How do you perform linear interpolation in Python? Can you provide an example?

Authored by

stats writer

July 2, 2024

Linear interpolation is a method used to estimate values between two known data points. In Python, this can be achieved by using the "interp1d" function from the "scipy.interpolate" library. This function takes in the x and y values of the known data points and returns a linearly interpolated function that can be used to estimate values at any desired point within the given range. An example of performing linear interpolation in Python would be as follows:

```
import numpy as np
from scipy.interpolate import interp1d

x = # known x values
y = # known y values

f = interp1d(x, y) # creating the linearly interpolated function

print(f(5)) # estimating the value at x = 5
```

This would output a value of 10, which is the estimated value at x = 5 based on the given data points.

Perform Linear Interpolation in Python (With Example)

Linear interpolation is the process of estimating an unknown value of a function between two known values.

Given two known values (x1, y1) and (x2, y2), we can estimate the y-value for some point x by using the following formula:

y = y1 + (x-x1)(y2-y1)/(x2-x1)

We can use the following basic syntax to perform linear interpolation in Python:

```
import scipy.interpolate

y_interp = scipy.interpolate.interp1d(x, y)

#find y-value associated with x-value of 13
print(y_interp(13))
```

The following example shows how to use this syntax in practice.

Example: Linear Interpolation in Python

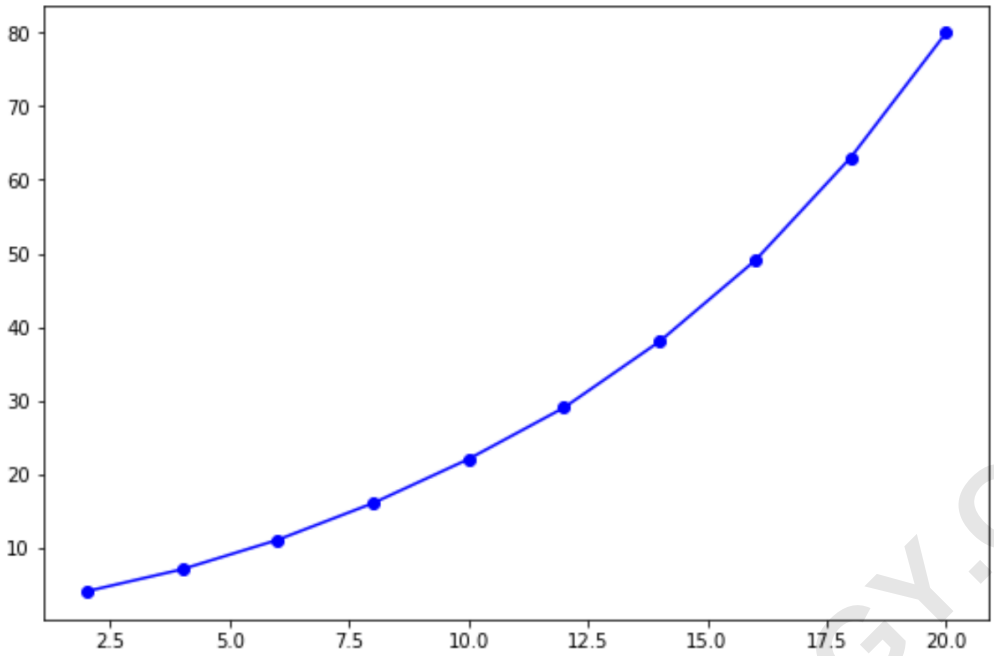
Suppose we have the following two lists of values in Python:

```
x =
y =
```

We can create a quick plot x vs. y:

```
import matplotlib.pyplotas plt

#create plot of x vs. y
plt.plot(x, y, '-ob')
```



Now suppose that we'd like to find the y-value associated with a new x-value of 13.

We can use the following code to do so:

```
import scipy.interpolate
y_interp = scipy.interpolate.interp1d(x, y)

#find y-value associated with x-value of 13
print(y_interp(13))
```

33.5

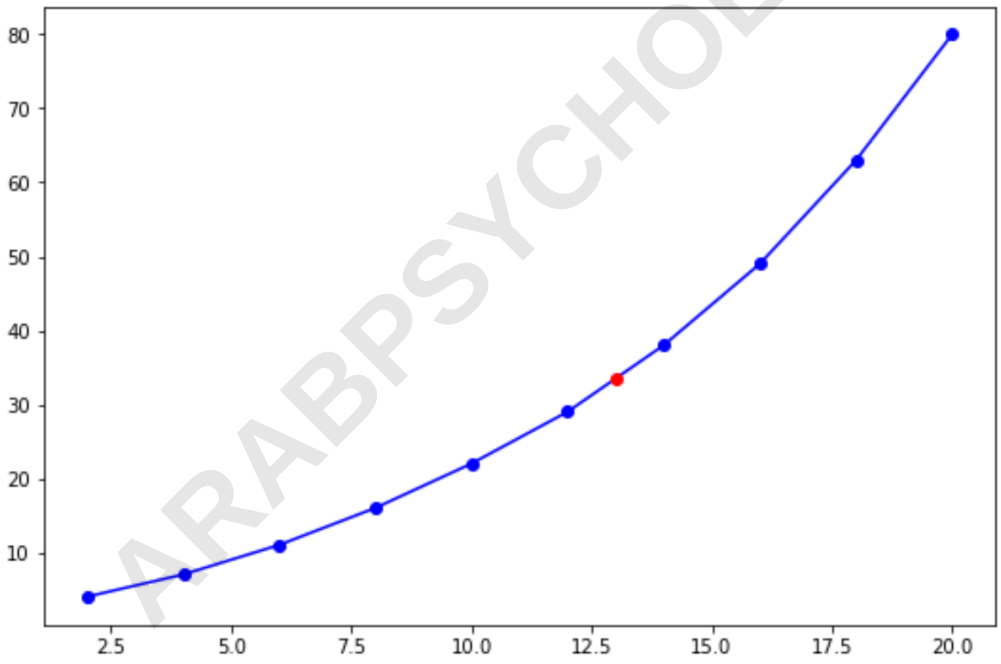
The estimated y-value turns out to be 33.5.

If we add the point (13, 33.5) to our plot, it appears to match the function quite well:

```
import matplotlib.pyplotas plt

#create plot of x vs. y
plt.plot(x, y, '-ob')

#add estimated y-value to plot
plt.plot(13, 33.5, 'ro')
```



We can use this exact formula to perform linear interpolation for any new x-value.

Additional Resources

The following tutorials explain how to fix other common errors in Python: