pandas
pd.plotting.scatter_matrix(
    iris[['sepal_length', 'sepal_width',
          'petal_length', 'petal_width']],
    # solo columnas numéricas
    figsize=(9, 9),
    diagonal='hist',
    color=iris['color'],
    alpha=0.8
#    edgecolor='black'
)

plt.suptitle('Matriz de dispersión',
             fontsize=14, fontweight='bold', y=1.02)
plt.tight_layout()
plt.show()
```

pd.plotting.scatter_matrix(iris, c=iris.name)



1.6. Visualizaciones con el dataset Iris

Diagrama de cajas y bigotes (boxplot):

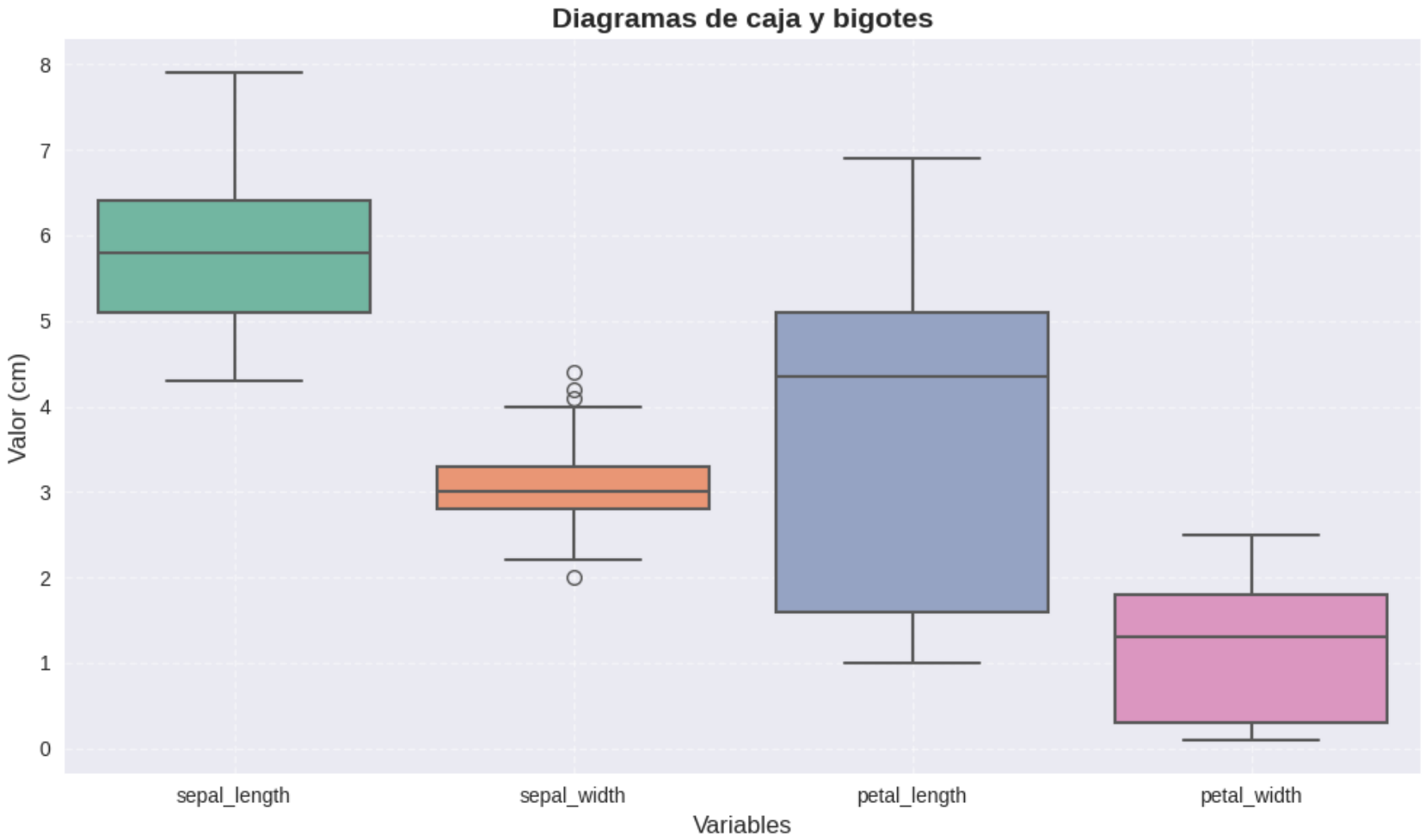
Diagrama de cajas para las cuatro variables numéricas del dataset.

```
# Crear figura
plt.figure(figsize=(10,6))

# Boxplot de todas las columnas numéricas
sns.boxplot(data=iris[['sepal_length',
                       'sepal_width',
                       'petal_length',
                       'petal_width']],
            palette='Set2', linewidth=1.5)

# Títulos y etiquetas
plt.title('Diagramas de caja y bigotes',
          fontsize=14, fontweight='bold')
plt.xlabel('Variables', fontsize=12)
plt.ylabel('Valor (cm)', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.5)

plt.tight_layout()
plt.show()
```



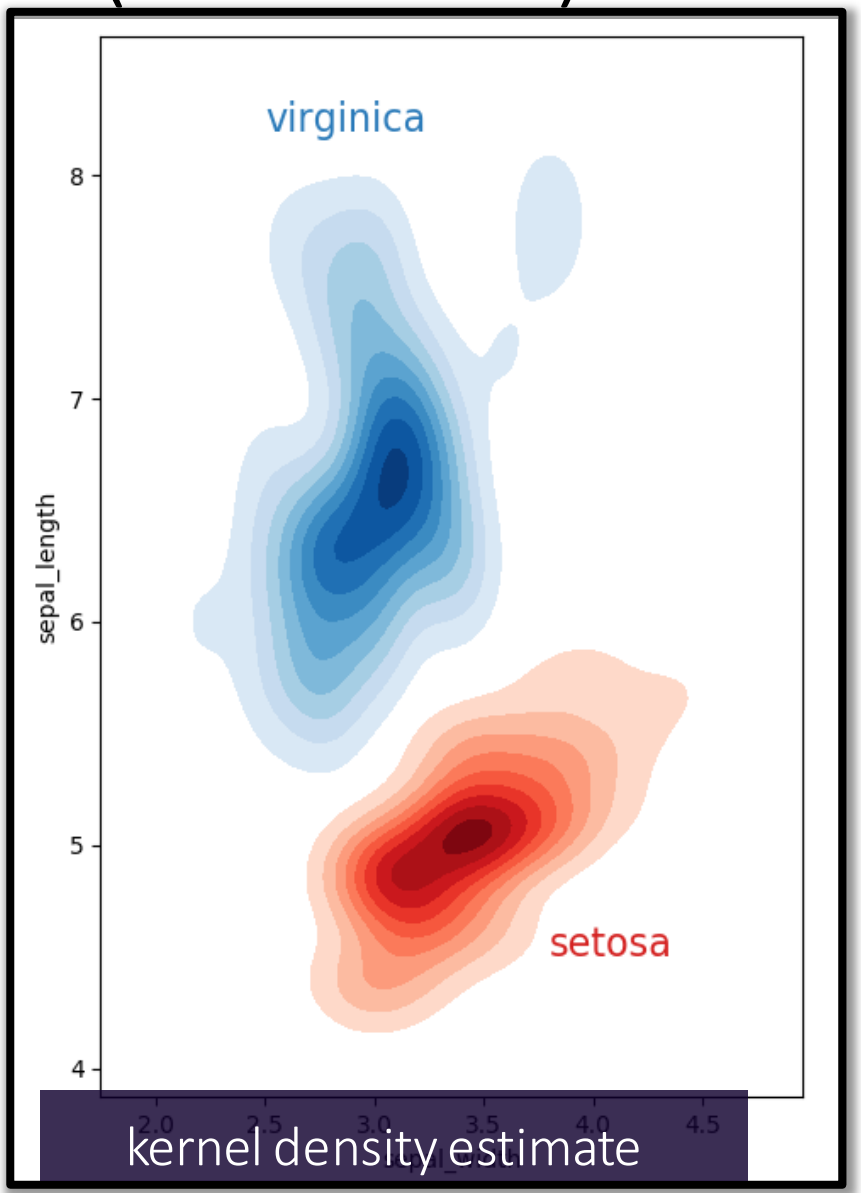
Visualización – gráficos de densidad (KDEPLOT)

```
setosa = iris.query("name == 'setosa'")
virginica = iris.query("name == 'virginica'")

# Set up the figure
f, ax = plt.subplots(figsize=(8, 8))
ax.set_aspect("equal")

# Draw the two density plots
ax = sns.kdeplot(setosa.sepal_width,
setosa.sepal_length,
                 cmap="Reds", shade=True,
shade_lowest=False)
ax = sns.kdeplot(virginica.sepal_width,
virginica.sepal_length,
                 cmap="Blues", shade=True,
shade_lowest=False)

# Add labels to the plot
red = sns.color_palette("Reds")[-2]
blue = sns.color_palette("Blues")[-2]
ax.text(2.5, 8.2, "virginica", size=16, color=blue)
ax.text(3.8, 4.5, "setosa", size=16, color=red)
```



kernel density estimate