### Corregir el gráfico

Ahora que podemos leer mejor el gráfico, podemos ver que los datos no están trazados correctamente. Observa que al final del gráfico el cuadrado de 4,0 aparece como ¡25! Vamos a corregirlo.

Cuando das a plot() una única secuencia de números, asume que el primer punto de datos corresponde a un valor *x*-de 0, pero nuestro primer punto corresponde a un valor *x*-de 1. Podemos anular el comportamiento por defecto dando a plot() tanto los valores de entrada como los de salida utilizados para calcular los cuadrados:

**mpl\_squares.py**

import matplotlib.pyplot as plt  
  
input\_values = [1, 2, 3, 4, 5]  
squares = [1, 4, 9, 16, 25]  
  
fig, ax = plt.subplots()  
ax.plot(input\_values, squares, linewidth=3)  
  
# Set chart title and label axes.  
--snip--

Ahora plot() no tiene que hacer ninguna suposición sobre cómo se generaron los números de salida. El gráfico resultante, que se muestra en [la Figura 15-3](#figure15-3), es correcto.

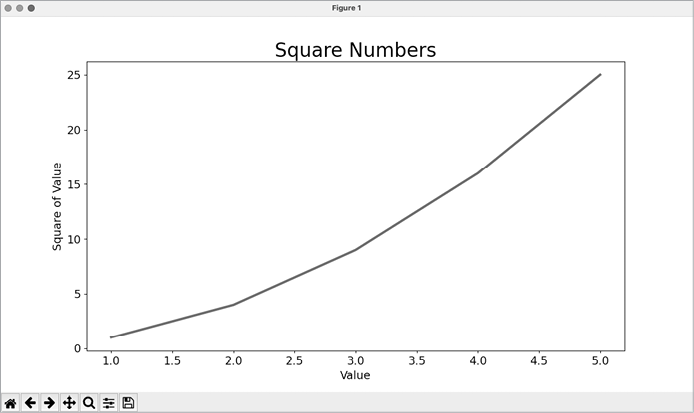


Figura 15-3: Ahora los datos se representan correctamente.

Puedes especificar una serie de argumentos al llamar a plot() y utilizar una serie de métodos para personalizar tus gráficos después de generarlos. Seguiremos explorando estos métodos de personalización a medida que trabajemos con conjuntos de datos más interesantes a lo largo de este capítulo.

[anterior](c15_4.html)[Subtema 5 de 35: (Ver todo)](c15.html)[siguiente](c15_6.html)