



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
DEPARTAMENTO DE COMPUTACIÓN Y SIMULACIÓN DE SISTEMAS
TÓPICOS ESPECIALES I
TALLER 3



Facilitador: Tomás J. Concepción Miranda

Integrantes:

Gustavo Colucci 8-951-2191

Luis Mejia 8-949-350

Greg Torres 8-956-675

Objetivo

Aprender las funciones y métodos fundamentales de la librería Matplotlib

Indicaciones

Se debe realizar un informe en el que se detalle, para cada problema, el desarrollo de la solución. Este trabajo se puede realizar en grupos de 2 o 3 personas. Envíe su informe en Moodle en formato PDF, así como el código fuente, en el bloque correspondiente antes de las 11:55 p.m. del día siguiente del taller.

Rúbrica

Este laboratorio tendrá una puntuación total de 100, donde la evaluación se basa en los aspectos de excelente, bueno, regular, deficiente. Los puntos que se evaluarán en la rúbrica se muestran en la tabla:

N.	Aspectos a evaluar				
1	CONTENIDO DE ACUERDO A LO SOLICITADO EN EL ENUNCIADO	Excelente (80)	Bueno (70)	Regular (50)	Deficiente (0)
	Resultados debidamente presentados (capturas de pantalla, explicación) - (80 puntos)				
2	PANTALLA DE PRESENTACIÓN - (10 puntos)	Excelente (10)	Bueno (8)	Regular (5)	Deficiente (0)
	Cumple con todos los parámetros dados en clase				
3	ENTREGA DE TRABAJO EN LA PLATAFORMA MOODLE - (10 puntos)	Excelente (10)	Bueno (8)	Regular (5)	Deficiente (0)
	Entrega a tiempo en la plataforma				

Enunciados

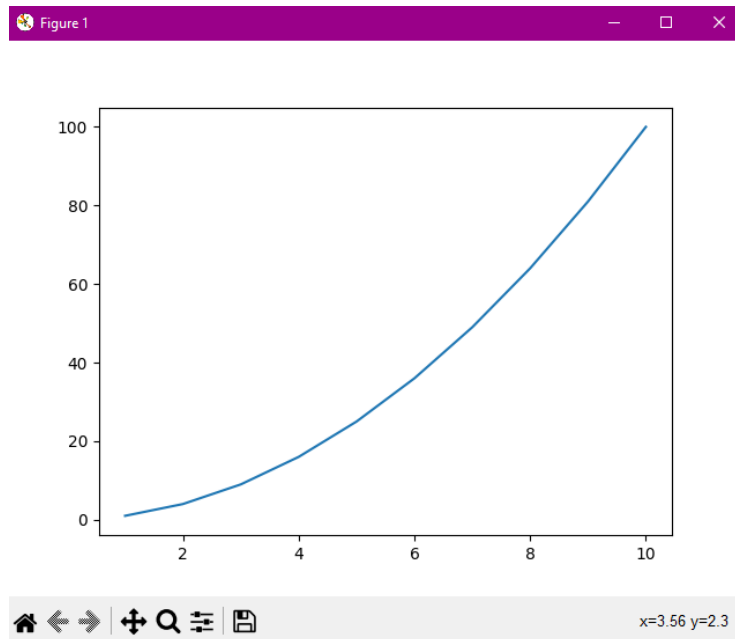
Información General

Matplotlib es una librería “completa para crear visualizaciones estáticas, dinámicas e interactivas en Python”. Entre los diferentes tipos de gráficas disponibles se encuentran: líneas, barras, dispersión, mapa de calor, histogramas, cajas, violín, y gráficas en 3D. La documentación oficial se encuentra disponible en matplotlib.org/stable/api Existen cheatsheets disponibles con datos resumidos de la librería.

Problema 1: Una vez lanzados los comandos, ¿qué figura se obtiene?

```
import matplotlib.pyplot as plt
import numpy as np

fig, ax = plt.subplots() # Crea una figura conteniendo un solo axis
x = np.arange(1,11) # Crea un arreglo del 1 al 10
y = np.array([a**2 for a in x]) # Crea un arreglo con los cuadrados de x
ax.plot(x, y) # Grafica las variables 'x' y 'y' en el axis
plt.show() # Muestra la figura en pantalla
```

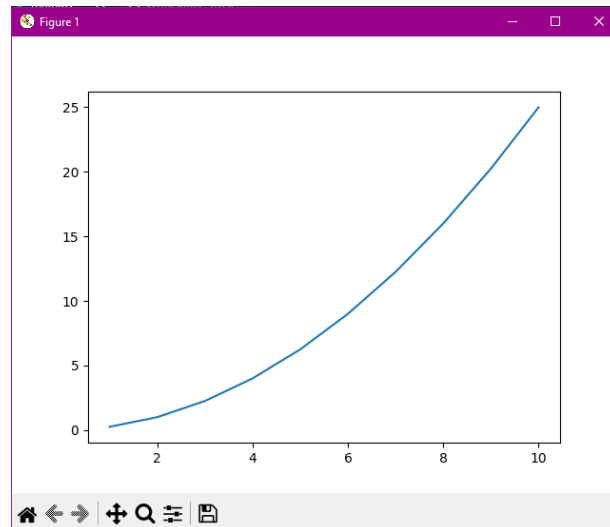


Explicación:

Se puede observar que se creó una ventana nueva la cual tiene los datos en XY sobre un gráfico que es una función x^2 .

Problema 2: agregue la función $g(x) = 0.5x^2$ a la figura (inspirese del ejemplo anterior)

```
fig, ax = plt.subplots() # Crea una figura conteniendo un solo axis
x = np.arange(1,11) # Crea un arreglo del 1 al 10
y = np.array([(0.5*a)**2 for a in x]) # Crea un arreglo con los cuadrados de x
ax.plot(x, y) # Grafica las variables 'x' y 'y' en el axis
plt.show() # Muestra la figura en pantalla'''
```

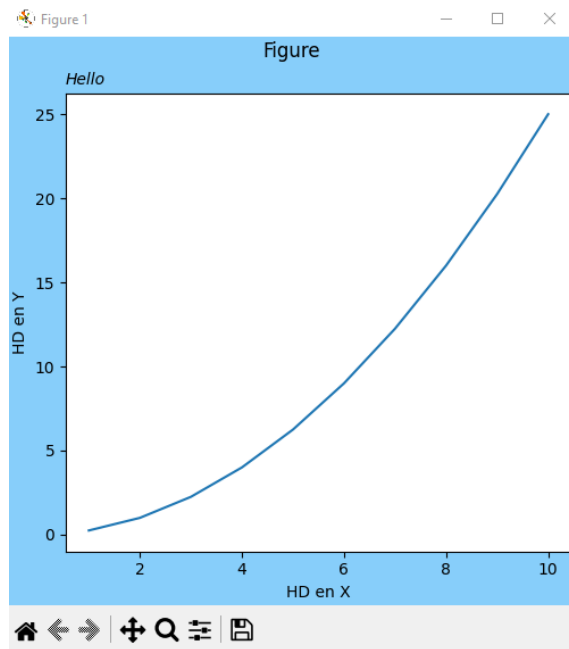


Explicación:

En este caso solo debemos multiplicar la función anterior por 0.5 y esto nos dará un gráfico el cual en x seguirá igual, pero en y ahora su valor final cuando esté en 10 en x en y era 25.

Problema 3: Modifique la figura anterior agregándole un título, etiquetas a los ejes horizontal y vertical, y una leyenda para las dos líneas (es libre de elegir los diferentes nombres)

```
fig, ax = plt.subplots(figsize=(5, 5), facecolor='lightskyblue',
                        layout='constrained')
fig.suptitle('Figure')
ax.set_title('Hello', loc='left', fontstyle='oblique', fontsize='medium')
ax.set_xlabel("HD en X")
ax.set_ylabel("HD en Y")
x = np.arange(1,11) # Crea un arreglo del 1 al 10
y = np.array([(0.5*a)**2 for a in x]) # Crea un arreglo con los cuadrados de x
ax.plot(x, y) # Grafica las variables 'x' y 'y' en el axis
```



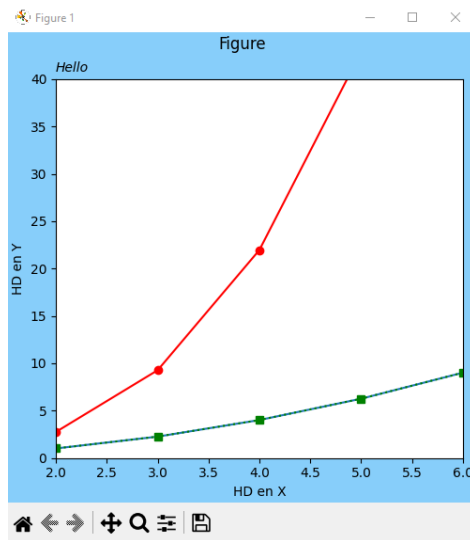
Explicación:

Primero debemos usar la función `subplot` para añadirle el título y la función `set_title` para el subtítulo, luego `set_xlabel` y `set_ylabel` para las etiquetas en dichos ejes.

Problema 4: Modifique la figura anterior cambiando la primera función por cuadros de color verde con una línea punteada y la segunda función por círculos de color rojo con una línea sólida. Muestre solo los datos entre 2 y 6 en horizontal, y entre 0 y 40 en vertical.

```
'''x = np.arange(1,11) # Crea un arreglo del 1 al 10'''
y = np.array([(0.5*a)**2 for a in x]) # Crea un arreglo con los cuadrados de
y2 = np.array([(0.7*a)**3 for a in x]) # Crea un arreglo con los cuadrados de
ax.plot(x, y, "s:g")
ax.plot(x, y2, "o-r")

ax.set_xlim(2,6)
ax.set_ylim(0,40)
```



Explicación:

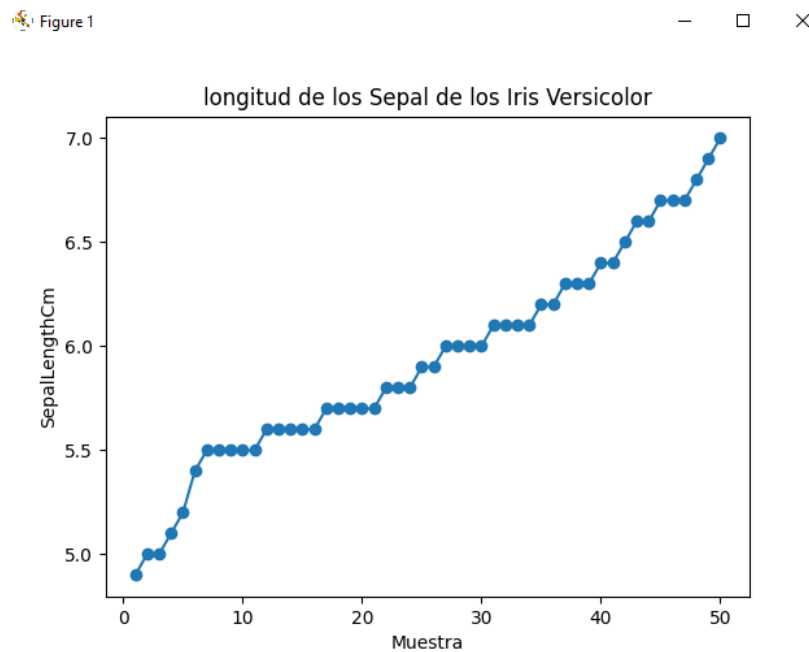
Creamos primero el rango de x como siempre lo hemos estado haciendo, pero a la hora de crear el rango y aquí es donde vamos a variar la función un poco eso para la primera parte. Luego en el `ax.plot` añadimos las coordenadas en x, y para luego darle el formato de “s” para los puntos en cuadrados, dos puntos para decir que es punto a punto y luego la letra g para decirle que es green o verde.

Para el siguiente solo variamos en la Y y en el formato con “o” para decirle que los puntos son circulares, guion para que sea una línea recta y “r” para el color rojo.

Problema 5: Cargue los datos de `iris.csv`. Obtenga un arreglo NumPy del `SepalLengthCm` para la especie `iris versicolor`. Ordene el arreglo de forma ascendente. Haga una gráfica de línea de estos datos. (Tip: para la horizontal, basta con poner números del 1 a la cantidad de entradas del arreglo)

```
C: > Users > 1 > Desktop > Taller2 > Taller 3 > Taller3-problema5.py > ...
1  import pandas as pd
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  iris = pd.read_csv('./iris.csv')
6
7  versicolor = iris[iris["Species"] == "Iris-versicolor"]["SepalLengthCm"]
8  versicolor=np.sort(versicolor)
9  x = np.arange(1, len(versicolor) + 1)
10
11 #gráfica
12 plt.plot(x, versicolor, marker='o', linestyle='-')
13 plt.xlabel('Muestra')
14 plt.ylabel('SepalLengthCm')
15 plt.title('longitud de los Sepal de los Iris Versicolor')
16 plt.show()
```

Resolución del problema 5.



Grafica resultante del problema 5.

Explicación:

Creado Iris, cargamos la variable **versicolor** con los valores de iris de las especies versicolor y solo con los valores de "**SepalLengthCm**" como se aprecia en la línea 7, luego se ordenó usando la función **sort()**. Cargamos x con un arreglo del 1 a la cantidad total de valores hay en el arreglo **versicolor**, línea 9.

Para la gráfica, cargamos la variable **x** en el eje x, y **versicolor** en el eje y. luego se añadió label de xy y un título.

Problema 6: Cargue los datos de iris.csv. Haga una gráfica de dispersión de SepalLengthCm vs SepalWidthCm de la especie iris setosa

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 iris = pd.read_csv('./iris.csv')
6
7 setosa = iris[iris["Species"] == "Iris-setosa"]
8 setosa_length=setosa["SepalLengthCm"]
9 setosa_width=setosa["SepalWidthCm"]
10
11
12 #gráfica
13 plt.scatter(setosa_length, setosa_width)
14 plt.xlabel("length")
15 plt.ylabel("width")
16 plt.title("SepalLengthCm vs SepalWidthCm de la especie iris setosa")
17 plt.show()
```

Resolución del problema 6.

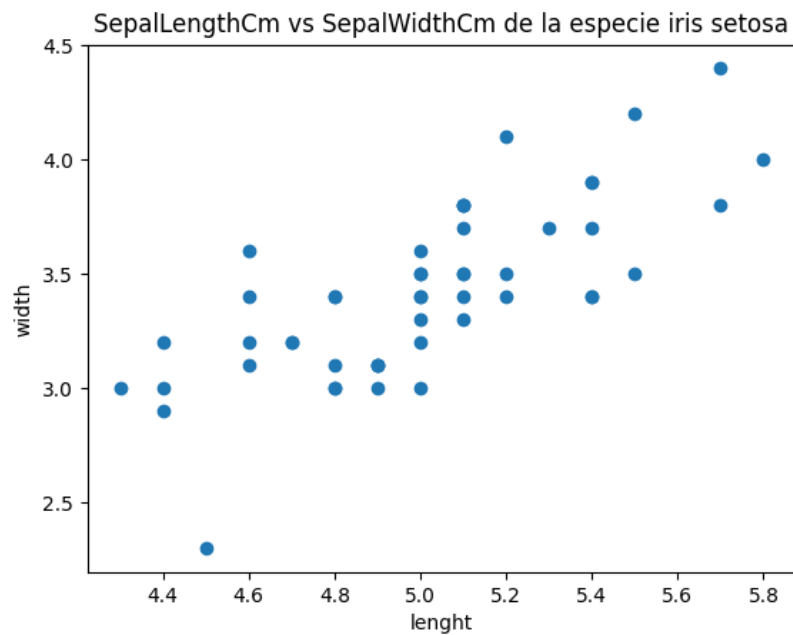


Gráfico resultado del problema 6.

Explicación:

Cargado los datos de iris. Cargamos en la variable **setosa** los valores del DataFrame de iris con todos los datos que la especie sea “**Iris-setosa**”, línea 7, luego en la variable **setosa_length** cargamos solo la columna de “**SepalLengthCm**” y en la variable **setosa_width** cargamos solo la columna de “**SepalWidthCm**” de setosa.

Para la gráfica, usamos la gráfica de dispersión **.scatter()**, la cual cargamos con los 2 arreglos, **setosa_length** y **setosa_width** y agregamos sus labels, y título correspondiente.

Problema 7: Grafique el promedio de SepalLengthCm de cada especie del dataset iris.

```
1  import pandas as pd
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  iris = pd.read_csv('./iris.csv')
6
7  iris_species= iris.groupby("Species")["SepalLengthCm"].mean()
8  promedio=iris_species.values
9  especie=iris_species.index
10 |
11 #gráfica
12 plt.bar(especie,promedio,color="lightblue")
13 plt.xlabel("Especies")
14 plt.ylabel("Promedio")
15 plt.title("Promedio de SepalLengthCm de cada especie")
16 plt.show()
```

Resolución del problema 7.

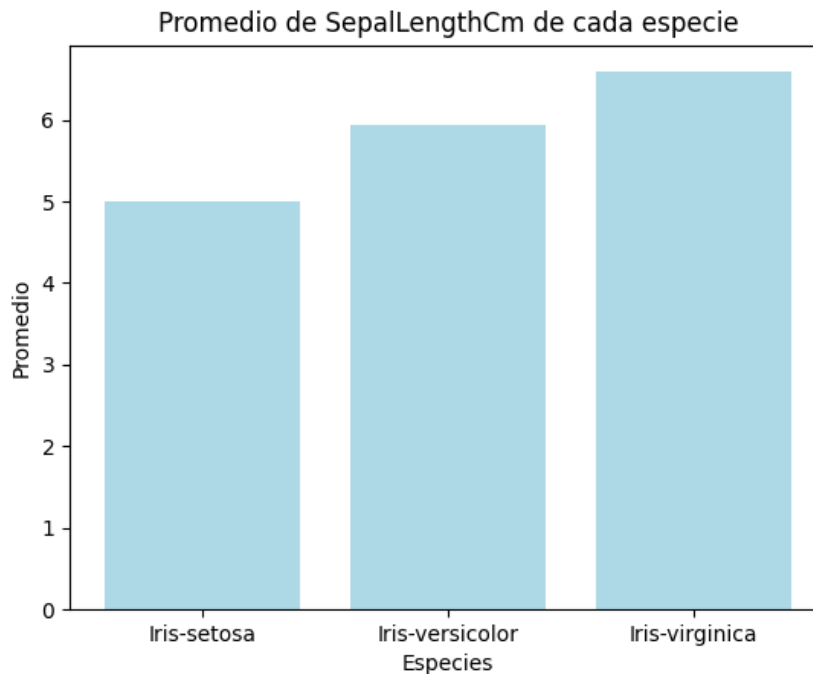


Gráfico resultado del problema 7.

Explicación:

Cargado los datos de iris. En la variable `iris_species`, colocamos los datos del DataFrame de `iris` agrupado por “**Species**” y solo con los valores de la columna “**SepalLengthCm**” y con la función `mean()` extraemos el promedio, línea 7, luego en una variable `promedio` la cargamos con los valores del promedio usando `.values` y otra variable con `especie` usando `.index`.

Para la gráfica, usamos la gráfica de barras usamos `.bar()`, la cual cargamos con las `especies` en el eje x, y los `promedio` en el eje y.

Problema 8: Grafique un histograma de `PetalLengthCm` de la especie iris virginica.

```
print("Problema 8")
iris=pd.read_csv("./iris.csv")
virginica = iris[iris["Species"]=="Iris-virginica"]
fig, ax = plt.subplots()
ax.hist(virginica["PetalLengthCm"],color="Green")
ax.set_title("Histograma de largo del petalo de la especie iris virginica")
ax.set_xlabel("largo del petalo")
ax.set_ylabel("Frecuencia")
plt.show()
```

Resolución del problema 8.

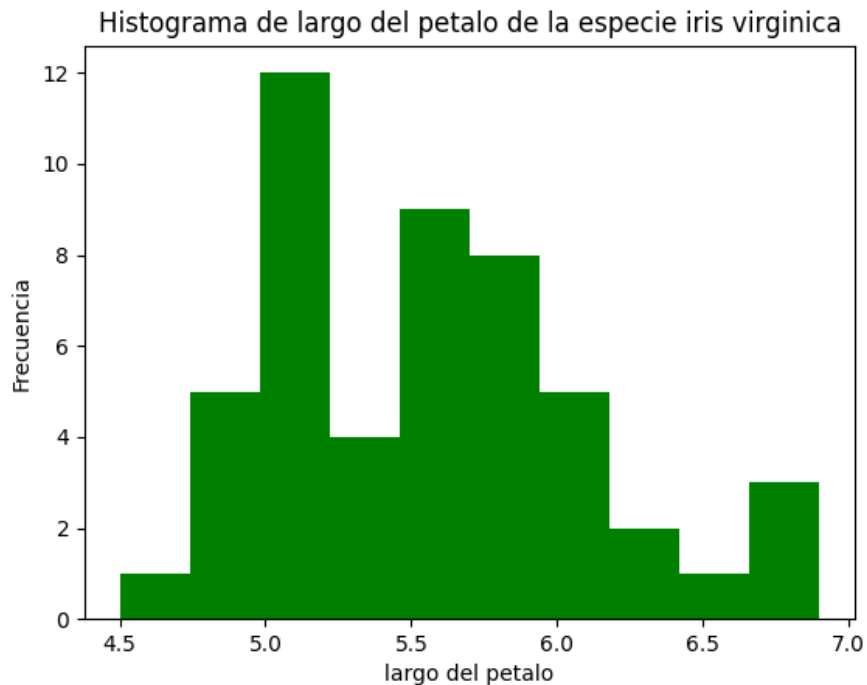


Gráfico resultado del problema 8.

Explicación:

Cargado los datos de iris. En la variable virginica, colocamos los datos del DataFrame de iris que sean de la especie ,línea 7, cuando se crea el histograma lo cargamos solo con los datos de “PetalLengthCm” que estén en virginica, y agregamos el color. Luego se colocan los label y títulos.

Problema 9: Grafique un histograma de PetalLengthCm de la especie iris virginica

```
iris=pd.read_csv("./iris.csv")
print("Problema 9")
virginica = iris[iris["Species"]=="Iris-virginica"]
fig, ax = plt.subplots()
ax.boxplot(virginica["PetalLengthCm"])
ax.set_title("Diagrama de caja del largo del petalo de la especie iris virginica")
ax.set_ylabel("largo del petalo")
ax.set_xlabel("Iris-virginica")
plt.show()
```

Resolución del problema 9.

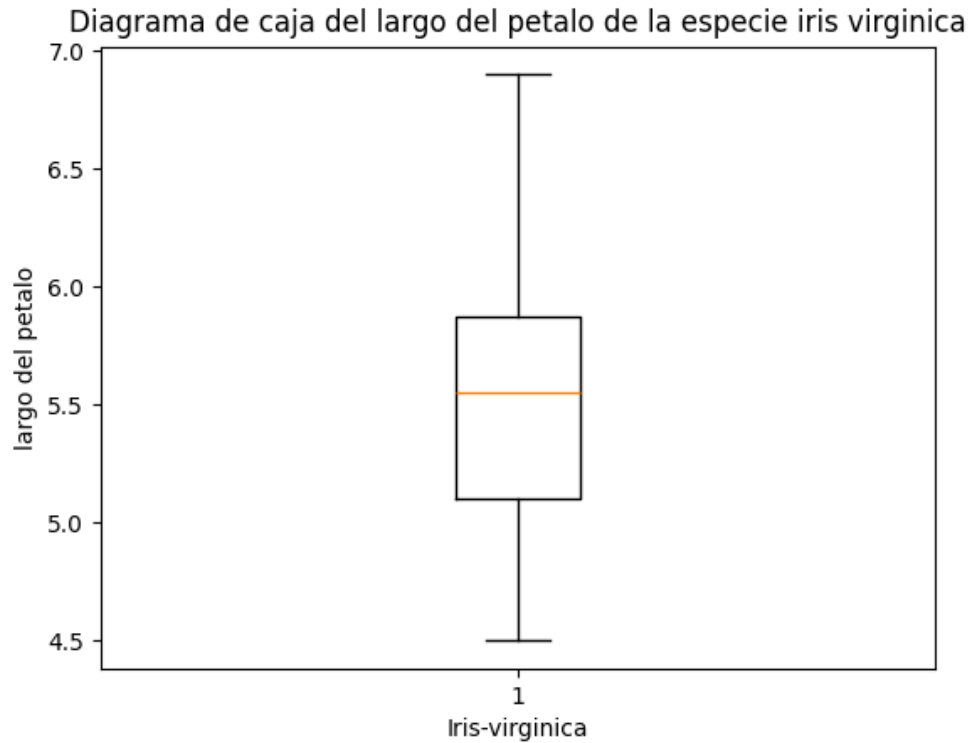


Gráfico resultado del problema 9.

Explicación:

Cargado los datos de iris. En la variable virginica, colocamos los datos del DataFrame de iris que sean de la especie ,línea 7, cuando se crea el diagrama de caja usando `ax.boxplot` lo cargamos solo con los datos de “PetalLengthCm” que estén en virginica. Luego se colocan los label y títulos.

Problema 10: Utilice un juego de datos de su elección, elija un tipo de gráfica diferente a aquellos vistos en problemas anteriores, y grafica atributos de su elección usando el tipo de gráfica seleccionada anteriormente. Explique su resultado.

Explicación:

Creamos una matriz en la variable **datos** con los datos de **iris** separaos por especie. Decidí escoger la gráfica de violín para este ejemplo función que es **violinplot()** la cual cargamos con 3 dataset, uno por cada especie, y aparte solo con los valores de “SepalLengthCm”. Con la función **xticks()** nombre a cada intervalo del eje x y colocándole el nombre de la especie.

```
iris=pd.read_csv("./iris.csv")

datos = {
    'setosa': iris[iris['Species'] == 'Iris-setosa'],
    'versicolor': iris[iris['Species'] == 'Iris-versicolor'],
    'virginica': iris[iris['Species'] == 'Iris-virginica']
}

plt.violinplot([datos['setosa']['SepallengthCm'], datos['versicolor']['SepallengthCm'], datos['virginica']['SepallengthCm']])
plt.xticks([1, 2, 3], ['setosa', 'versicolor', 'virginica'])
plt.title('Longitud de Sépalo por Especie')
```

Resolución del problema 10 (Longitud de sépalo).

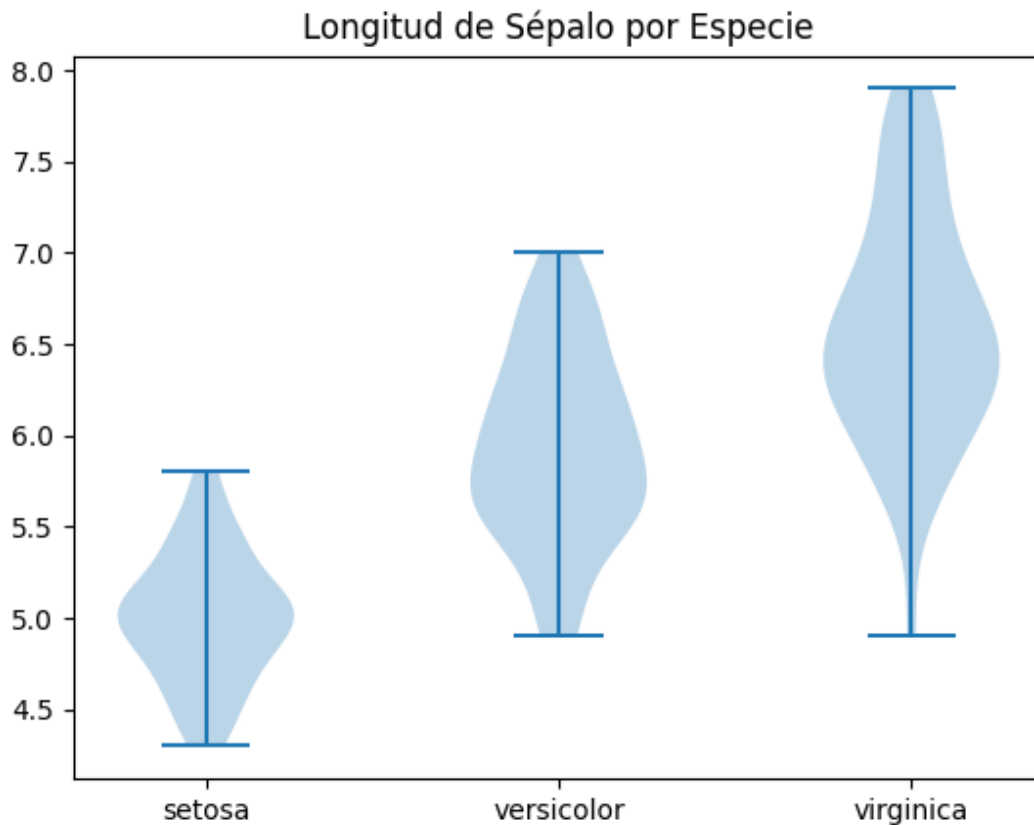


Gráfico-a resultado del problema 10.

Explicación de la gráfica-a: podemos ver el grafico de violín de la longitud de los sépalos de cada uno de las especies. La línea de azul oscuro muestra todo lo que abarca desde el mínimo hasta el máximo valor que alcanzan los sépalos de cada especie, y en azul más claro podemos observar que en las partes mas gruesas significan que es un valor que se repite mas que otros, podemos deducir que donde más grueso será la moda .

```

sepal=plt.violinplot([datos["setosa"]["SepalWidthCm"],datos["versicolor"]["SepalWidthCm"],datos["virginica"]["SepalWidthCm"]])
for sp in sepal["bodies"]:
    sp.set_facecolor("#D43F3A")
    sp.set_edgecolor("red")
plt.xticks([1, 2, 3], ["setosa", "versicolor", "virginica"])
plt.title("Ancho de Sépalo por Especie")
plt.tight_layout()
plt.show()

```

Variación del código del problema 10 (ancho de sépalo).

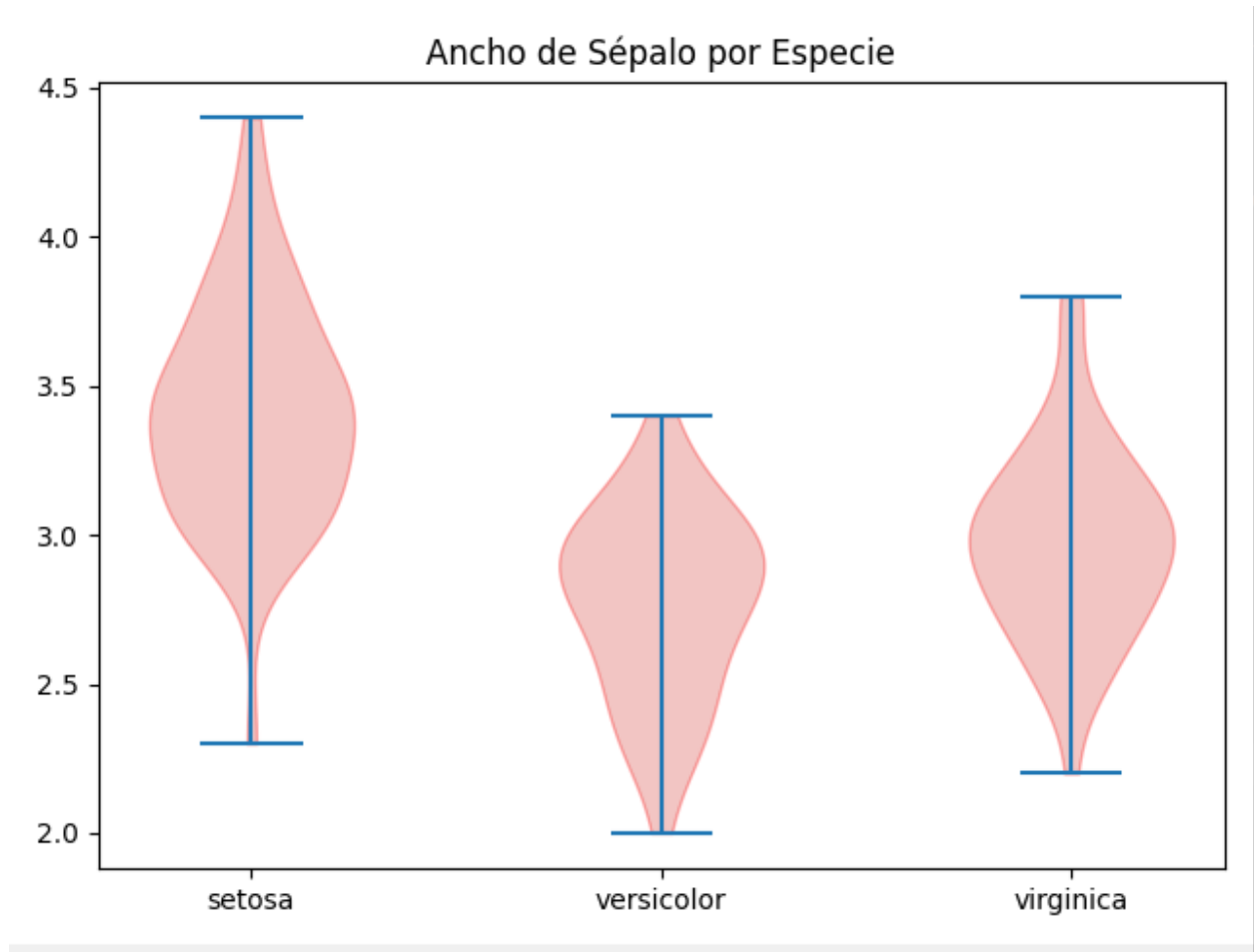


Gráfico-b resultado del problema 9.

Explicación de la gráfica-b: podemos ver el grafico de violín del ancho de los sépalos de cada uno de las especies. La línea de azul oscuro muestra todo lo que abarca desde el mínimo hasta el máximo valor que alcanzan los sépalos de cada especie, y en rosa podemos observar que en las partes más gruesas significan que es un valor que se repite más que otros, podemos deducir que donde más grueso será la moda .