



# 04

## **VISUALIZACIÓN**



# ¿QUÉ ES?

Proceso de representación de datos, en formato gráfico, de forma clara y eficaz.

Permite captar conceptos difíciles o identificar nuevos patrones.



# VENTAJAS

1

**ABSORCIÓN DE GRANDES CANTIDADES DE DATOS**



2

**ACELERA EL PROCESO DE LA TOMA DE DECISIONES**



3

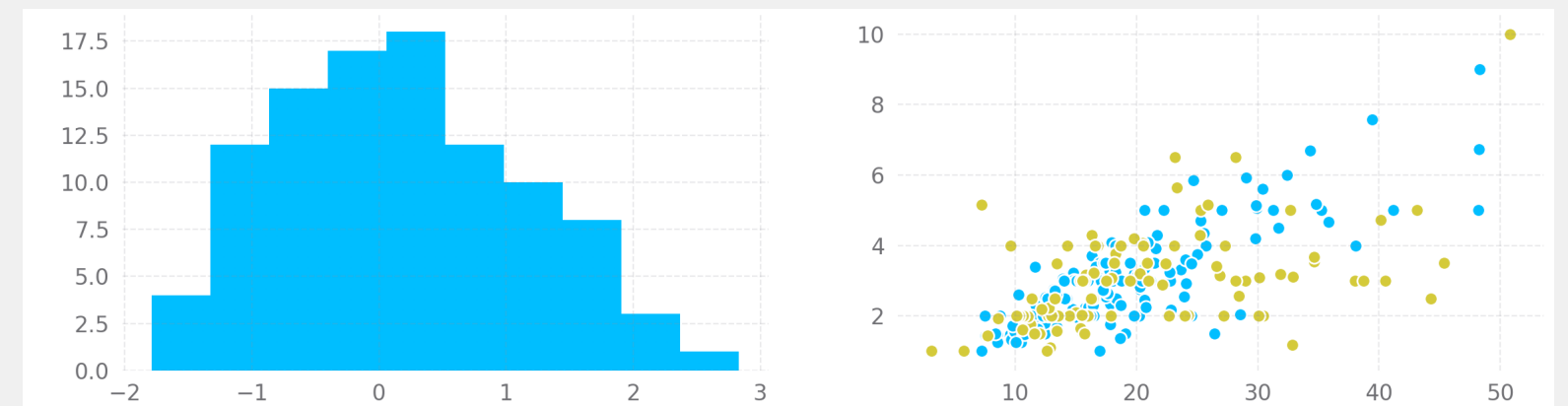
**PUEDE REVELAR PATRONES Y TENDENCIAS**



# LIBRERÍAS

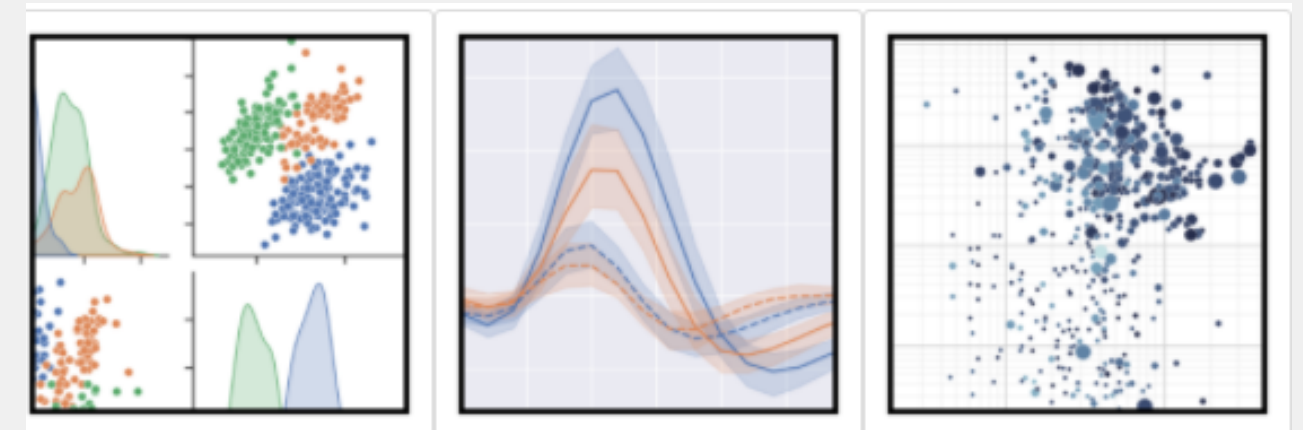
## MATPLOTLIB

Gráficos sencillos en pocas líneas de código



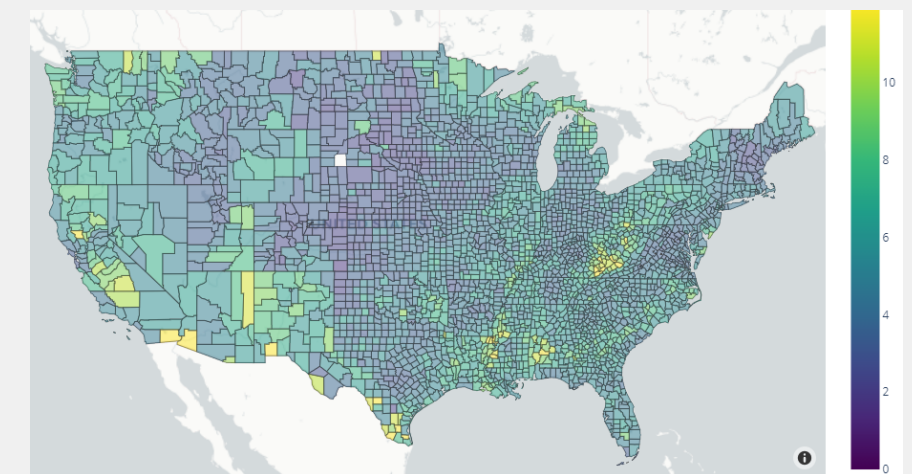
## SEABORN

Basado en matplotlib, más estilos de gráficos



## PLOTLY

Hace gráficos interactivos que no tienen otras librerías



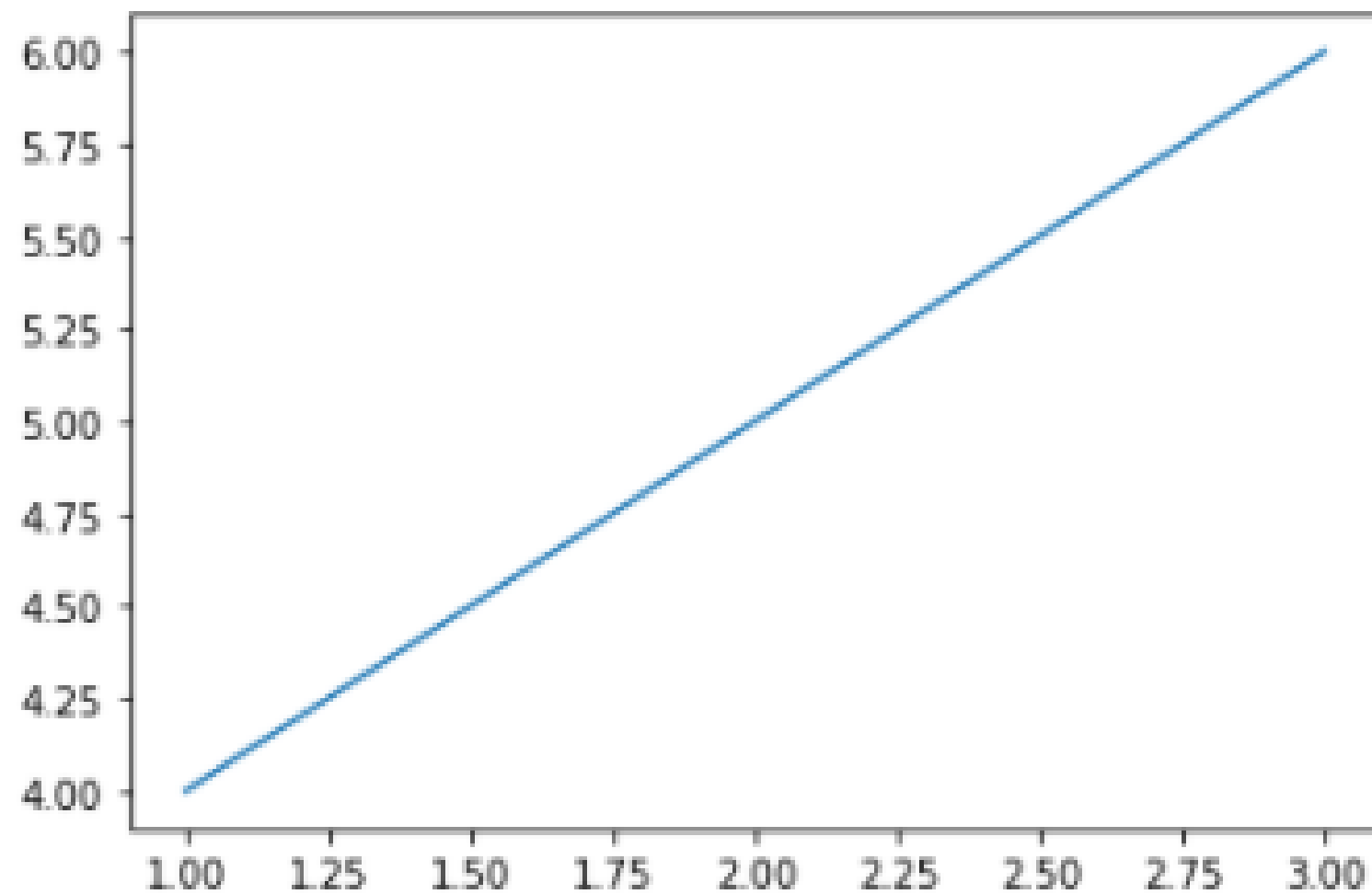


# TIPOS DE GRÁFICAS

# GRÁFICO DE LÍNEAS

`plt.plot(x, y)`

**PUNTOS CONECTADOS POR  
MEDIO DE UNA LÍNEA QUE  
MUESTRAN UNA TENDENCIA**



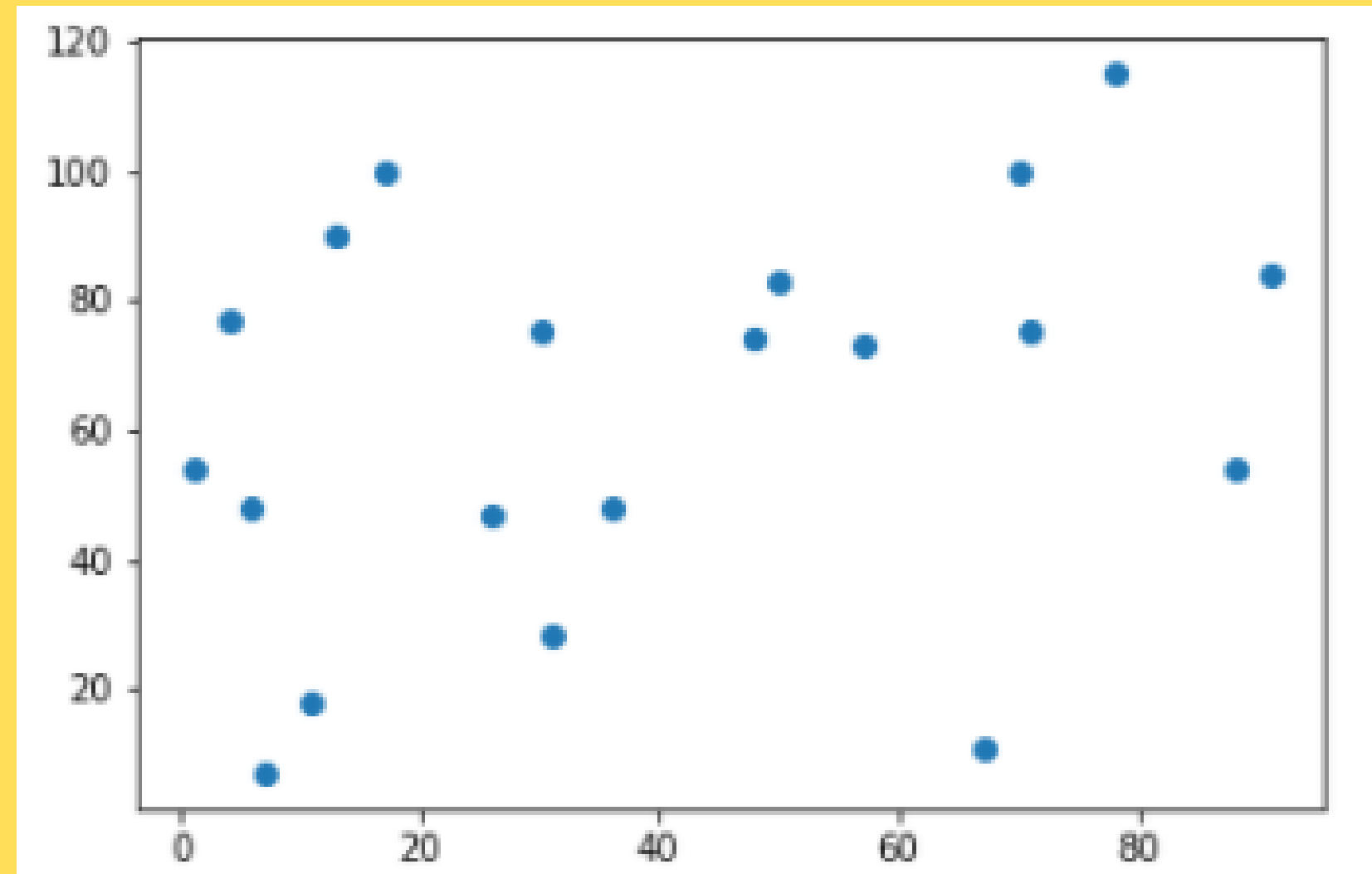
ej. tendencia de ventas  
en un tiempo de un año

# GRÁFICO DE DISPERSIÓN

`plt.scatter(x, y)`

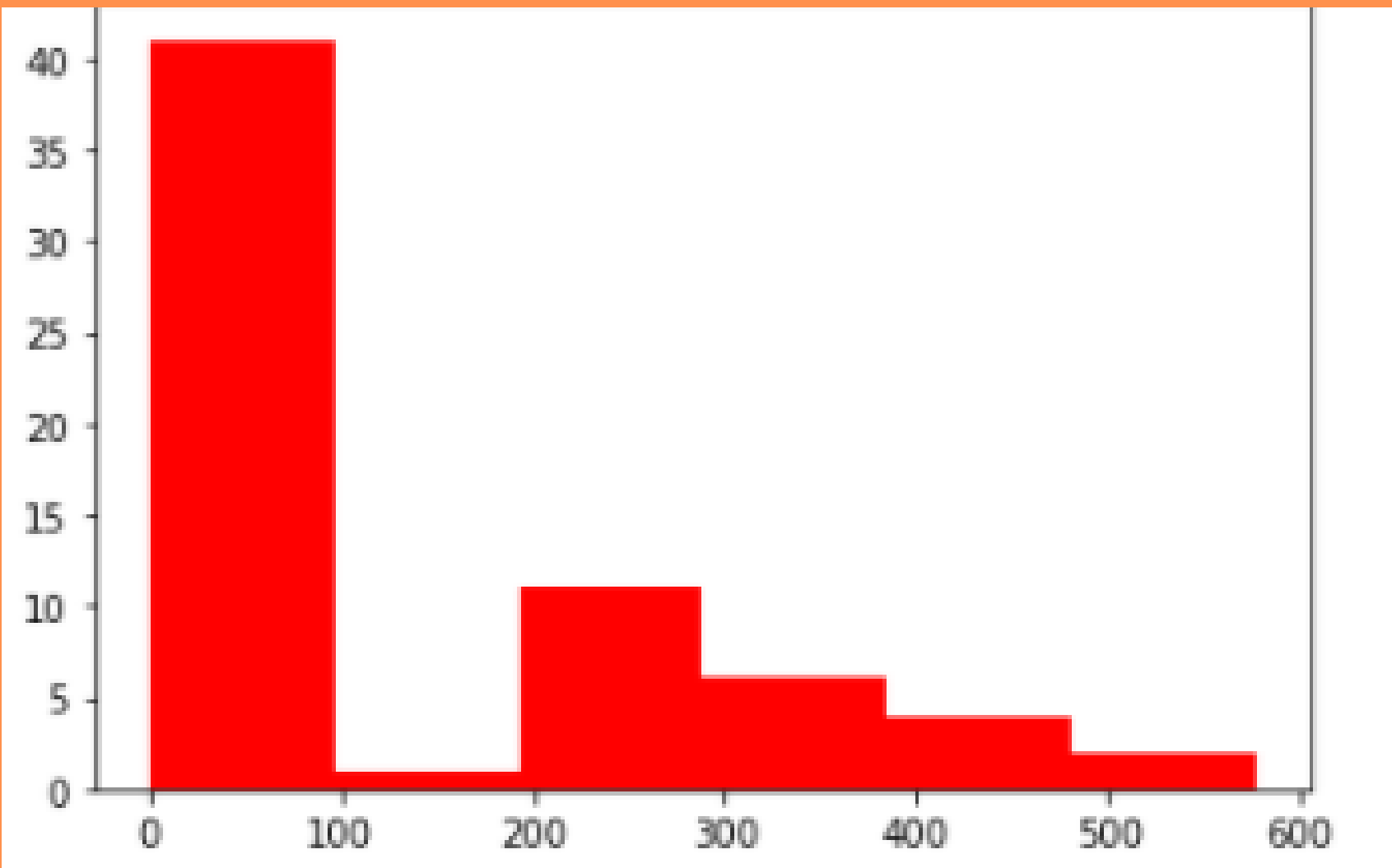
**MUESTRA RELACIÓN ENTRE  
DOS CONJUNTOS DE DATOS**

ej. edad vs peso



# HISTOGRAMA

```
plt.hist(x, num_bins, facecolor = 'red')
```



**MUESTRA DISTRIBUCIÓN DE  
VALORES EN UN RANGO  
DETERMINADO**

ej. frecuencia de edades  
por intervalos

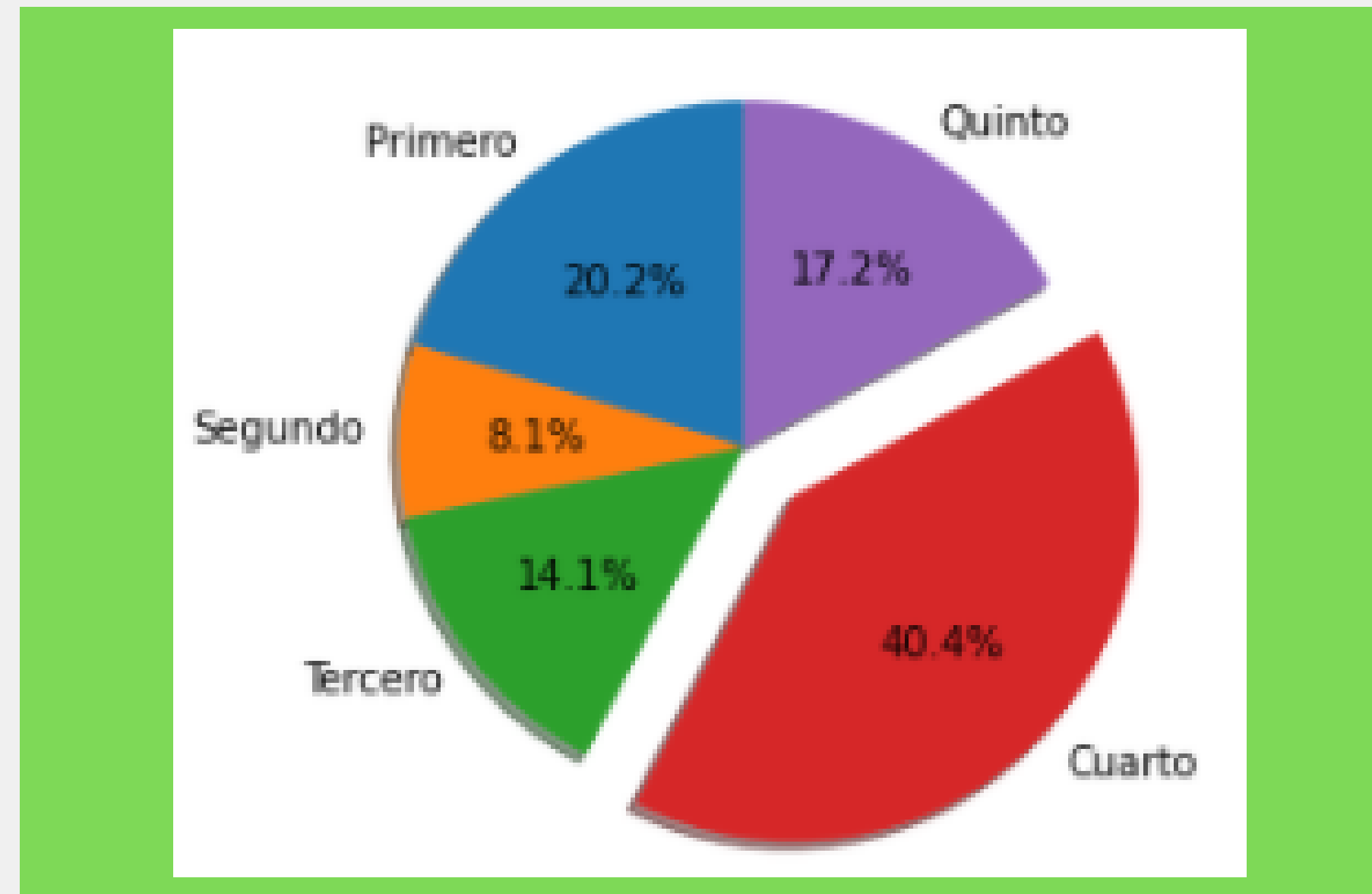


# GRÁFICO DE PASTEL

**REPRESENTA DATOS POR  
MEDIO DE PORCENTAJES Y  
PROPORCIONES**

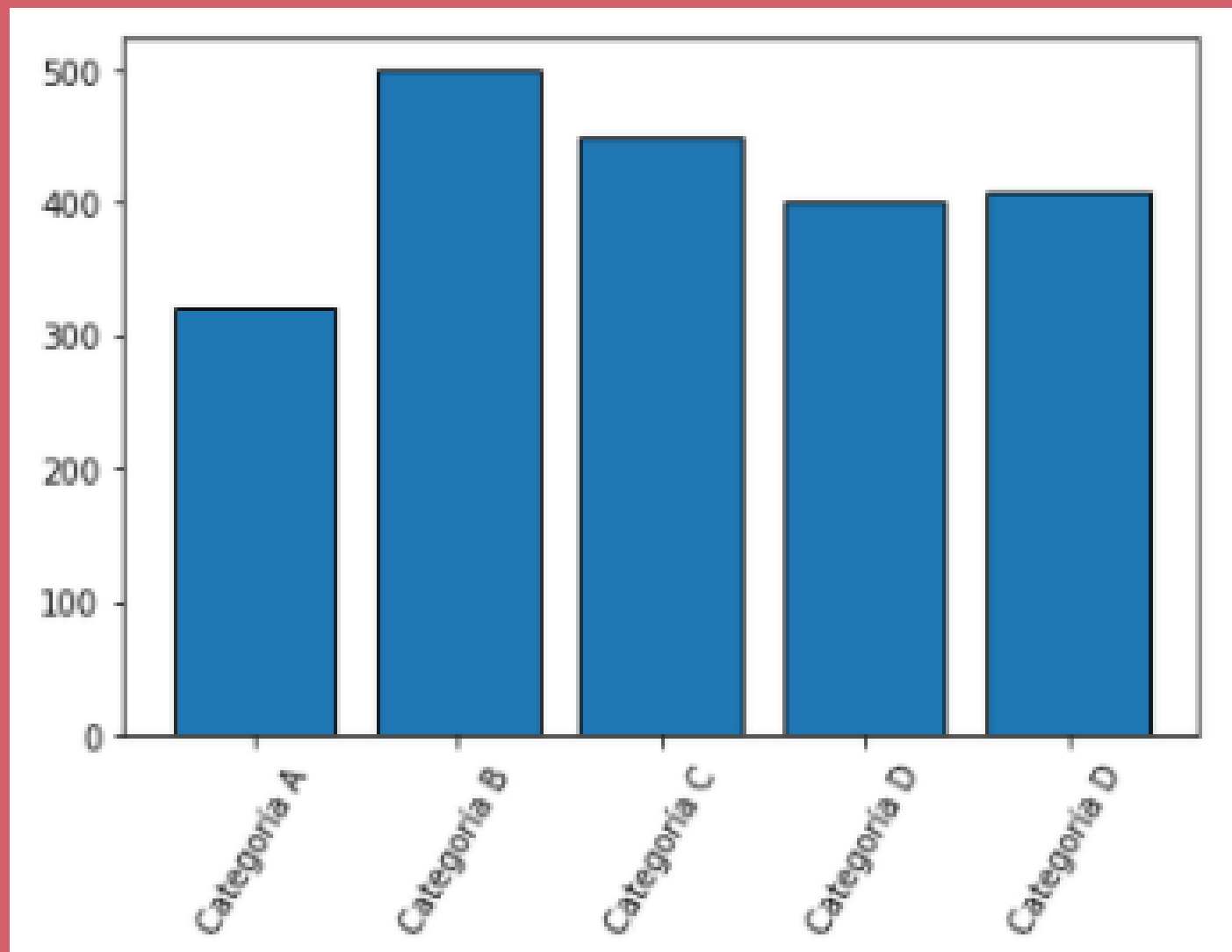
ej. proporción de  
estudiantes de cinco  
semestres

```
plt.pie(x, labels=y)
```



# GRÁFICO DE BARRAS

```
plt.bar(range(5), y, edgecolor='black')  
plt.xticks(range(5), x, rotation=60)
```



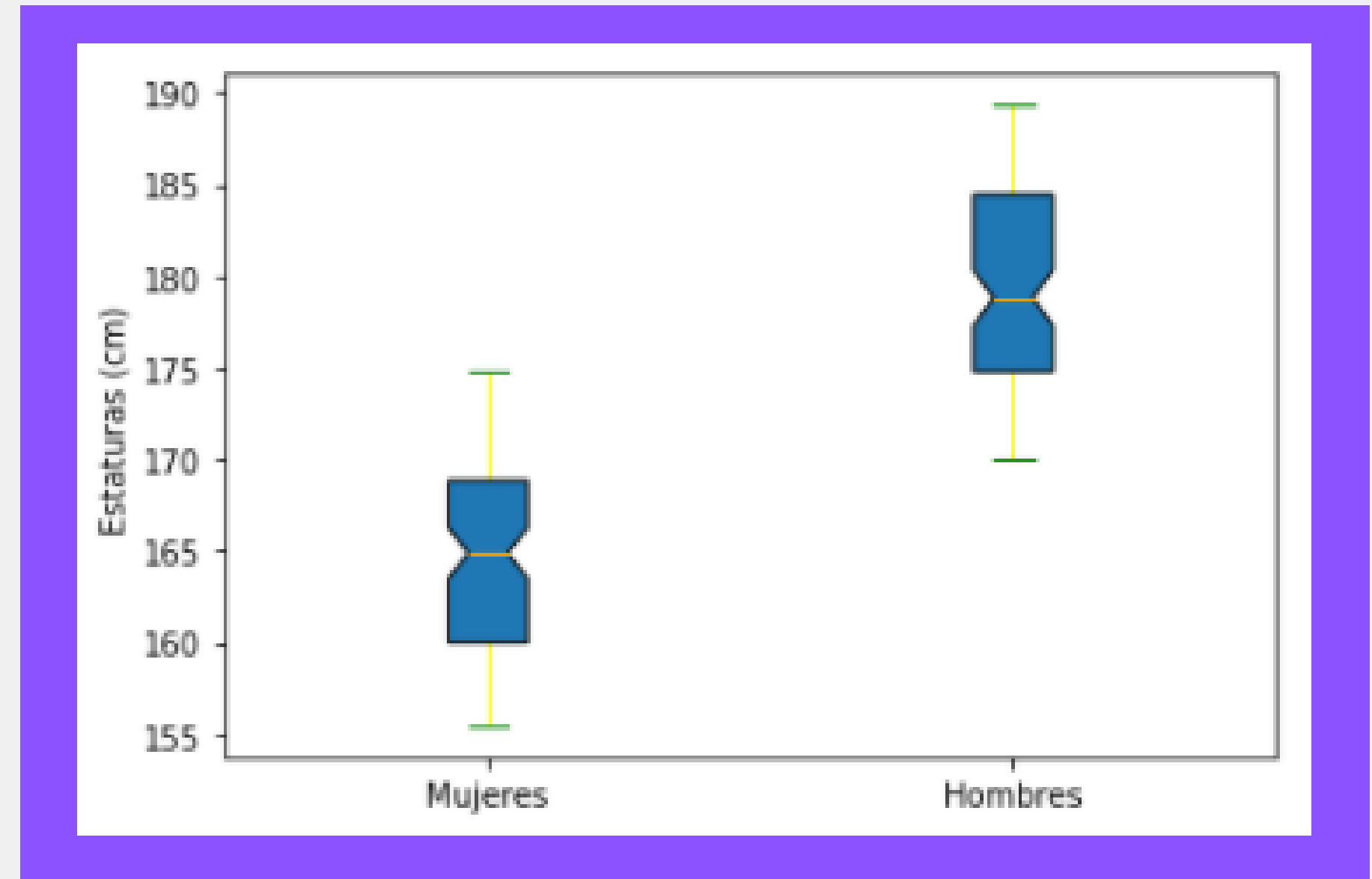
**REPRESENTA CONJUNTO DE  
DATOS DISCRETOS POR  
CATEGORÍAS**

NOTA: plt.barh

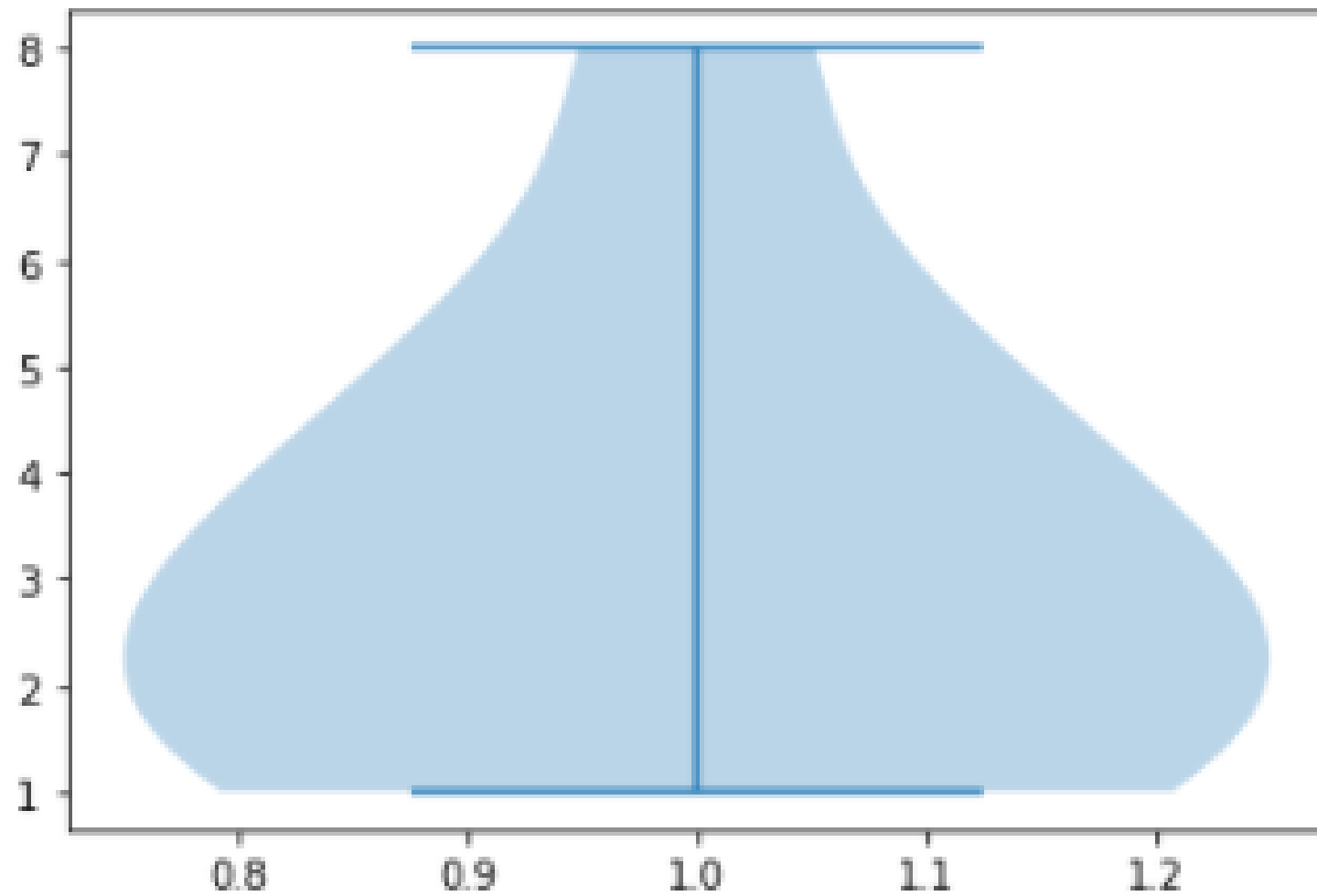
# BOXPLOT

**REPRESENTA DATOS POR  
MEDIO DE CUARTILES,  
INDICA MEDIANA  
BIGOTES- LIMITES**

boxplot(x)



# VIOLIN



**VISUALIZAR DISTRIBUCIÓN  
DE DATOS Y DENSIDAD DE  
PROBABILIDAD.**

# SEABORN

## EJEMPLO: IRIS SPECIES

```
import pandas as pd
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

[HTTPS://WWW.KAGGLE.COM/UCIML/IRIS](https://www.kaggle.com/uciml/iris)

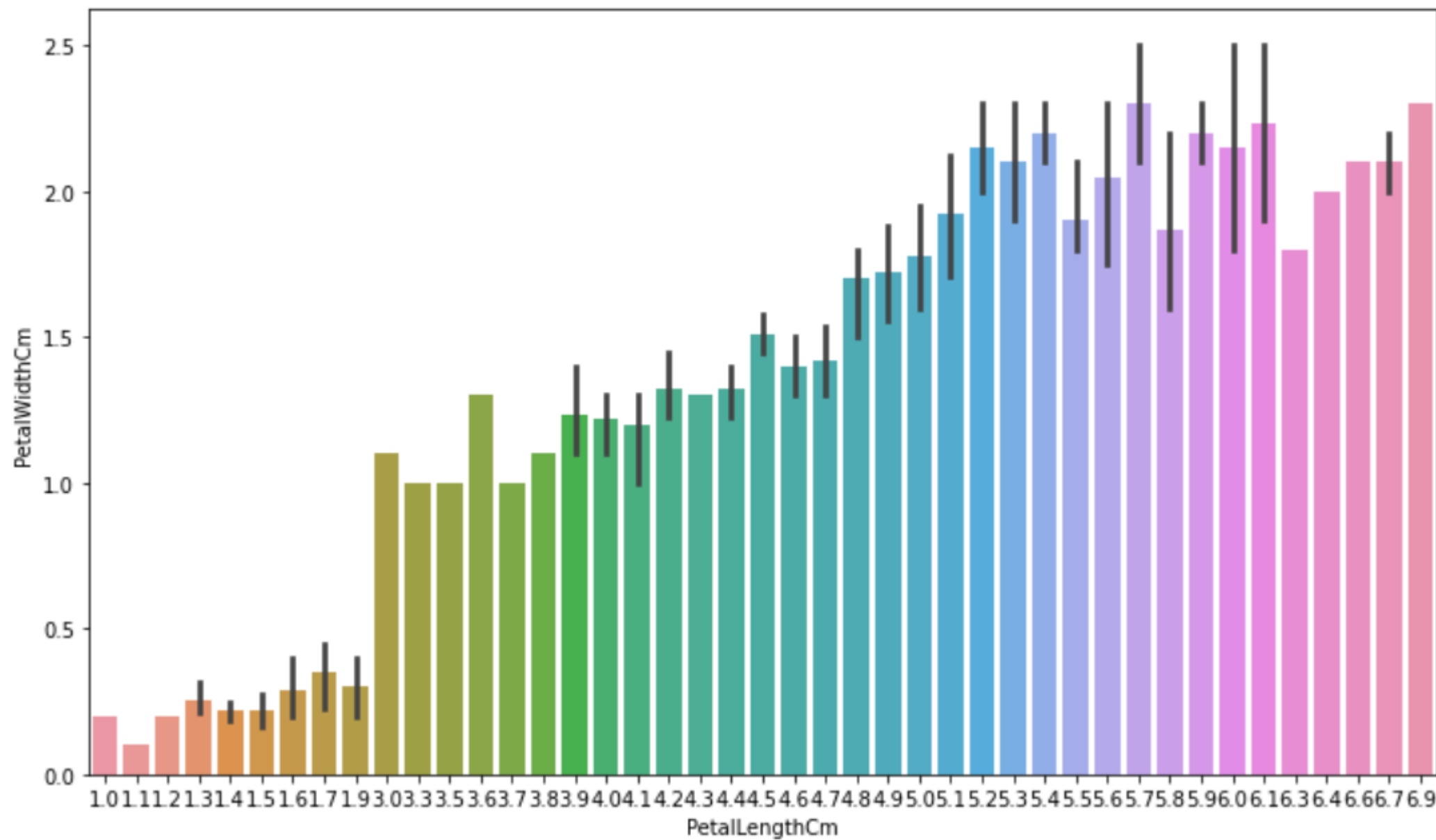
```
iris_filepath = "C:/Users/rbote/Downloads/Iris.csv"

#Asginar una variable para la lectura del archivo
iris_data = pd.read_csv(iris_filepath, index_col="Id")
iris_data.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
Id					
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa

# GRÁFICO DE BARRAS

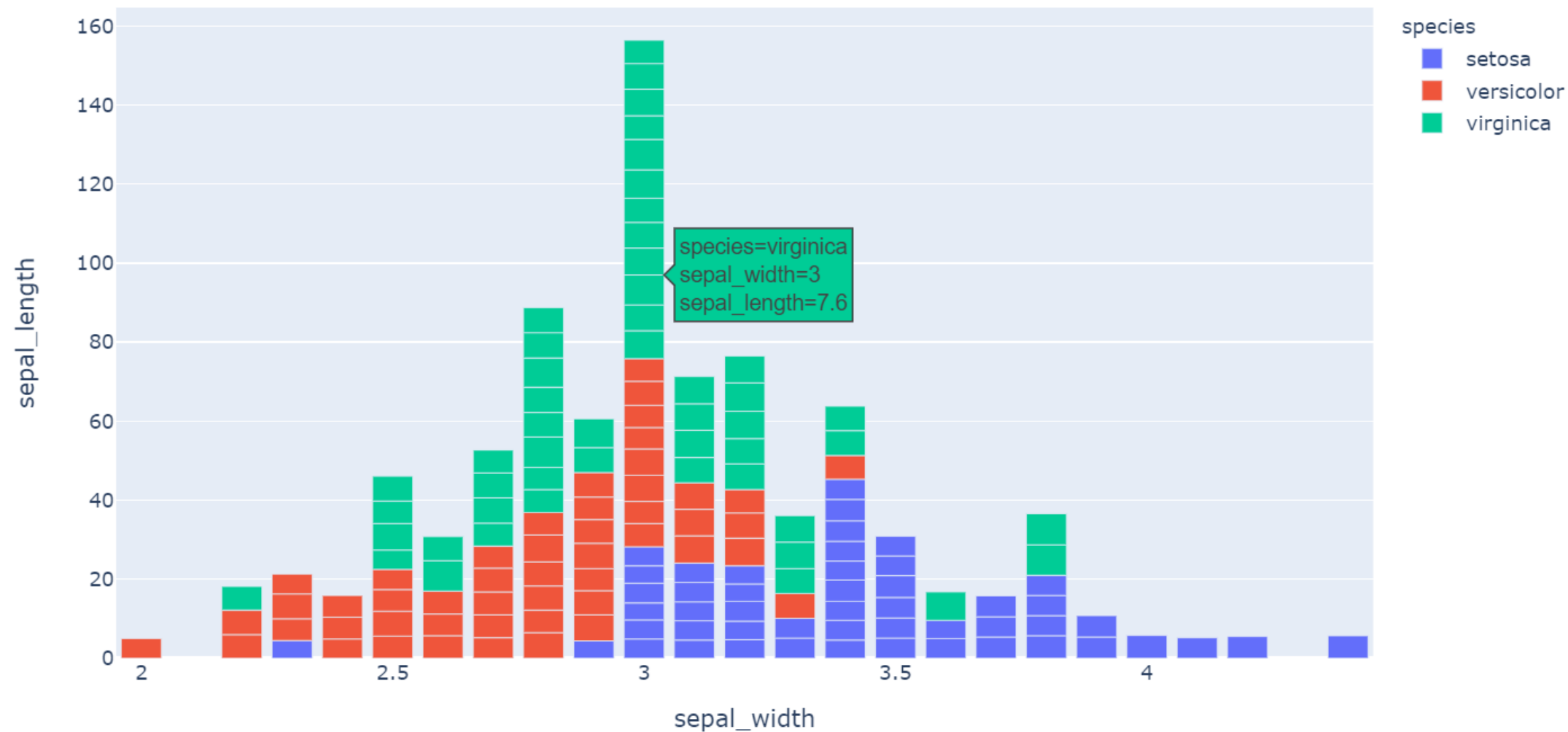
```
sns.barplot(x=iris_data['PetalLengthCm'], y=iris_data['PetalWidthCm'])
```



NOTA: plt.figure(figsize = (x,y))

```
!python -m pip install --upgrade pip
!pip3 install plotly
import plotly
plotly.__version__
```

```
long_df = px.data.iris()
fig = px.bar(long_df,
              x="sepal_width",
              y="sepal_length",
              color="species")
fig.show()
```

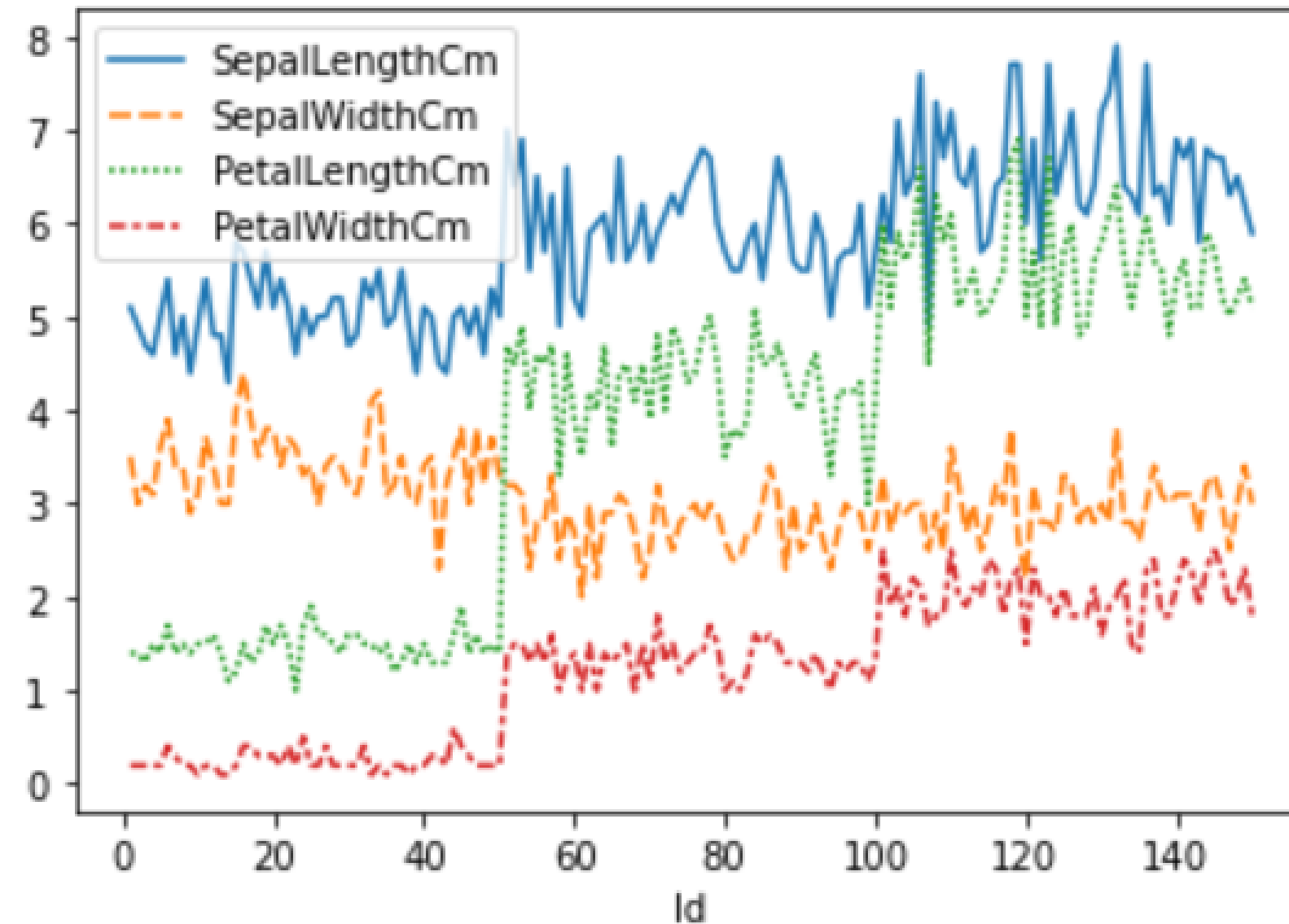




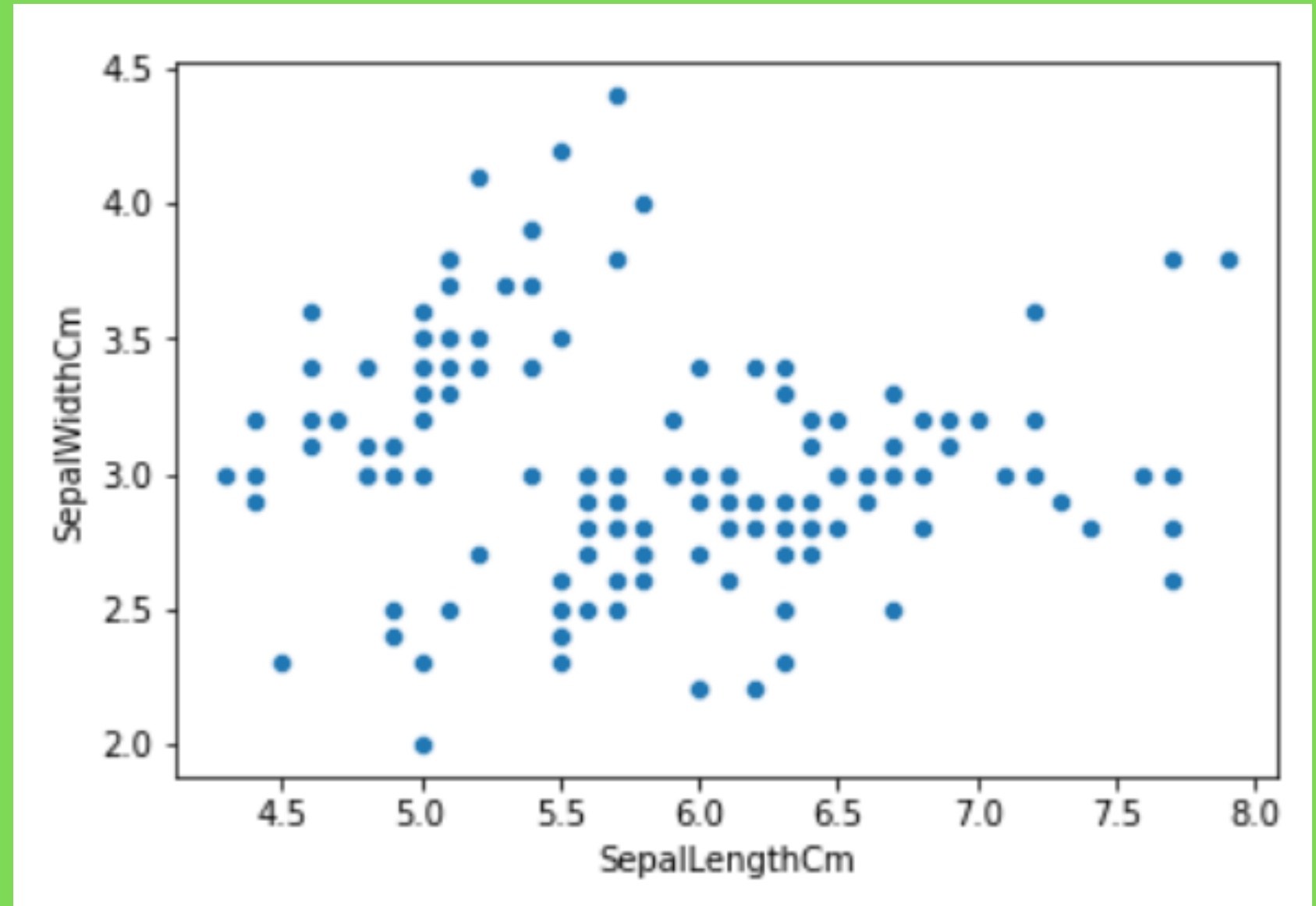
# GRÁFICO DE LINEAS

```
# Gráfico de Lineas  
sns.lineplot(data=iris_data)
```

<AxesSubplot: xlabel='Id'>



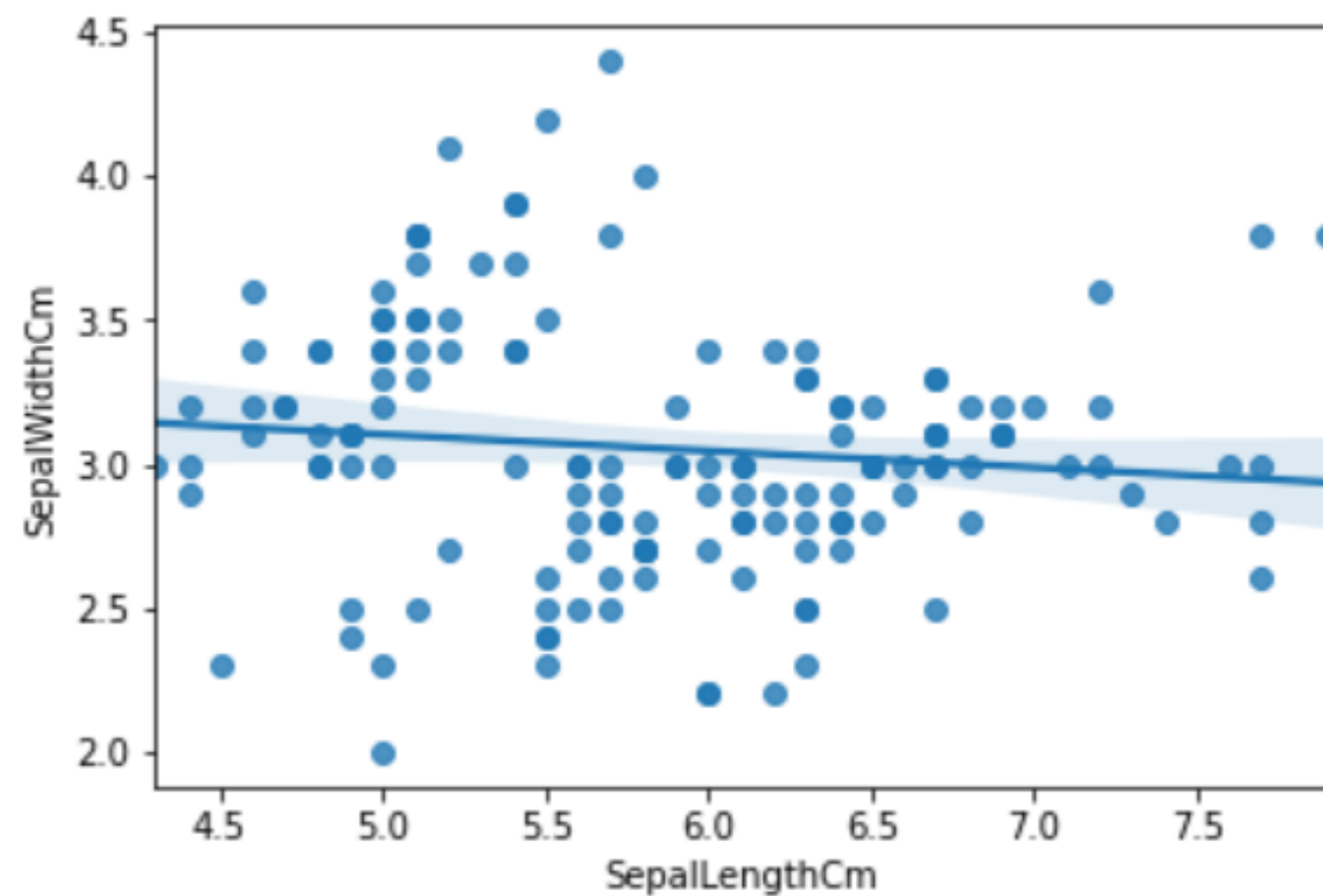
# GRÁFICO DE DISPERSIÓN



```
sns.scatterplot(x=iris_data['SepalLengthCm'], y=iris_data['SepalWidthCm'])
```

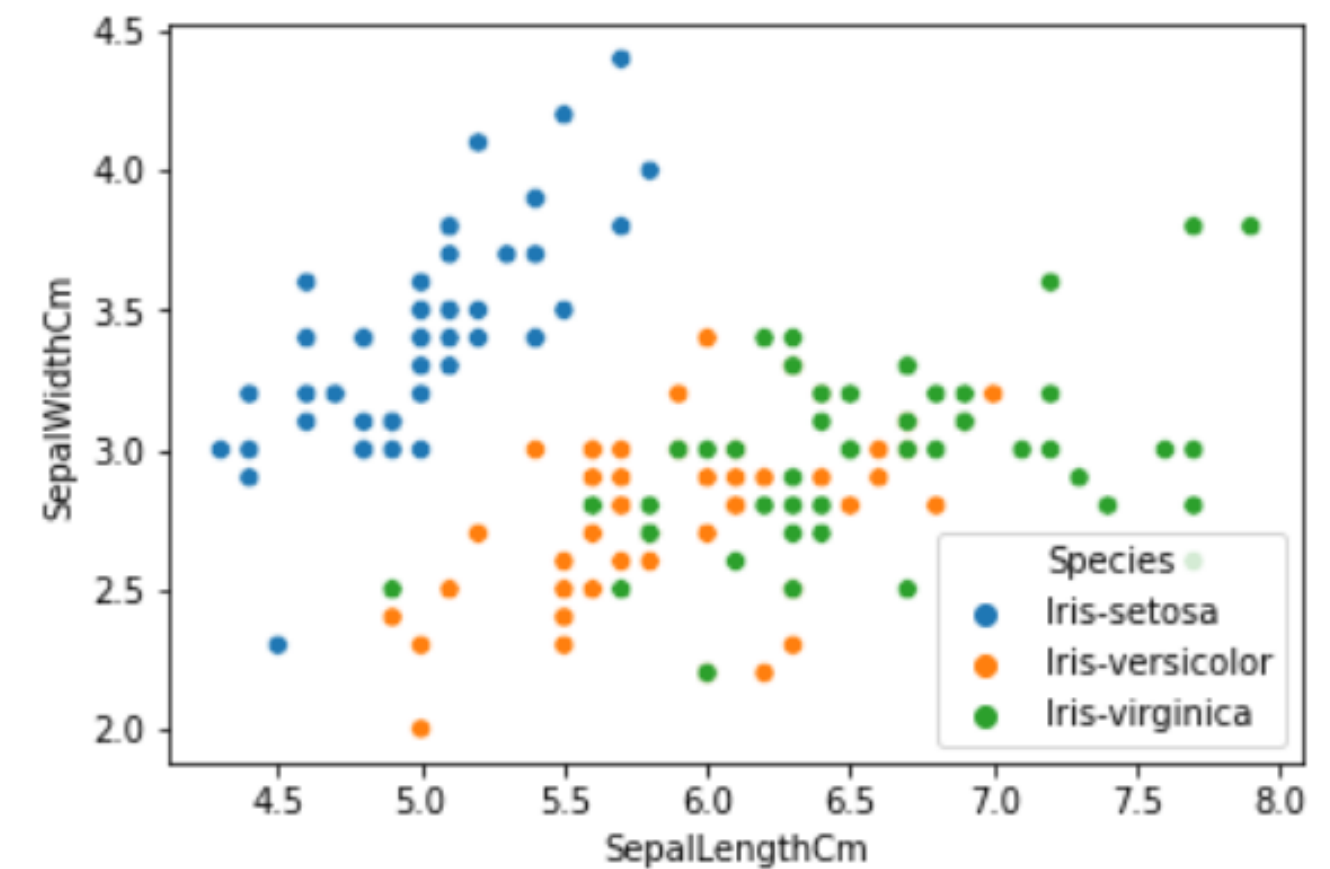
```
sns.regplot(x=iris_data['SepalLengthCm'],  
            y=iris_data['SepalWidthCm'])
```

<AxesSubplot:xlabel='SepalLengthCm', ylabel='SepalWidthCm'>



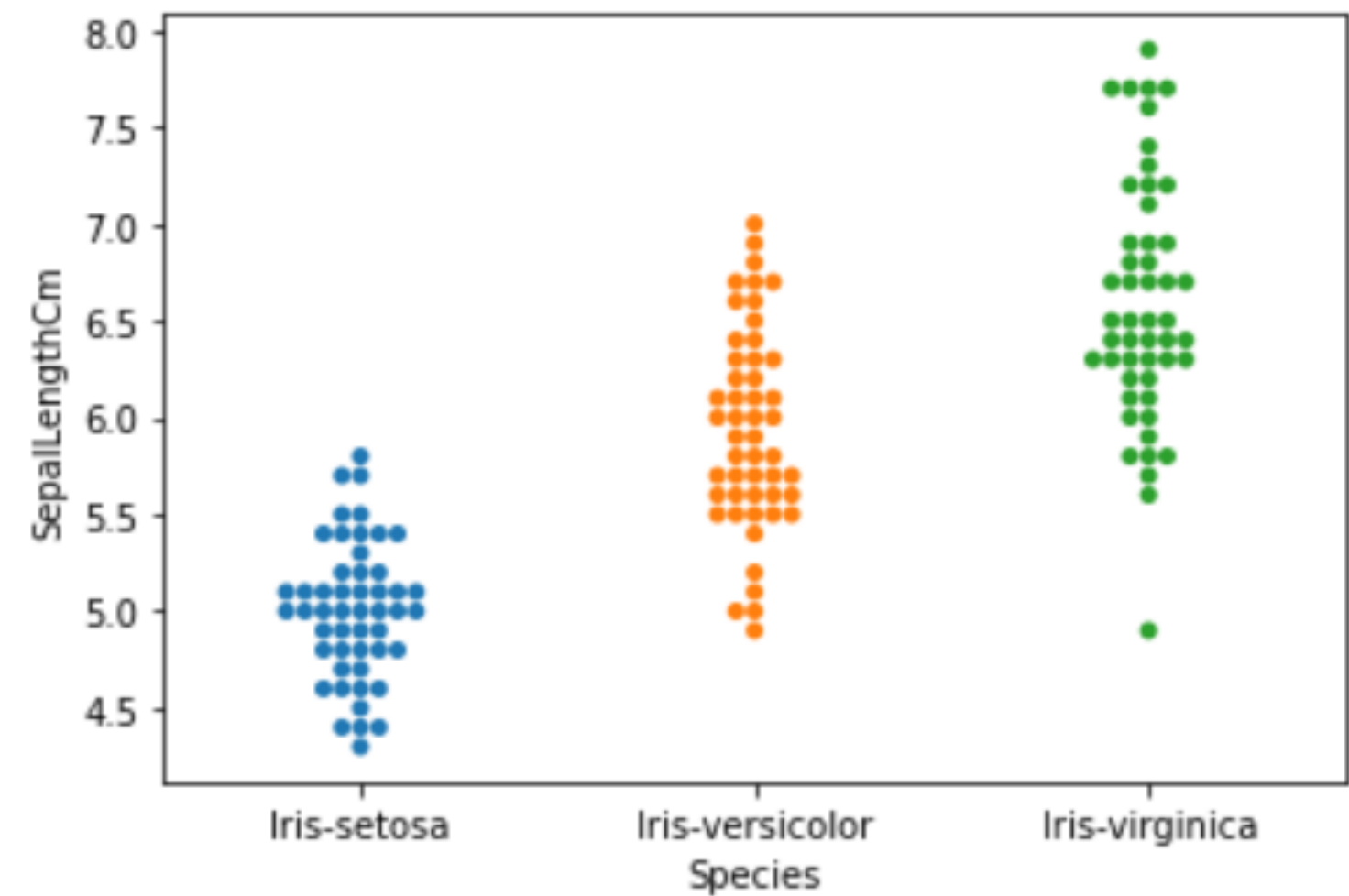
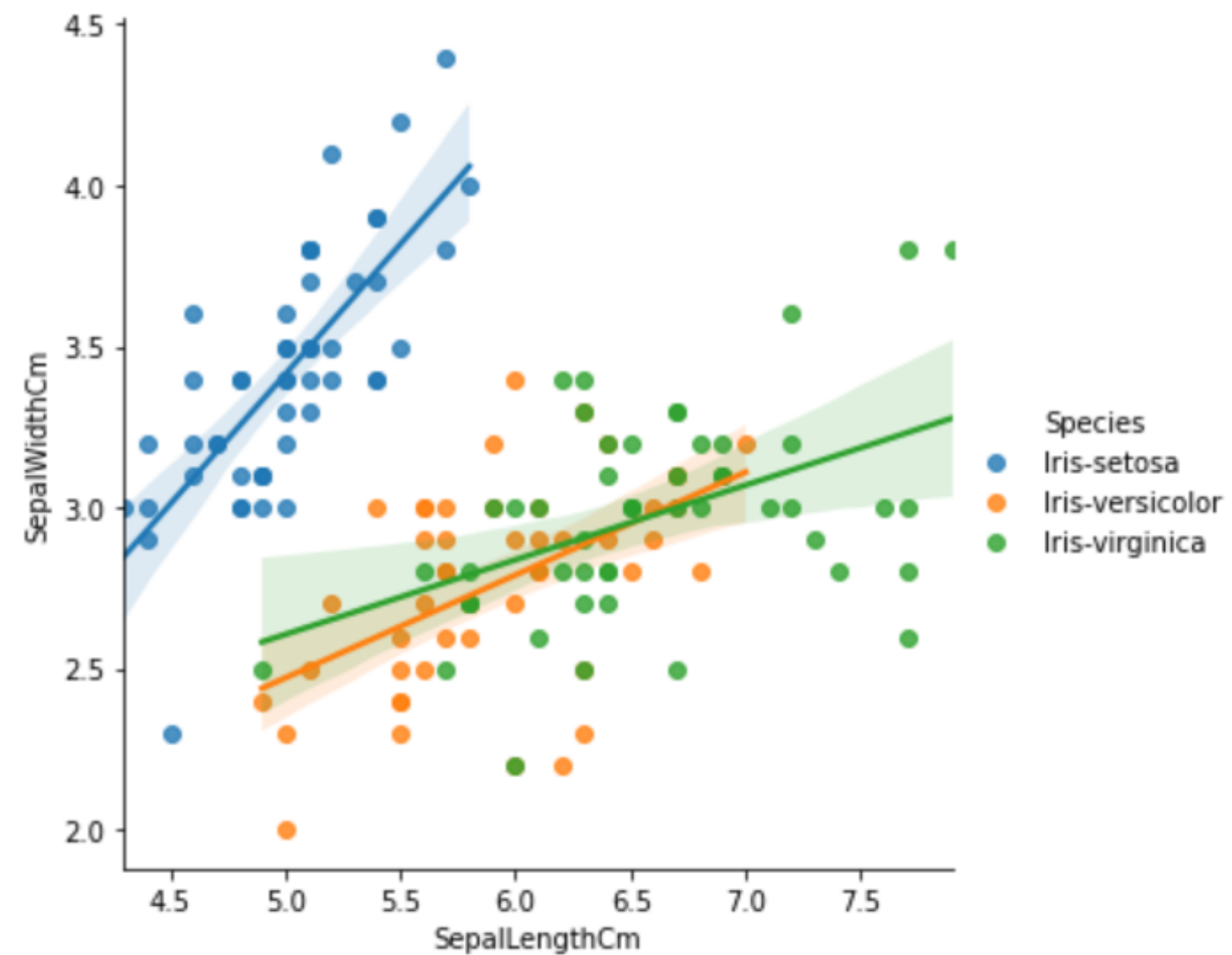
```
sns.scatterplot(x=iris_data['SepalLengthCm'],  
                y=iris_data['SepalWidthCm'],  
                hue=iris_data['Species'])
```

<AxesSubplot:xlabel='SepalLengthCm', ylabel='SepalWidthCm'>



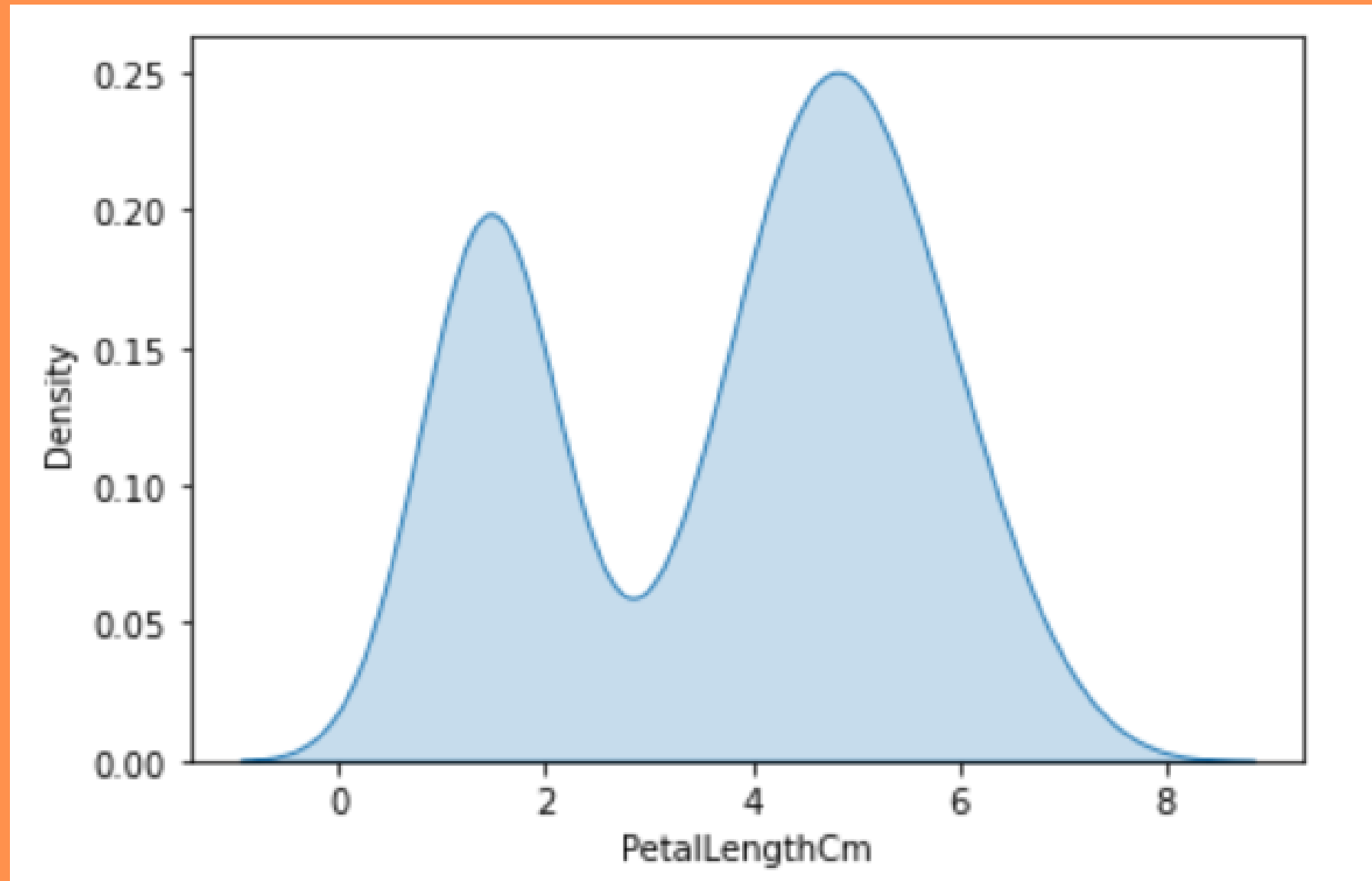
```
sns.lmplot(x='SepalLengthCm', y='SepalWidthCm',  
          hue='Species', data = iris_data)
```

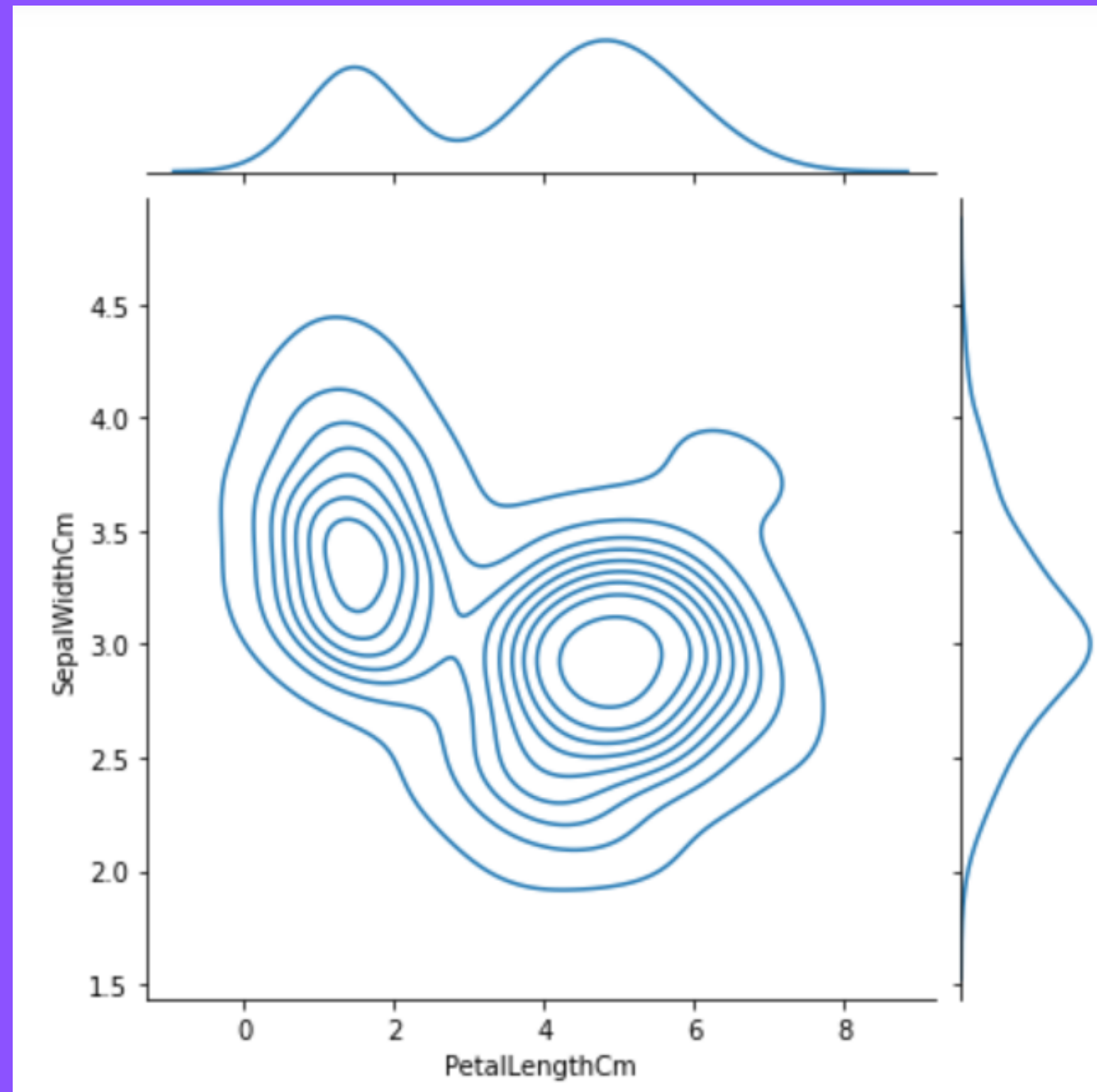
```
sns.swarmplot(x=iris_data['Species'],  
             y=iris_data['SepalLengthCm'],  
             data = iris_data)
```



# GRÁFICO DE DENSIDAD

```
# KDE plot  
sns.kdeplot(data=iris_data['PetalLengthCm'], shade=True)
```



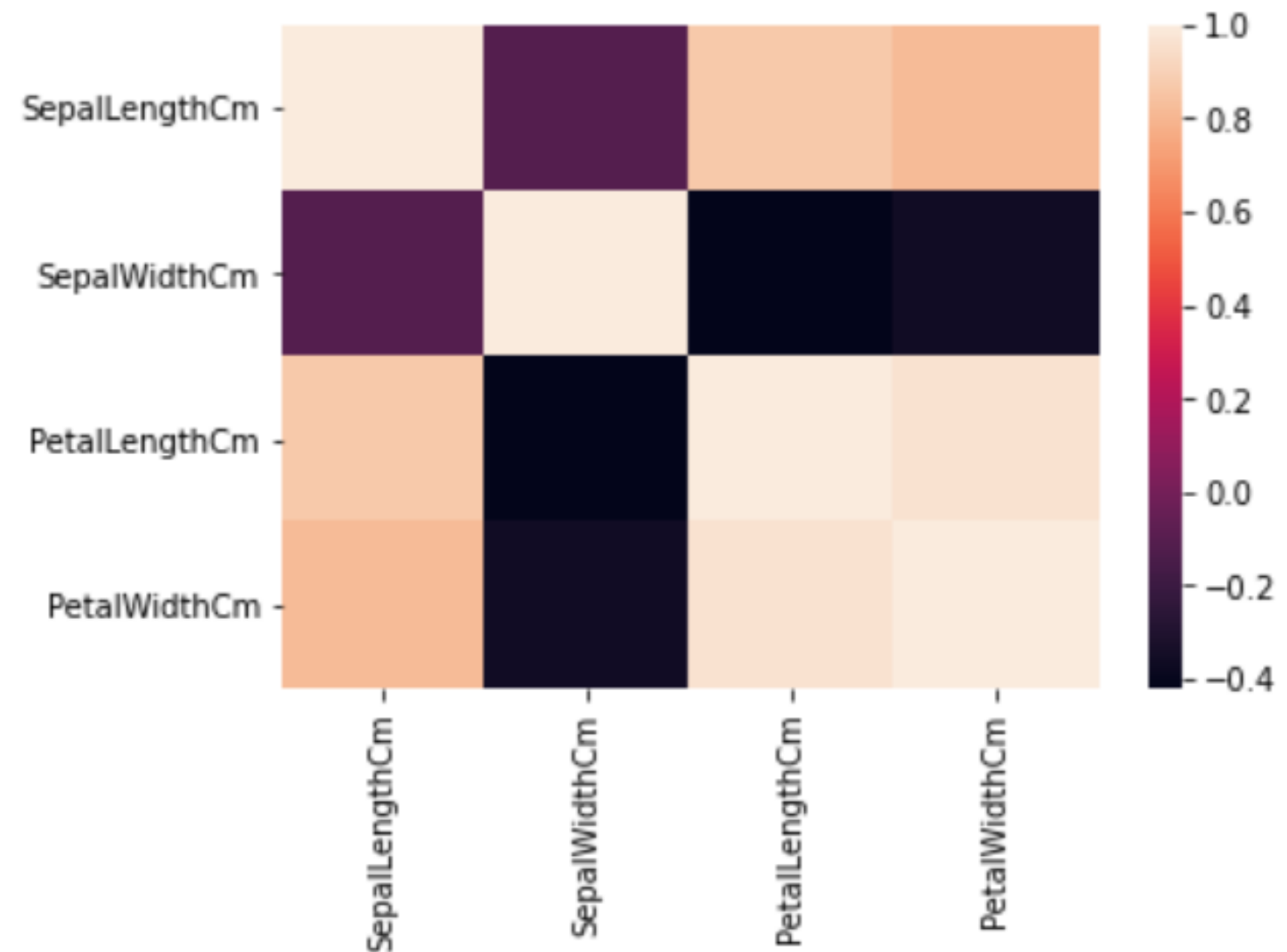


# GRÁFICO DE DENSIDAD DE DOS VARIABLES 2D

```
# 2D KDE plot  
sns.jointplot(x=iris_data['PetalLengthCm'],  
              y=iris_data['SepalWidthCm'], kind="kde")
```

# MAPA DE CALOR

```
sns.heatmap(iris_data.drop(['Species'],axis = 1).corr())
```



# EJEMPLO: US POLICE SHOOTINGS

```
import pandas as pd
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

[HTTPS://WWW.KAGGLE.COM/AHSEN1330/US-POLICE-SHOOTINGS](https://www.kaggle.com/AHSEN1330/US-POLICE-SHOOTINGS)



```

poli_filepath = "C:/Users/enriq/Desktop/shootings.csv"

# Leer el archivo y guardar en variable poli_data
poli_data = pd.read_csv(poli_filepath, index_col="date", parse_dates=True)

poli_data.head()

```

	id	name	manner_of_death	armed	age	gender	race	city	state	signs_of_mental_illness	threat_level	flee	body_camera	arms_category
date														
2015-01-02	3	Tim Elliot	shot	gun	53.0	M	Asian	Shelton	WA	True	attack	Not fleeing	False	Guns
2015-01-02	4	Lewis Lee Lembke	shot	gun	47.0	M	White	Aloha	OR	False	attack	Not fleeing	False	Guns
2015-01-03	5	John Paul Quintero	shot and Tasered	unarmed	23.0	M	Hispanic	Wichita	KS	False	other	Not fleeing	False	Unarmed
2015-01-04	8	Matthew Hoffman	shot	toy weapon	32.0	M	White	San Francisco	CA	True	attack	Not fleeing	False	Other unusual objects
2015-01-04	9	Michael Rodriguez	shot	nail gun	39.0	M	Hispanic	Evans	CO	False	attack	Not fleeing	False	Piercing objects

# HISTOGRAMA

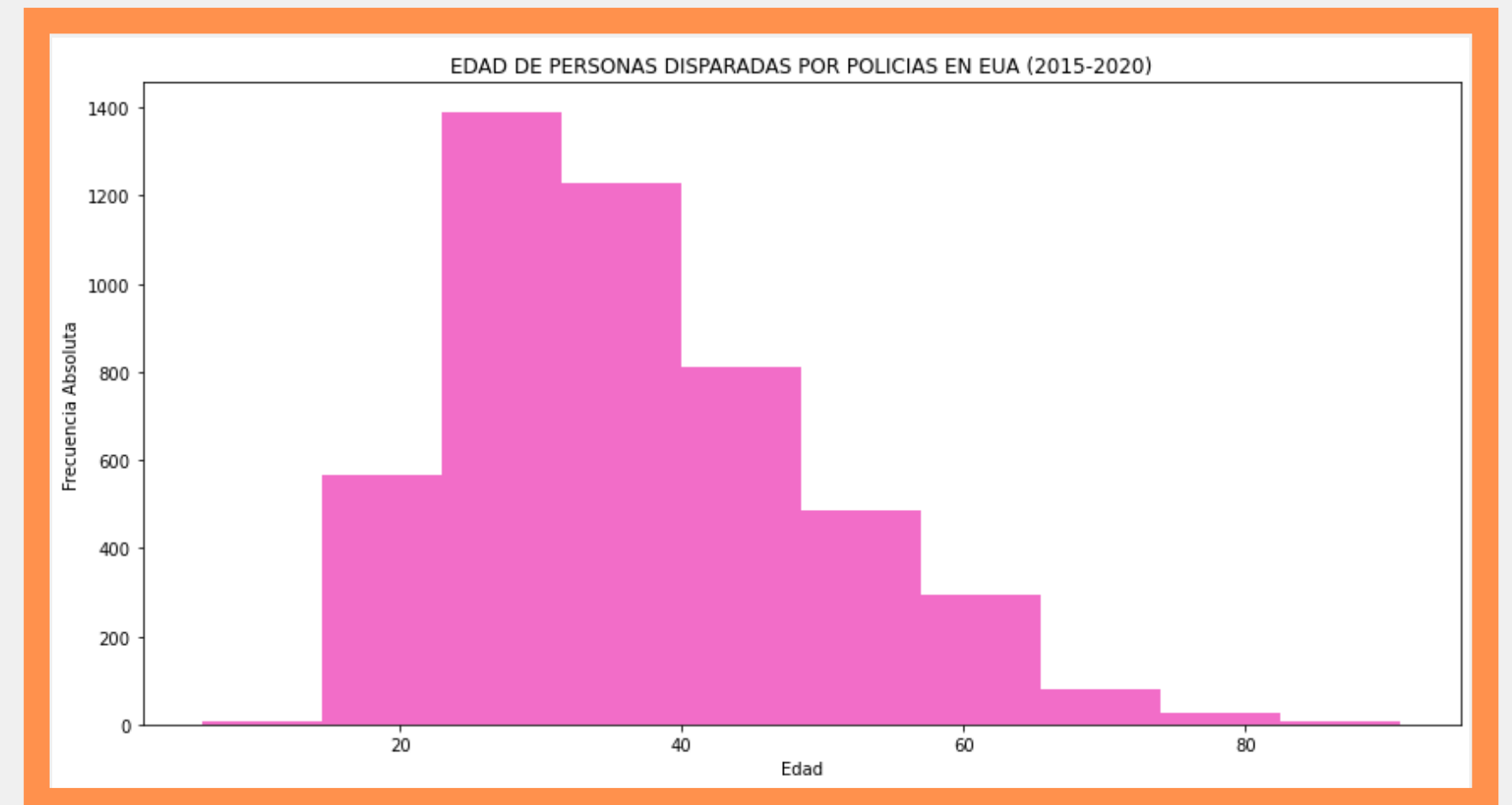
```
import matplotlib as plt
from matplotlib import pyplot

plt.pyplot.figure(figsize=(14,7))

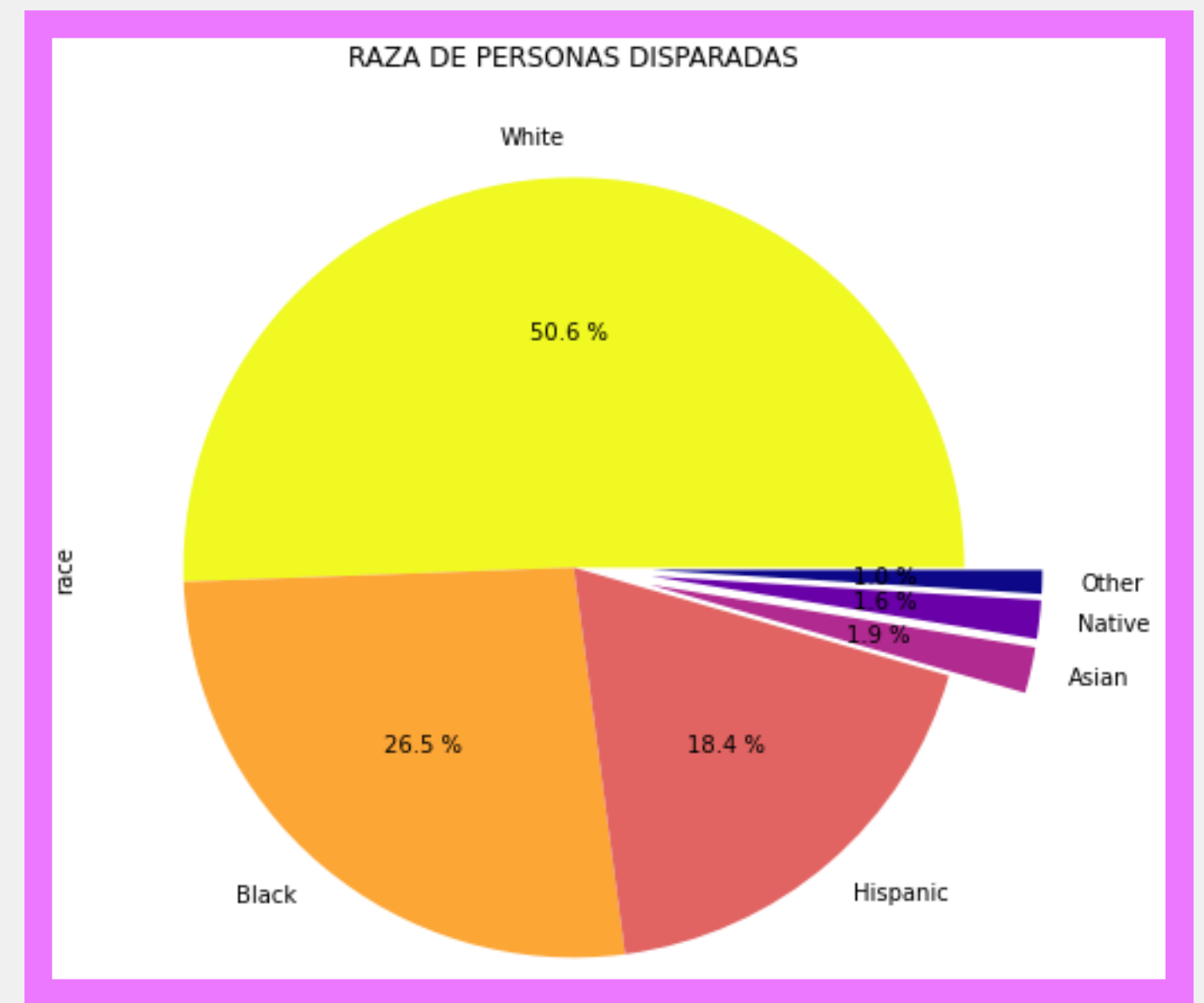
plt.pyplot.hist(poli_data['age'], color='#f26dc8')

plt.pyplot.xlabel('Edad')
plt.pyplot.ylabel("Frecuencia Absoluta")
plt.pyplot.title('EDAD DE PERSONAS DISPARADAS POR POLICIAS EN EUA (2015-2020)')
plt.pyplot.show()
```

```
sns.distplot(a=poli_data['age'], kde=False)
```



# GRÁFICO DE PASTEL



```
from matplotlib import colors
from matplotlib import cm

plt.pyplot.figure(figsize=(8,8))

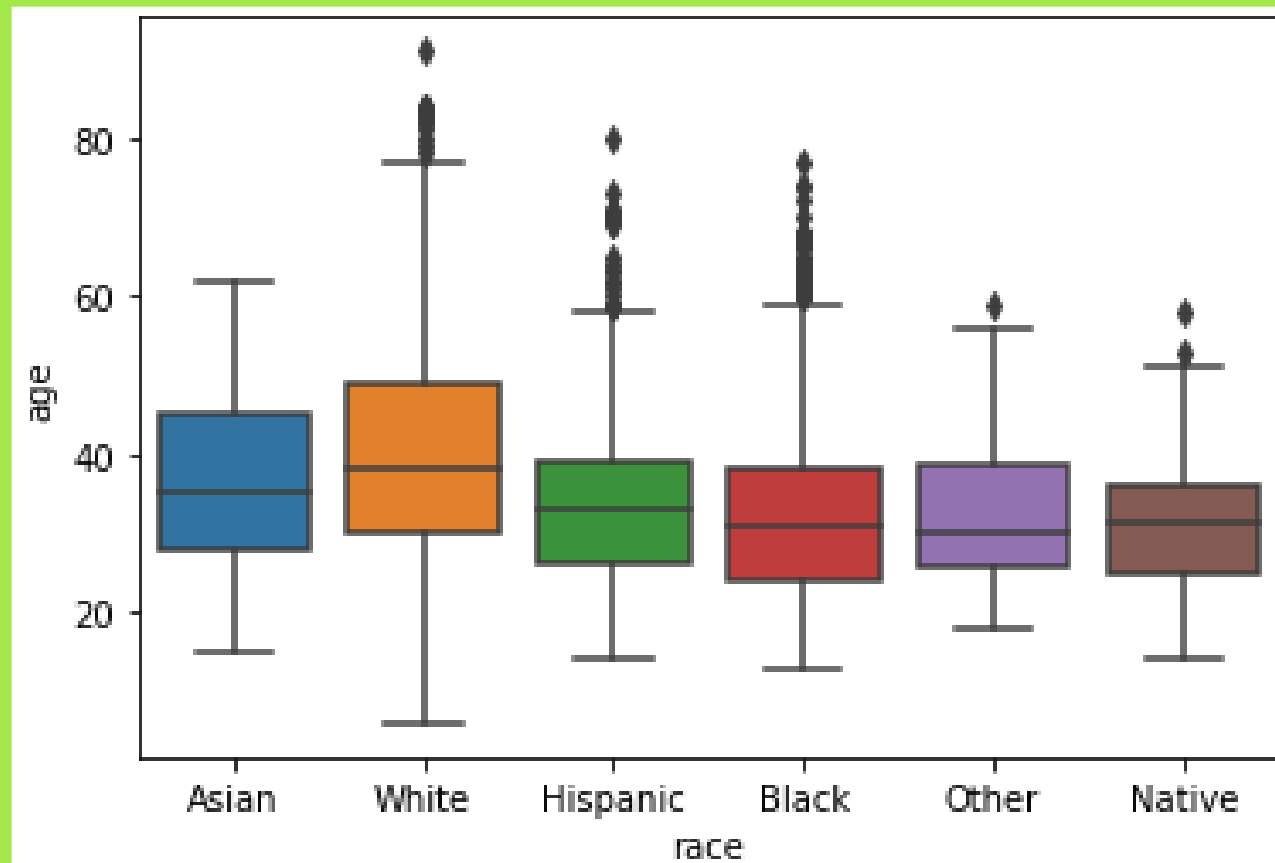
desfase= [0, 0, 0, 0.2, 0.2, 0.2]

poli_data['race'].value_counts().plot.pie(autopct="%0.1f %%", cmap='plasma_r', explode=desfase)
# Otros colormaps: Blues, Accent, prism, rainbow, oranges

plt.pyplot.title('RAZA DE PERSONAS DISPARADAS')
plt.pyplot.show()
```

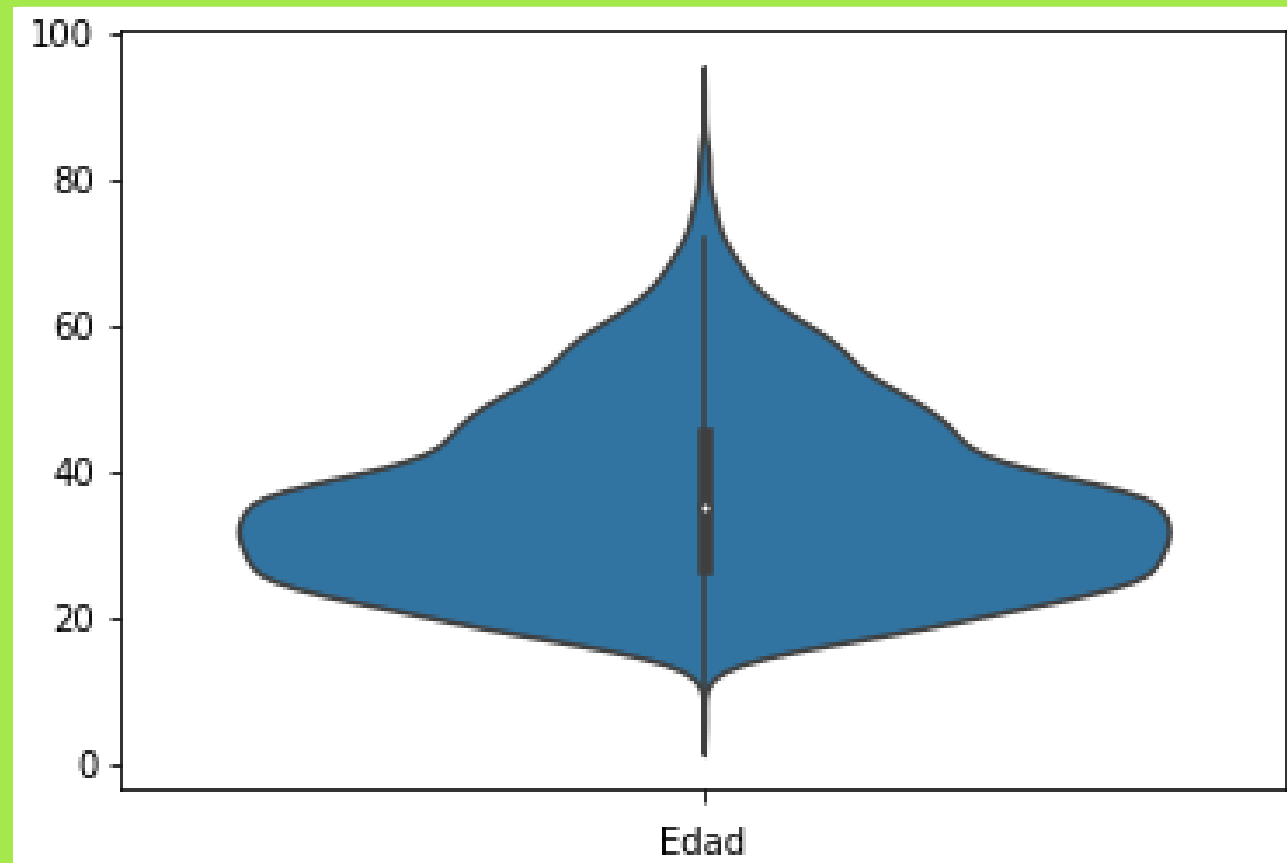
# BOXPLOT

```
import seaborn as sns
sns.boxplot(x='race', y='age', data=poli_data)
plt.pyplot.show()
```



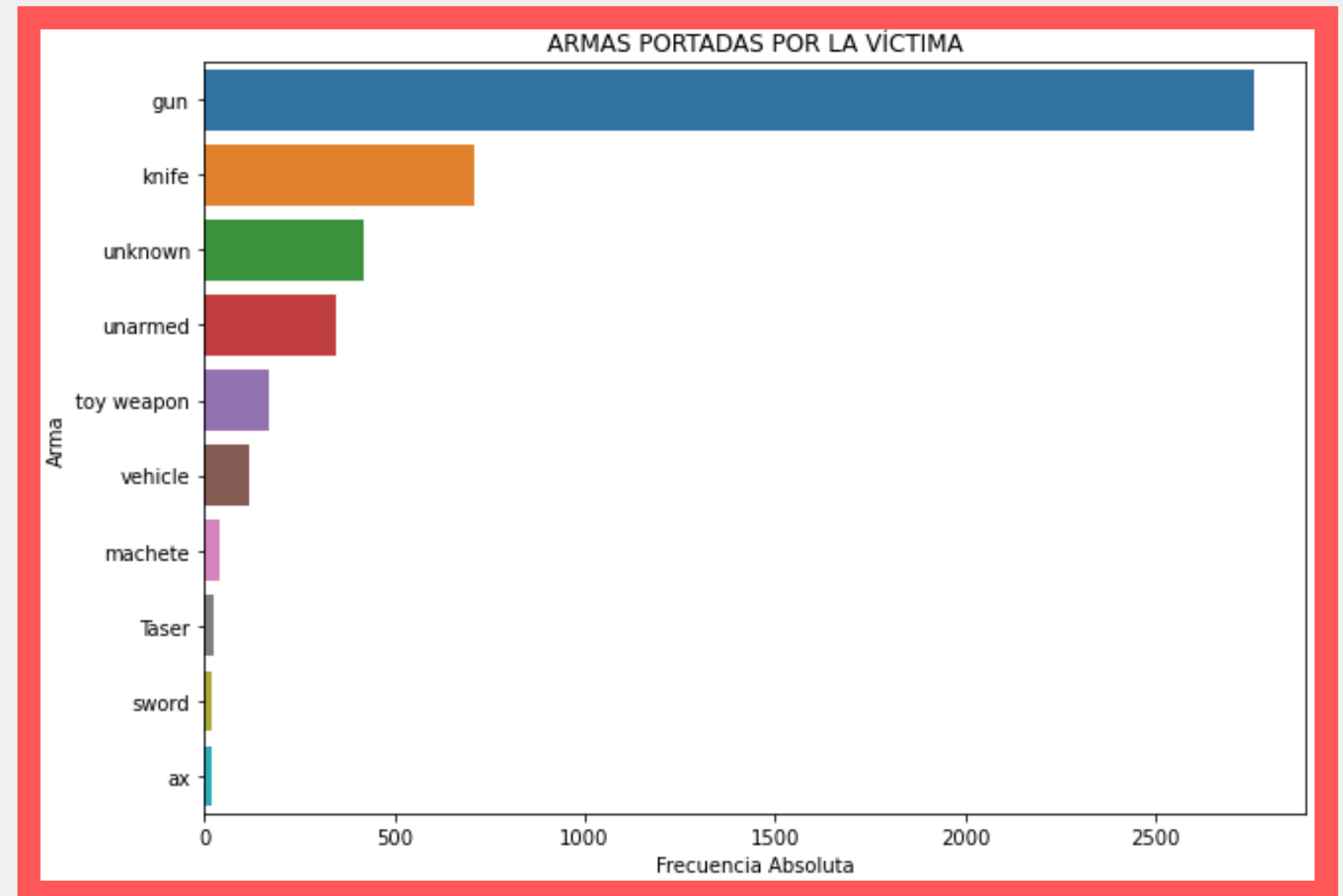
# VIOLIN

```
sns.violinplot(y=poli_data['age'])
plt.pyplot.xlabel("Edad")
plt.pyplot.ylabel(" ")
```



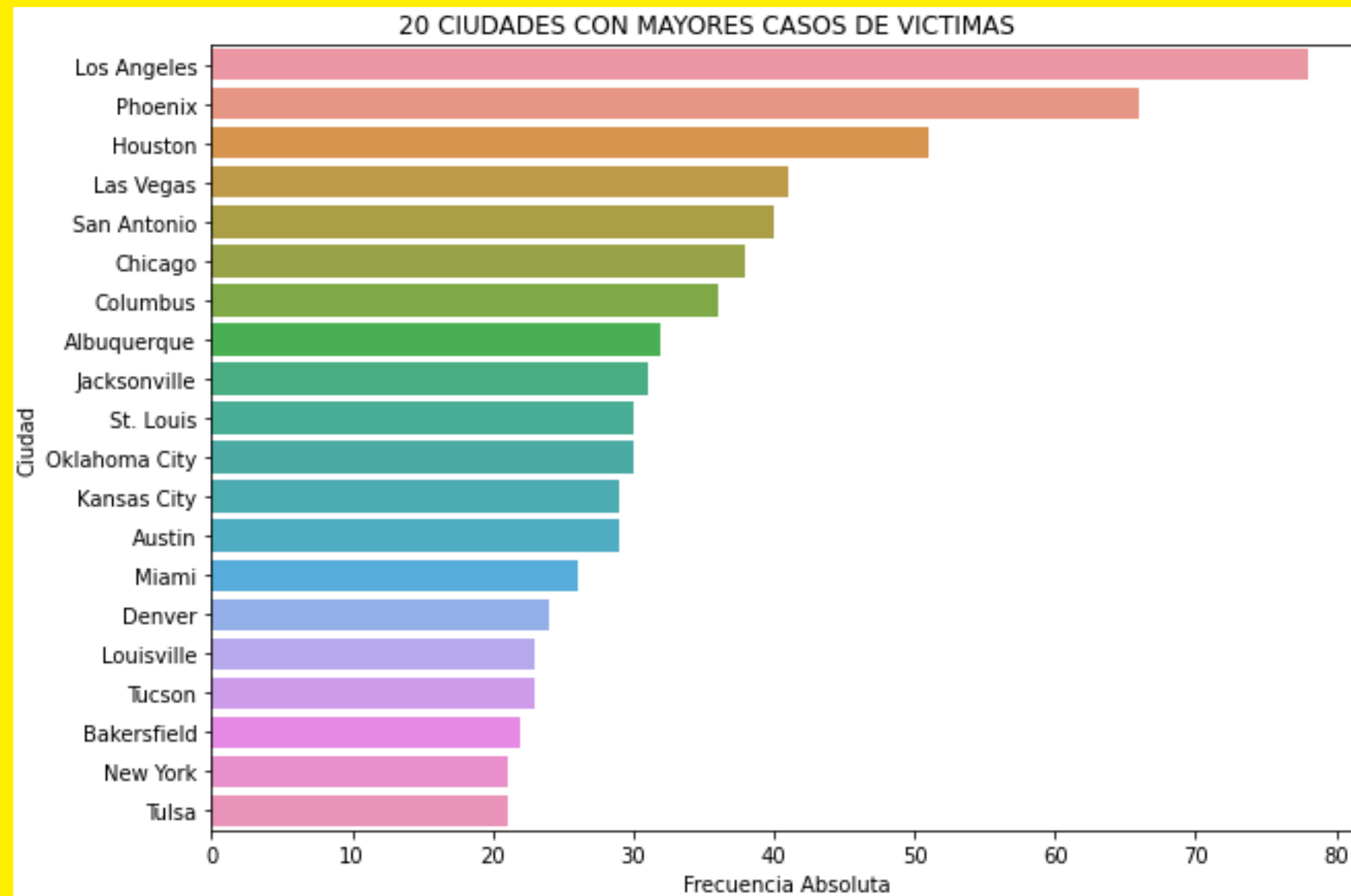
```
plt.pyplot.figure(figsize = (10,7))
chains = poli_data['armed'].value_counts()[:10] #top 10
sns.barplot(x = chains, y = chains.index)
plt.pyplot.title("ARMAS PORTADAS POR LA VÍCTIMA")
plt.pyplot.xlabel("Frecuencia Absoluta")
plt.pyplot.ylabel("Arma")
```

# GRÁFICO DE BARRAS



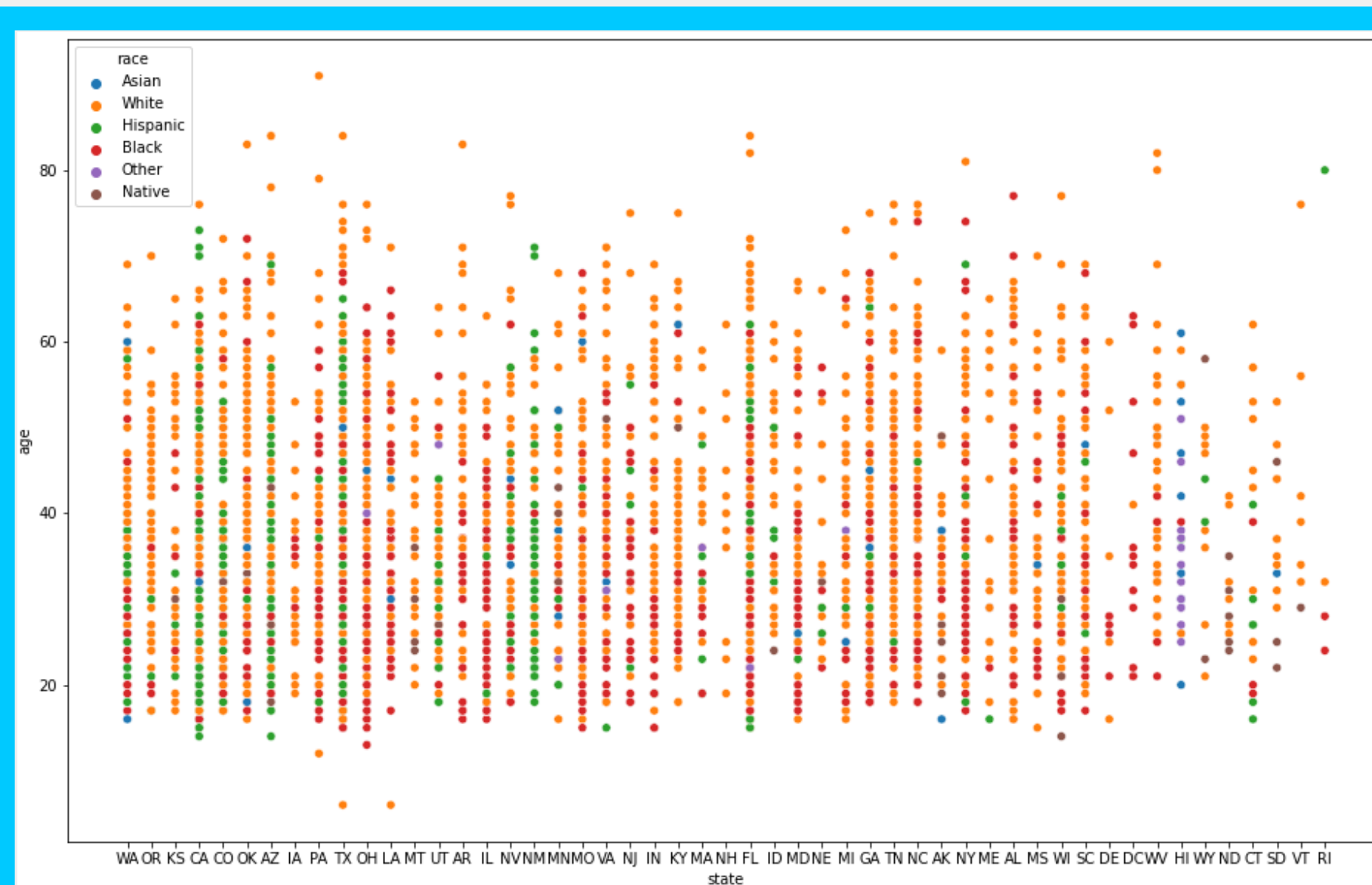
```
plt.pyplot.figure(figsize = (10,7))
chains = poli_data['city'].value_counts()[:20] #top 20
sns.barplot(x = chains, y = chains.index)
plt.pyplot.title("20 CIUDADES CON MAYORES CASOS DE VICTIMAS DE POLICIAS")
plt.pyplot.ylabel("Ciudad")
plt.pyplot.xlabel("Frecuencia Absoluta")
```

```
plt.pyplot.figure(figsize = (10,7))
chains = poli_data['state'].value_counts()[:20] #top 20
sns.barplot(x = chains, y = chains.index)
plt.pyplot.title("20 ESTADOS CON MAYORES CASOS DE VICTIMAS DE POLICIAS")
plt.pyplot.ylabel("Estado")
plt.pyplot.xlabel("Frecuencia Absoluta")
```



# DIAGRAMA DE DISPERSIÓN

```
plt.pyplot.figure(figsize = (16,10))  
sns.scatterplot(x=poli_data['state'], y=poli_data['age'], hue=poli_data['race'])
```



# HEATMAP

