### Calcular datos automáticamente

Escribir listas a mano puede resultar ineficaz, sobre todo cuando tenemos muchos puntos. En lugar de escribir cada valor, utilicemos un bucle que haga los cálculos por nosotros.

Así se vería esto con 1.000 puntos:

**scatter\_squares.py**

import matplotlib.pyplot as plt  
  
❶ x\_values = range(1, 1001)  
y\_values = [x\*\*2 for x in x\_values]  
  
plt.style.use('seaborn')  
fig, ax = plt.subplots()  
❷ ax.scatter(x\_values, y\_values, s=10)  
  
# Set chart title and label axes.  
--snip--  
  
# Set the range for each axis.  
❸ ax.axis([0, 1100, 0, 1\_100\_000])  
  
plt.show()

Empezamos con un rango de *x*-valores que contienen los números del 1 al 1.000 ❶. A continuación, una comprensión de listas genera los *y*-valores recorriendo en bucle los *x*-valores (for x in x\_values), elevando al cuadrado cada número (x\*\*2) y asignando los resultados a y\_values. Luego pasamos las listas de entrada y salida a scatter() ❷. Como se trata de un conjunto de datos grande, utilizamos un tamaño de punto más pequeño.

Antes de mostrar el gráfico, utilizamos el método axis() para especificar el rango de cada eje ❸. El método axis() requiere cuatro valores: los valores mínimo y máximo del eje *x* y del eje *y*. Aquí, ejecutamos el eje *x* de 0 a 1.100 y el eje *y* de 0 a 1.100.000. [La Figura 15-7](#figure15-7) muestra el resultado.

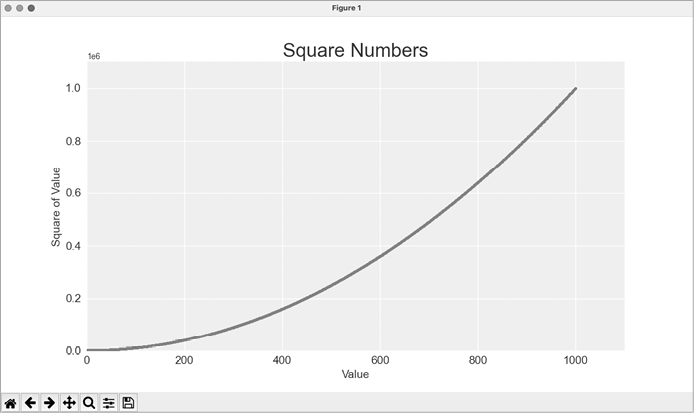


Figura 15-7: Python puede trazar 1.000 puntos tan fácilmente como traza 5 puntos.

[anterior](c15_8.html)[Subtema 9 de 35: (Ver todo)](c15.html)[siguiente](c15_10.html)