### Trazar fechas

Podemos mejorar nuestro gráfico extrayendo las fechas de las lecturas diarias de temperatura alta, y utilizando estas fechas en el eje *x*:

**sitka\_highs.py**

from pathlib import Path  
import csv  
from datetime import datetime  
  
import matplotlib.pyplot as plt  
  
path = Path('weather\_data/sitka\_weather\_07-2021\_simple.csv')  
lines = path.read\_text().splitlines()  
  
reader = csv.reader(lines)  
header\_row = next(reader)  
  
# Extract dates and high temperatures.  
❶ dates, highs = [], []  
for row in reader:  
❷ current\_date = datetime.strptime(row[2], '%Y-%m-%d')  
 high = int(row[4])  
 dates.append(current\_date)  
 highs.append(high)  
  
# Plot the high temperatures.  
plt.style.use('seaborn')  
fig, ax = plt.subplots()  
❸ ax.plot(dates, highs, color='red')  
  
# Format plot.  
ax.set\_title("Daily High Temperatures, July 2021", fontsize=24)  
ax.set\_xlabel('', fontsize=16)  
❹ fig.autofmt\_xdate()  
ax.set\_ylabel("Temperature (F)", fontsize=16)  
ax.tick\_params(labelsize=16)  
  
plt.show()

Creamos dos listas vacías para almacenar las fechas y las altas temperaturas del archivo ❶. A continuación, convertimos los datos que contienen la información de la fecha (row[2]) en un objeto datetime ❷ y lo añadimos a dates. Pasamos las fechas y los valores de las altas temperaturas a plot() ❸. La llamada a fig.autofmt\_xdate() ❹ dibuja las etiquetas de las fechas en diagonal para evitar que se solapen. La [Figura 16-2](#figure16-2) muestra el gráfico mejorado.

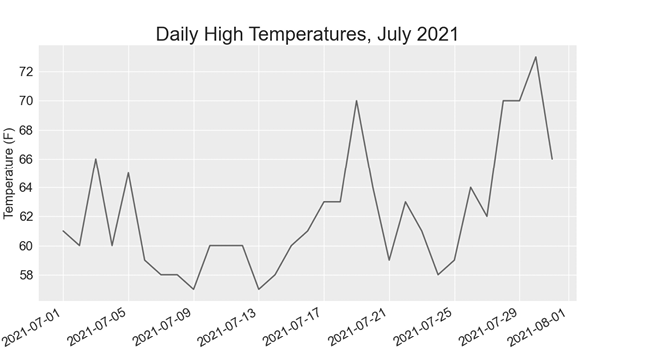


Figura 16-2: El gráfico es más significativo, ahora que tiene fechas

en el eje



*x*.

[anterior](c16_8.html)[Subtema 9 de 29: (Ver todo)](c16.html)[siguiente](c16_10.html)