

Reporte de Matplotlib

Eduardo Peñuñuri Bolado

Grupo 1

10 de febrero de 2019

Reporte

En este caso utilizamos Matplotlib, lo que nos permitió realizar gráficas distintas para poder organizar los datos de la actividad anterior, obteniendo así cinco gráficas de tres tipos: dos de barras, una lineal y dos de caja.

Cabe aclarar que los datos que se obtuvieron no fueron del todo suficientes, abarcando un periodo de tres meses, por lo que se sustituyó las gráficas anuales por unas con todos los datos y las mensuales con el primer mes.

Lo primero fue hacer una copia de la actividad tres para poder trabajar con ella, para después comenzar con la primera gráfica, que al igual que la segunda, pedía la precipitación según el tiempo, y para la cual se utilizó un ejemplo de internet como base y después se modificó, quedando:

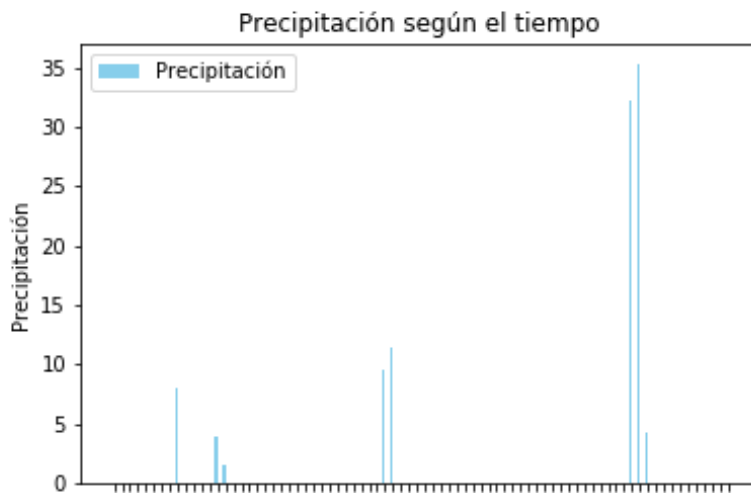
```
import numpy as np
import matplotlib.pyplot as plt

ind = np.arange(len(df.PRECIP))
width = 0.35 #El ancho de las barras

fig, ax = plt.subplots()
Barras = ax.bar(ind - width/2, df.PRECIP, width, color="SkyBlue", label="Precipitación")

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel("Precipitación")
ax.set_title("Precipitación según el tiempo")
ax.set_xticks(ind)
ax.set_xticklabels((" "))
ax.legend()

fig.savefig("Imagen1.png")
plt.show()
```



En la segunda se trataba de una variación de la primera, pero tomando solo ciertos datos, así que se creó un arreglo auxiliar con los primeros 30 datos a ocupar (ya que comenzaban desde el 1 de octubre, y así abarcamos aproximadamente un mes), para hacer lo mismo que con la anterior pero con datos algo diferentes, quedando de la siguiente manera:

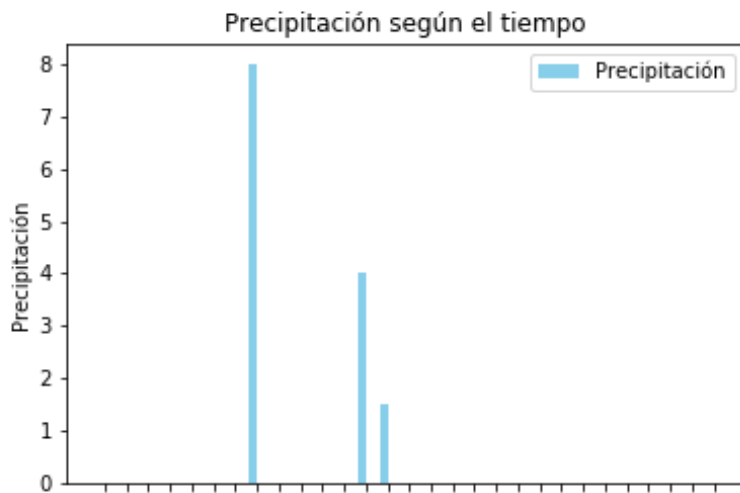
```
import numpy as np
import matplotlib.pyplot as plt

aux = (df.PRECIP[0], df.PRECIP[1], df.PRECIP[3], df.PRECIP[4], df.PRECIP[5], df.PRECIP[6], df.PRECIP[7],
        df.PRECIP[9], df.PRECIP[10], df.PRECIP[11], df.PRECIP[12], df.PRECIP[13], df.PRECIP[14],
        df.PRECIP[16], df.PRECIP[17], df.PRECIP[18], df.PRECIP[19], df.PRECIP[20], df.PRECIP[21],
        df.PRECIP[23], df.PRECIP[24], df.PRECIP[25], df.PRECIP[26], df.PRECIP[27], df.PRECIP[28], df.PRECIP[29])
ind = np.arange(len(aux))
width = 0.35 #El ancho de las barras

fig, ax = plt.subplots()
Barras = ax.bar(ind - width/2, aux, width, color="SkyBlue", label="Precipitación")

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel("Precipitación")
ax.set_title("Precipitación según el tiempo")
ax.set_xticks(ind)
ax.set_xticklabels((" "))
ax.legend()

fig.savefig("Imagen2.png")
plt.show()
```



Seguimos con la gráfica lineal (de evolución), pero al modificarla hicimos que hubieran dos líneas, ya que pedía las temperaturas máxima y mínima respecto al tiempo, para poder compararlas entre sí, y al final nos quedó:

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

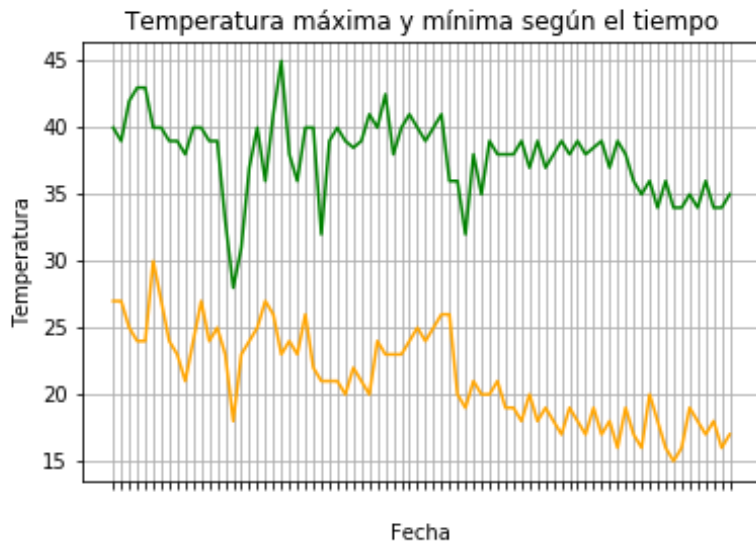
# Data for plotting
t = df.FECHA
s = df.TMAX
j = df.TMIN

fig, ax = plt.subplots()
line1 = ax.plot(t, s, color="Green", label="Temperatura máxima")
line2 = ax.plot(t, j, color="Orange", label="Temperatura mínima")

ax.set(xlabel='Fecha', ylabel='Temperatura',
       title='Temperatura máxima y mínima según el tiempo')
ax.grid()

ax.set_xticklabels((" "))

fig.savefig("Imagen3.png")
plt.show()
```



La penúltima, que era un gráfico de cajas de la temperatura máxima y la mínima, siguió el mismo proceso de modificar ejemplos de internet, pero requirió más trabajo ya que se me hizo más difícil de entender y modificar, principalmente porque daba dos gráficos en una imagen, así que una vez hice que fuera solo uno, solo requirió modificar los datos para obtener el resultado:

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

all_data = df.TMIN, df.TMAX
labels = ["Mínima", "Máxima"]

fig, axe = plt.subplots(nrows=1, ncols=1, figsize=(5, 5))

# rectangular box plot
bplot1 = axe.boxplot(all_data,
                     vert=True, # vertical box alignment
                     patch_artist=True, # fill with color
                     labels=labels) # will be used to label x-ticks
axe.set_title("Tempearaturas máxima y mínima")

# fill with colors
colors = ['pink', 'lightblue', 'lightgreen']
for bplot in (bplot1, bplot2):
    for patch, color in zip(bplot['boxes'], colors):
        patch.set_facecolor(color)

# adding horizontal grid lines
```

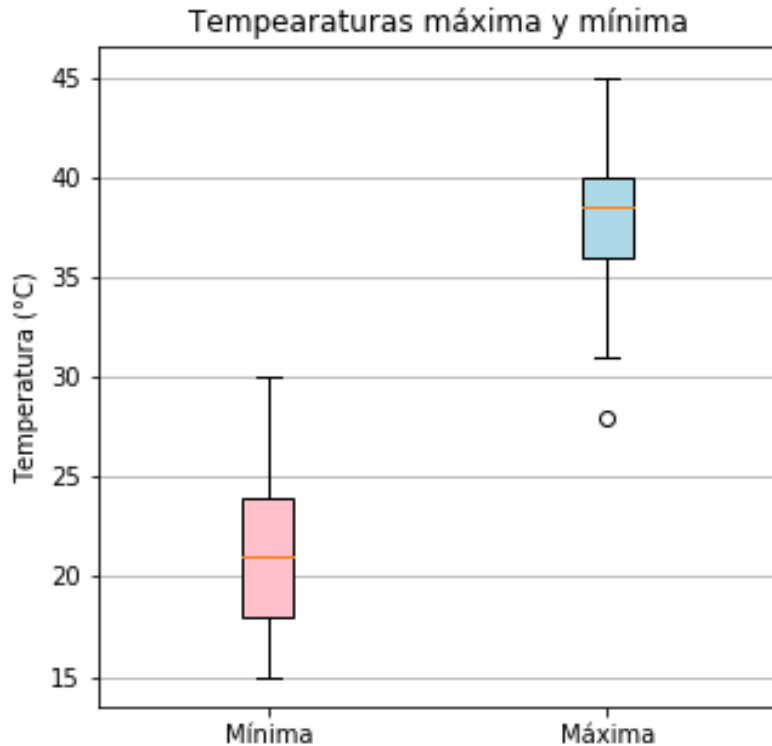
```

    axe.yaxis.grid(True)
    axe.set_ylabel("Temperatura (C)")

plt.show()

fig.savefig("Imagen4.png")
plt.show()

```



La última gráfica tuvo el mismo proceso que la segunda, ya que solo se requirió hacer arreglos auxiliares (uno para cada temperatura) y darlos al código en lugar de los datos completos, quedando:

```

import matplotlib.pyplot as plt
import numpy as np

aux1 = (df.TMIN[0], df.TMIN[1], df.TMIN[3], df.TMIN[4], df.TMIN[5], df.TMIN[6], df.TMIN[7], df.TMIN[8],
        df.TMIN[9], df.TMIN[10], df.TMIN[11], df.TMIN[12], df.TMIN[13], df.TMIN[14], df.TMIN[15], df.TMIN[16],
        df.TMIN[17], df.TMIN[18], df.TMIN[19], df.TMIN[20], df.TMIN[21], df.TMIN[22], df.TMIN[23], df.TMIN[24],
        df.TMIN[25], df.TMIN[26], df.TMIN[27], df.TMIN[28], df.TMIN[29])

aux2 = (df.TMAX[0], df.TMAX[1], df.TMAX[3], df.TMAX[4], df.TMAX[5], df.TMAX[6], df.TMAX[7], df.TMAX[8],
        df.TMAX[9], df.TMAX[10], df.TMAX[11], df.TMAX[12], df.TMAX[13], df.TMAX[14], df.TMAX[15], df.TMAX[16],
        df.TMAX[17], df.TMAX[18], df.TMAX[19], df.TMAX[20], df.TMAX[21], df.TMAX[22], df.TMAX[23], df.TMAX[24],
        df.TMAX[25], df.TMAX[26], df.TMAX[27], df.TMAX[28], df.TMAX[29])

```

```

df.TMAX[16],df.TMAX[17],df.TMAX[18],df.TMAX[19],df.TMAX[20],df.TMAX[21],df.TMAX[22],
df.TMAX[23],df.TMAX[24],df.TMAX[25],df.TMAX[26],df.TMAX[27],df.TMAX[28],df.TMAX[29],df.TMAX[30],df.TMAX[31],df.TMAX[32],df.TMAX[33],df.TMAX[34],df.TMAX[35],df.TMAX[36],df.TMAX[37],df.TMAX[38],df.TMAX[39],df.TMAX[40],df.TMAX[41],df.TMAX[42],df.TMAX[43],df.TMAX[44],df.TMAX[45],df.TMAX[46],df.TMAX[47],df.TMAX[48],df.TMAX[49],df.TMAX[50],df.TMAX[51],df.TMAX[52],df.TMAX[53],df.TMAX[54],df.TMAX[55],df.TMAX[56],df.TMAX[57],df.TMAX[58],df.TMAX[59],df.TMAX[60],df.TMAX[61],df.TMAX[62],df.TMAX[63],df.TMAX[64],df.TMAX[65],df.TMAX[66],df.TMAX[67],df.TMAX[68],df.TMAX[69],df.TMAX[70],df.TMAX[71],df.TMAX[72],df.TMAX[73],df.TMAX[74],df.TMAX[75],df.TMAX[76],df.TMAX[77],df.TMAX[78],df.TMAX[79],df.TMAX[80],df.TMAX[81],df.TMAX[82],df.TMAX[83],df.TMAX[84],df.TMAX[85],df.TMAX[86],df.TMAX[87],df.TMAX[88],df.TMAX[89],df.TMAX[90],df.TMAX[91],df.TMAX[92],df.TMAX[93],df.TMAX[94],df.TMAX[95],df.TMAX[96],df.TMAX[97],df.TMAX[98],df.TMAX[99]

all_data = aux1, aux2
labels = ["Mínima", "Máxima"]

fig, axe = plt.subplots(nrows=1, ncols=1, figsize=(5, 5))

# rectangular box plot
bplot1 = axe.boxplot(all_data,
                    vert=True, # vertical box alignment
                    patch_artist=True, # fill with color
                    labels=labels) # will be used to label x-ticks
axe.set_title("Tempearaturas máxima y mínima")

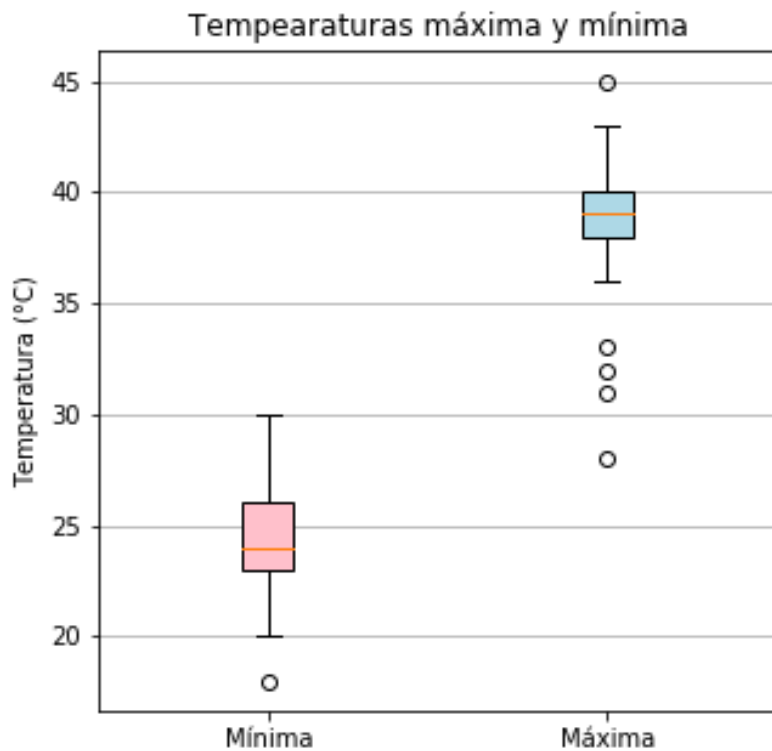
# fill with colors
colors = ['pink', 'lightblue', 'lightgreen']
for bplot in (bplot1, bplot2):
    for patch, color in zip(bplot['boxes'], colors):
        patch.set_facecolor(color)

# adding horizontal grid lines
axe.yaxis.grid(True)
axe.set_ylabel("Temperatura (C)")

plt.show()

fig.savefig("Imagen5.png")
plt.show()

```



Conclusión

La actividad no estuvo difícil como tal, el problema fue estar sin saber nada prácticamente y tener que averiguar todo por mi cuenta, pero fuera de eso igualmente logré entender el cómo gráficas los datos y trabajar con ellos, al menos de una forma básica y sabiendo lo necesario para las actividades y lo que se pudiera ofrecer, teniendo como quizás único problema a destacar que puedo tardar un poco, ya que no estoy acostumbrado y no se me hace tan sencillo como quisiera.

Referencias

- [1] Matplotlib (2018). *Gallery*. Recuperado en febrero de 2019 de: <https://matplotlib.org/gallery/index.html>
- [2] Karlijn Willems (2017). *Matplotlib Tutorial: Python Plotting*. Recuperado en febrero de 2019 de: <https://www.datacamp.com/community/tutorials/matplotlib-tutorial-python>
- [3] Matplotlib (2018). *Tutorials*. Recuperado en febrero de 2019 de: <https://matplotlib.org/tutorials/index.html>