Creado por:

Isabel Maniega

Bokeh

1) Representación Heatmap

Un mapa de calor (heat map, en inglés) es una técnica de visualización de datos que mide la magnitud de un fenómeno en colores en dos dimensiones. La variación del color puede ser por tono o intensidad, haciendo obvia la lectura del fenómeno sobre el espacio que se trata. Existen dos categorías fundamentales de mapas de calor: el mapa de calor de análisis de grupos y el mapa de calor espacial. En un mapa de calor de conglomerados, las magnitudes se disponen en una matriz de tamaño de celda fijo, cuyas filas y columnas son fenómenos y categorías discretos, además la clasificación de filas y columnas es intencional y algo arbitraria, con el objetivo de sugerir los conglomerados o representarlos como descubrimientos a través del análisis estadístico. El tamaño de la celda es arbitrario pero lo suficientemente grande para ser claramente visible. Por el contrario, la posición de una magnitud en un mapa de calor espacial está forzada por la ubicación de la magnitud en ese espacio, y no existe la noción de celdas; se considera que el fenómeno varía continuamente.

https://seaborn.pydata.org/generated/seaborn.heatmap.html
(https://seaborn.pydata.org/generated/seaborn.heatmap.html)

2) Aprender de forma autodidata a hacer las gráficas para el dataset Iris con "Bokeh"

In [1]:

```
# pip install bokeh
```

In [2]:

```
# pip install jupyter_bokeh
```

In [3]:

```
from bokeh.plotting import figure, show, output_file
from bokeh.sampledata.iris import flowers
from bokeh.io import output_notebook
from bokeh.models import Legend, LegendItem
```

In [4]:

```
print(flowers)
```

```
     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
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

Gráfico de Pétalo con Bokeh

In [5]:

```
# file to save the model
# output_file("1_Test_Bokeh.html")

colormap = {'setosa': 'red', 'versicolor': 'green', 'virginica': 'blue'}
colors = [colormap[x] for x in flowers['species']]
```

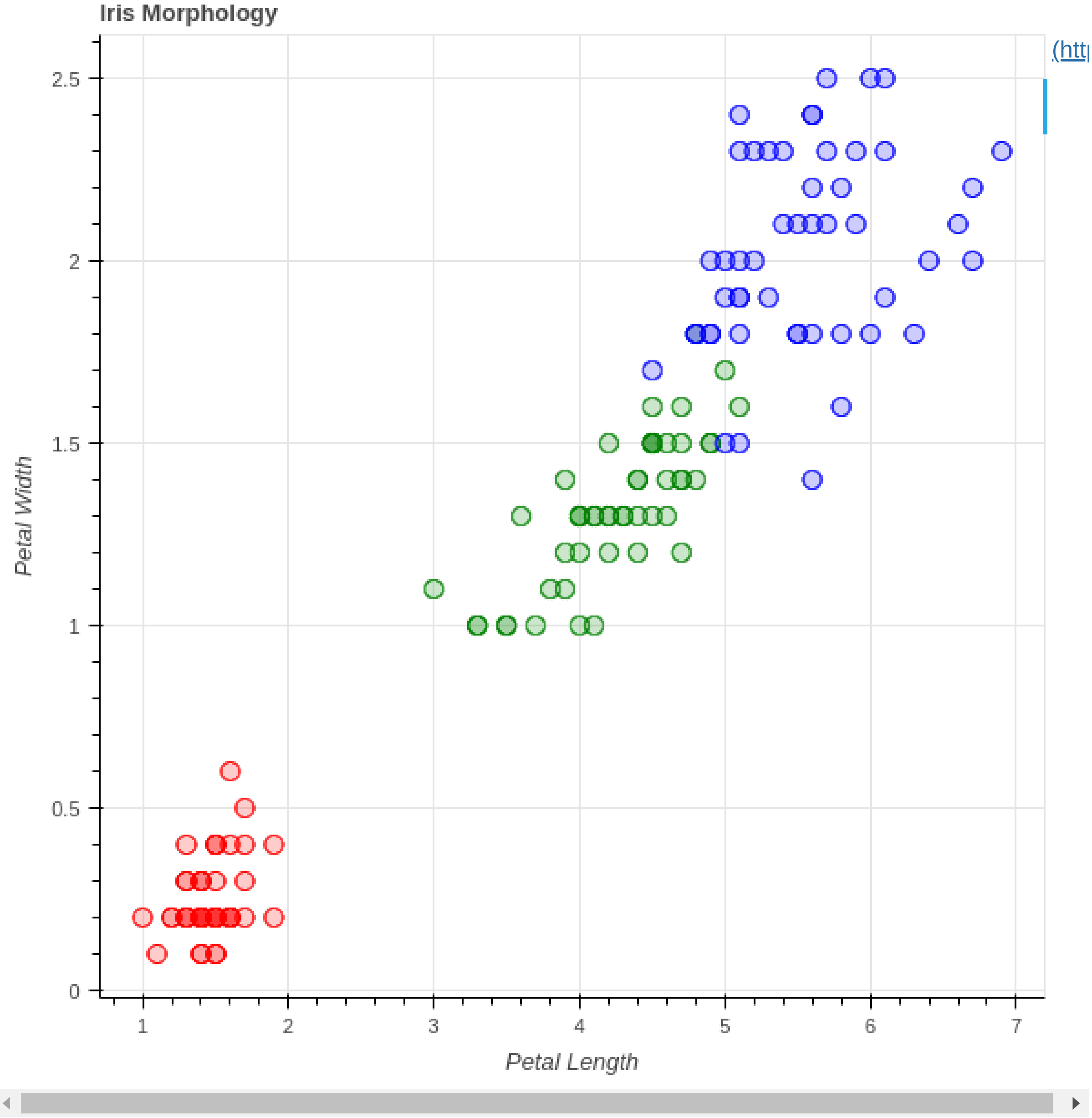
In [6]:

```
output_notebook()
```

(https://bokeh.org) BokehJS 2.4.3 successfully loaded.

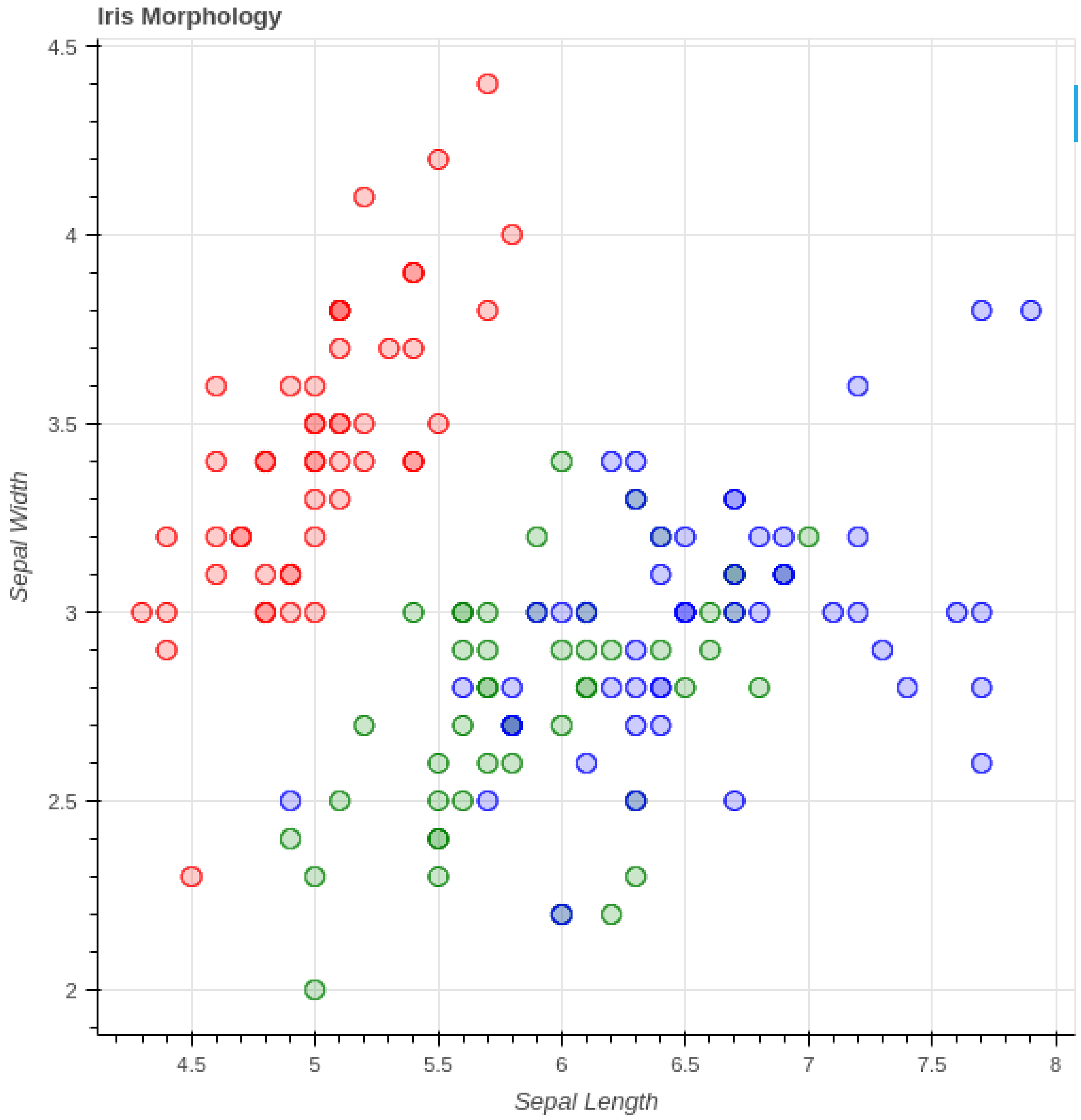
In [13]:

```
p = figure(title="Iris Morphology")
p.xaxis.axis_label = 'Petal Length'
p.yaxis.axis_label = 'Petal Width'

p.scatter(flowers["petal_length"], flowers["petal_width"],
          color=colors, fill_alpha=0.2, size=10)

show(p)
```



In [14]:

```
# file to save the model
# output_file("2_Test_Bokeh.html")

colormap = {'setosa': 'red', 'versicolor': 'green', 'virginica': 'blue'}
colors = [colormap[x] for x in flowers['species']]

p = figure(title="Iris Morphology")
p.xaxis.axis_label = 'Sepal Length'
p.yaxis.axis_label = 'Sepal Width'

p.scatter(flowers["sepal_length"], flowers["sepal_width"],
          color=colors, fill_alpha=0.2, size=10)

show(p)
```

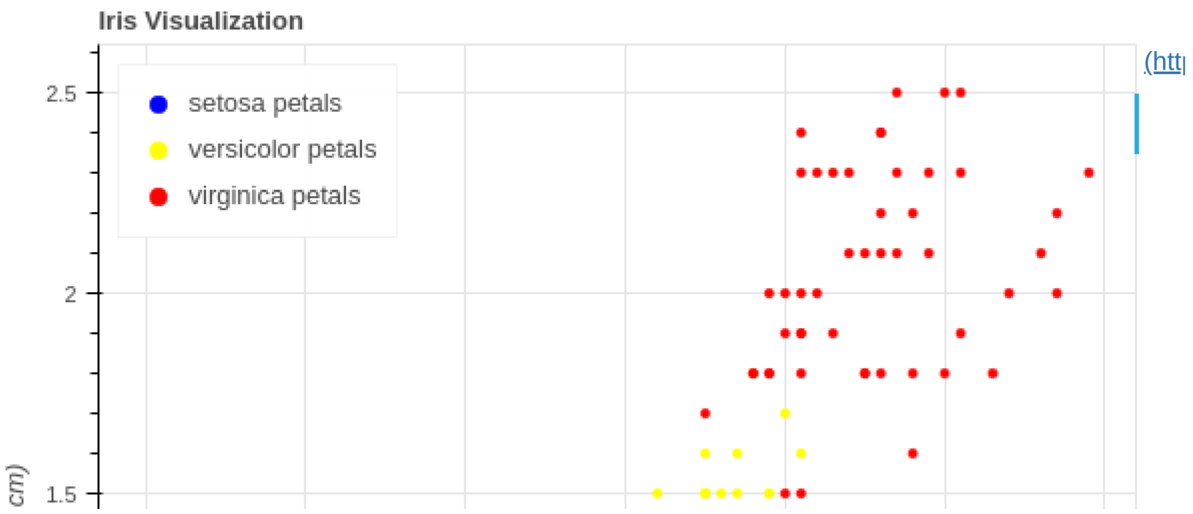


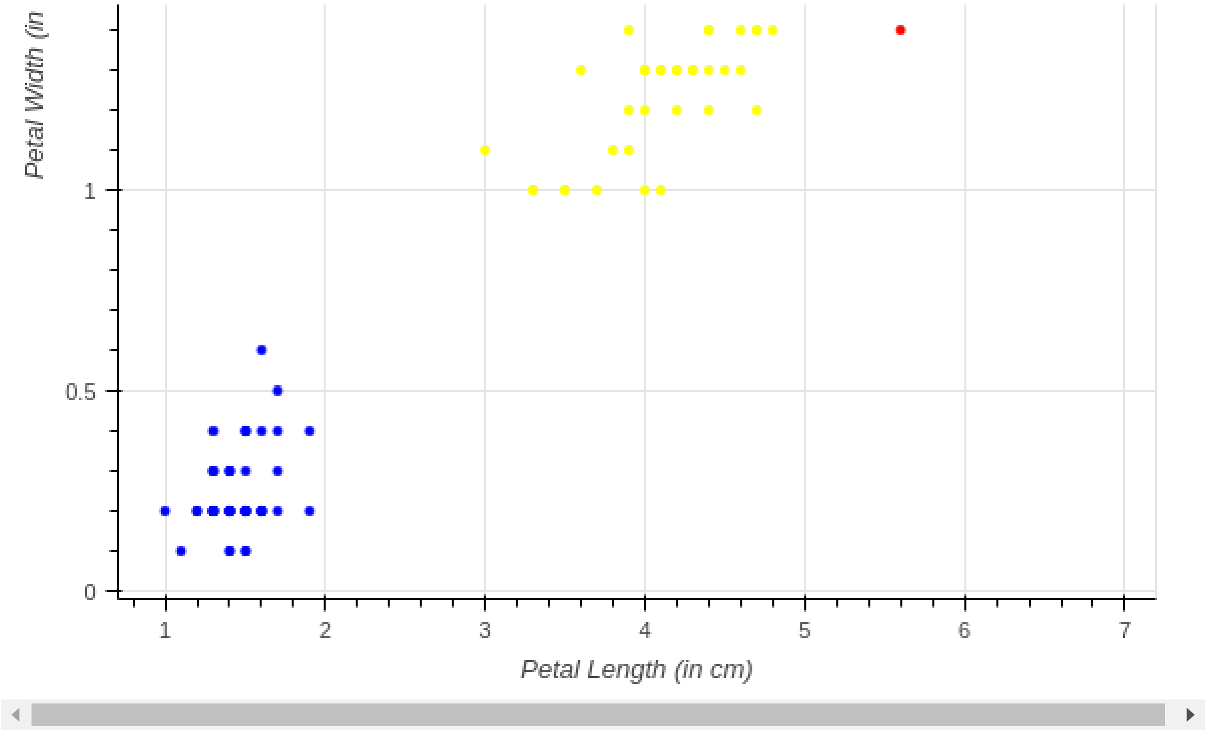
(htt

Otra forma de representar el Pétalo con Bokeh

In [9]:

```
# file to save the model
# output_file("3_Test_Bokeh.html")

# instantiating the figure object
graph = figure(title = "Iris Visualization")

# labeling the x-axis and the y-axis
graph.xaxis.axis_label = "Petal Length (in cm)"
graph.yaxis.axis_label = "Petal Width (in cm)"

# plotting for setosa petals
x = flowers[flowers["species"] == "setosa"]["petal_length"]
y = flowers[flowers["species"] == "setosa"]["petal_width"]
color = "blue"
legend_label = "setosa petals"
graph.circle(x, y,
             color = color,
             legend_label = legend_label)

# plotting for versicolor petals
x = flowers[flowers["species"] == "versicolor"]["petal_length"]
y = flowers[flowers["species"] == "versicolor"]["petal_width"]
color = "yellow"
legend_label = "versicolor petals"
graph.circle(x, y,
             color = color,
             legend_label = legend_label)

# plotting for virginica petals
x = flowers[flowers["species"] == "virginica"]["petal_length"]
y = flowers[flowers["species"] == "virginica"]["petal_width"]
color = "red"
legend_label = "virginica petals"
graph.circle(x, y,
             color = color,
             legend_label = legend_label)

# relocating the legend table to
# avoid abstruction of the graph
graph.legend.location = "top_left"

# displaying the model
show(graph)
```

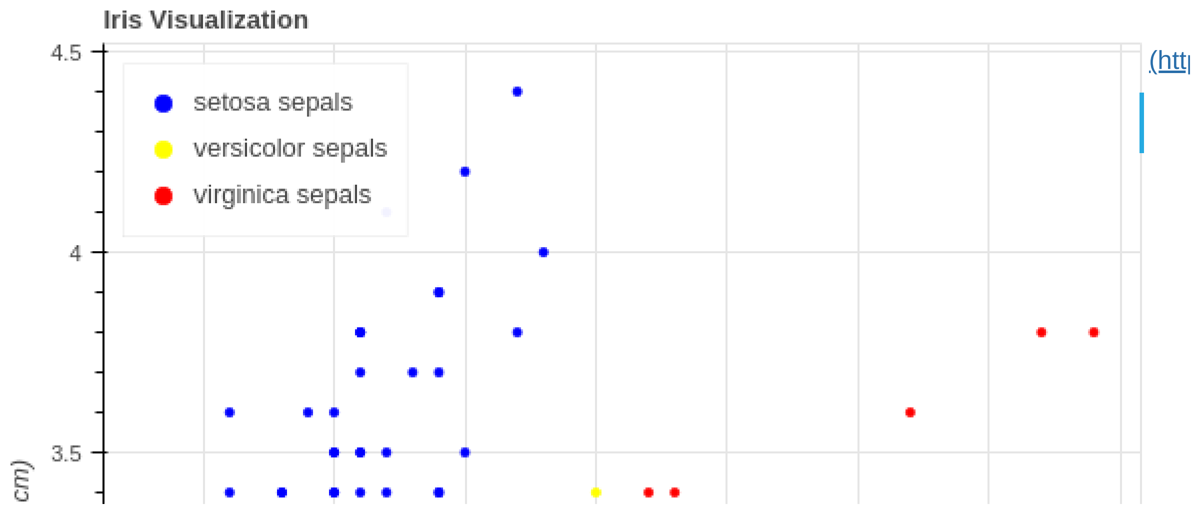


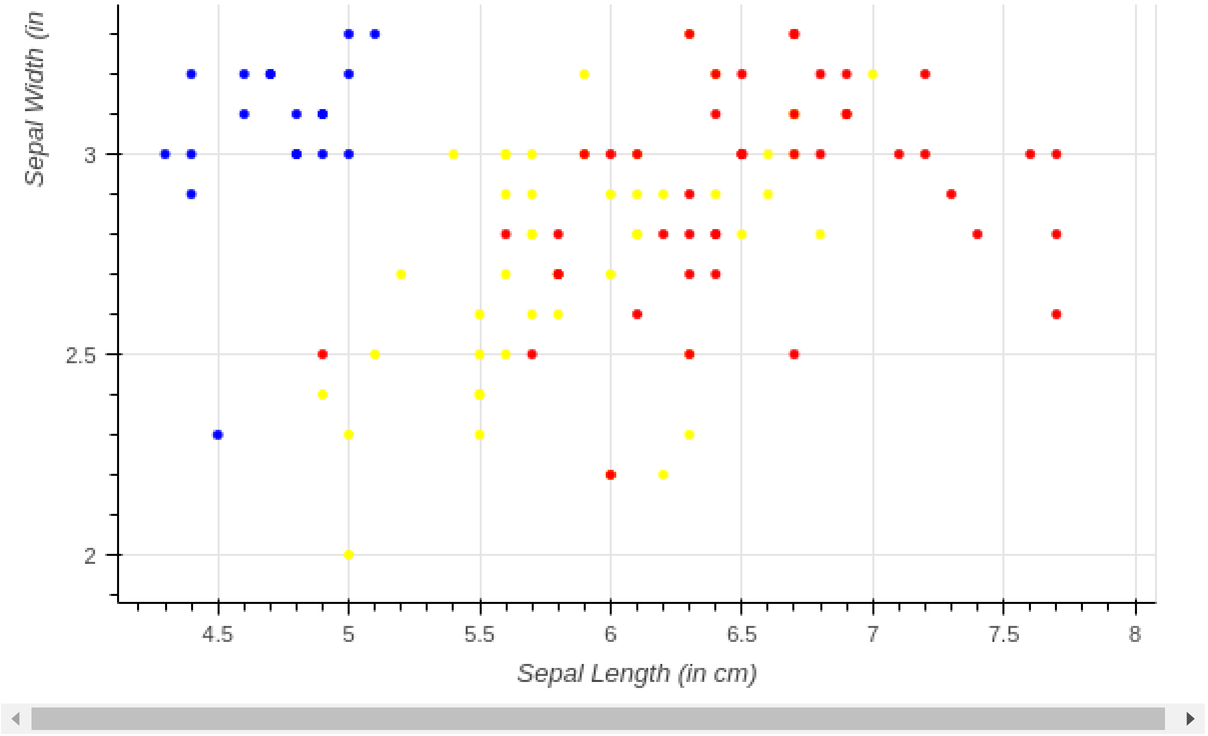


Sépalo ...

In [10]:

```
# file to save the model
# output_file("4_Test_Bokeh.html")

# instantiating the figure object
graph = figure(title = "Iris Visualization")

# labeling the x-axis and the y-axis
graph.xaxis.axis_label = "Sepal Length (in cm)"
graph.yaxis.axis_label = "Sepal Width (in cm)"

# plotting for setosa petals
x = flowers[flowers["species"] == "setosa"]["sepal_length"]
y = flowers[flowers["species"] == "setosa"]["sepal_width"]
color = "blue"
legend_label = "setosa sepals"
graph.circle(x, y,
             color = color,
             legend_label = legend_label)

# plotting for versicolor petals
x = flowers[flowers["species"] == "versicolor"]["sepal_length"]
y = flowers[flowers["species"] == "versicolor"]["sepal_width"]
color = "yellow"
legend_label = "versicolor sepals"
graph.circle(x, y,
             color = color,
             legend_label = legend_label)

# plotting for virginica petals
x = flowers[flowers["species"] == "virginica"]["sepal_length"]
y = flowers[flowers["species"] == "virginica"]["sepal_width"]
color = "red"
legend_label = "virginica sepals"
graph.circle(x, y,
             color = color,
             legend_label = legend_label)

# relocating the legend table to
# avoid abstruction of the graph
graph.legend.location = "top_left"

# displaying the model
show(graph)
```



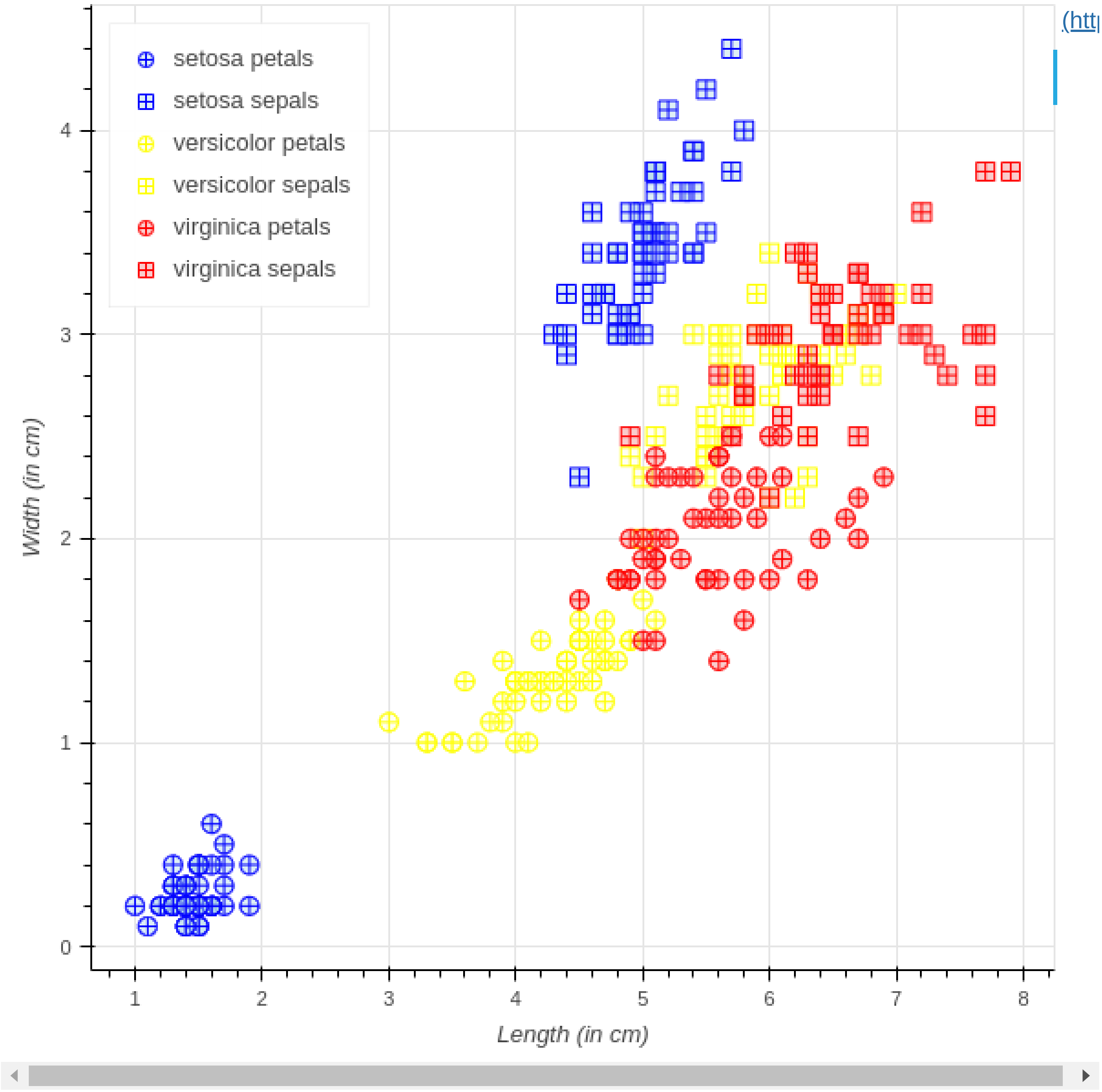


Otra más...

In [11]:

```
# file to save the model
# output_file("5_Test_Bokeh.html")

# instantiating the figure object
graph = figure(title = "Iris Visualization")

# labeling the x-axis and the y-axis
graph.xaxis.axis_label = "Length (in cm)"
graph.yaxis.axis_label = "Width (in cm)"

# plotting for setosa petals
x = flowers[flowers["species"] == "setosa"]["petal_length"]
y = flowers[flowers["species"] == "setosa"]["petal_width"]
marker = "circle_cross"
line_color = "blue"
fill_color = "lightblue"
fill_alpha = 0.4
size = 10
legend_label = "setosa petals"
graph.scatter(x, y,
              marker = marker,
              line_color = line_color,
              fill_color = fill_color,
              fill_alpha = fill_alpha,
              size = size,
              legend_label = legend_label)

# plotting for setosa sepals
x = flowers[flowers["species"] == "setosa"]["sepal_length"]
y = flowers[flowers["species"] == "setosa"]["sepal_width"]
marker = "square_cross"
line_color = "blue"
fill_color = "lightblue"
fill_alpha = 0.4
size = 10
legend_label = "setosa sepals"
graph.scatter(x, y,
              marker = marker,
              line_color = line_color,
              fill_color = fill_color,
              fill_alpha = fill_alpha,
              size = size,
              legend_label = legend_label)

# plotting for versicolor petals
x = flowers[flowers["species"] == "versicolor"]["petal_length"]
y = flowers[flowers["species"] == "versicolor"]["petal_width"]
marker = "circle_cross"
line_color = "yellow"
fill_color = "lightyellow"
fill_alpha = 0.4
size = 10
legend_label = "versicolor petals"
graph.scatter(x, y,
              marker = marker,
              line_color = line_color,
              fill_color = fill_color,
              fill_alpha = fill_alpha,
              size = size,
              legend_label = legend_label)

# plotting for versicolor sepals
x = flowers[flowers["species"] == "versicolor"]["sepal_length"]
y = flowers[flowers["species"] == "versicolor"]["sepal_width"]
marker = "square_cross"
line_color = "yellow"
fill_color = "lightyellow"
fill_alpha = 0.4
size = 10
legend_label = "versicolor sepals"
graph.scatter(x, y,
              marker = marker,
              line_color = line_color,
              fill_color = fill_color,
              fill_alpha = fill_alpha,
              size = size,
              legend_label = legend_label)

# plotting for virginica petals
x = flowers[flowers["species"] == "virginica"]["petal_length"]
y = flowers[flowers["species"] == "virginica"]["petal_width"]
marker = "circle_cross"
line_color = "red"
fill_color = "lightcoral"
fill_alpha = 0.4
size = 10
legend_label = "virginica petals"
graph.scatter(x, y,
              marker = marker,
              line_color = line_color,
              fill_color = fill_color,
              fill_alpha = fill_alpha,
              size = size,
              legend_label = legend_label)

# plotting for virginica sepals
x = flowers[flowers["species"] == "virginica"]["sepal_length"]
y = flowers[flowers["species"] == "virginica"]["sepal_width"]
marker = "square_cross"
line_color = "red"
fill_color = "lightcoral"
fill_alpha = 0.4
size = 10
legend_label = "virginica sepals"
graph.scatter(x, y,
              marker = marker,
              line_color = line_color,
              fill_color = fill_color,
              fill_alpha = fill_alpha,
              size = size,
              legend_label = legend_label)

# relocating the legend table to
# avoid abstruction of the graph
graph.legend.location = "top_left"

# displaying the model
show(graph)
```

Iris Visualization



Creado por:

Isabel Maniega