

4.0-Data Interpolation

December 31, 2023

0.1 Data Interpolation

Data interpolation is the process of estimating unknown values within a dataset based on the known values. In Python, there are various libraries available that can be used for data interpolation, such as NumPy, SciPy, and Pandas. Here is an example of how to perform data interpolation using the NumPy library:

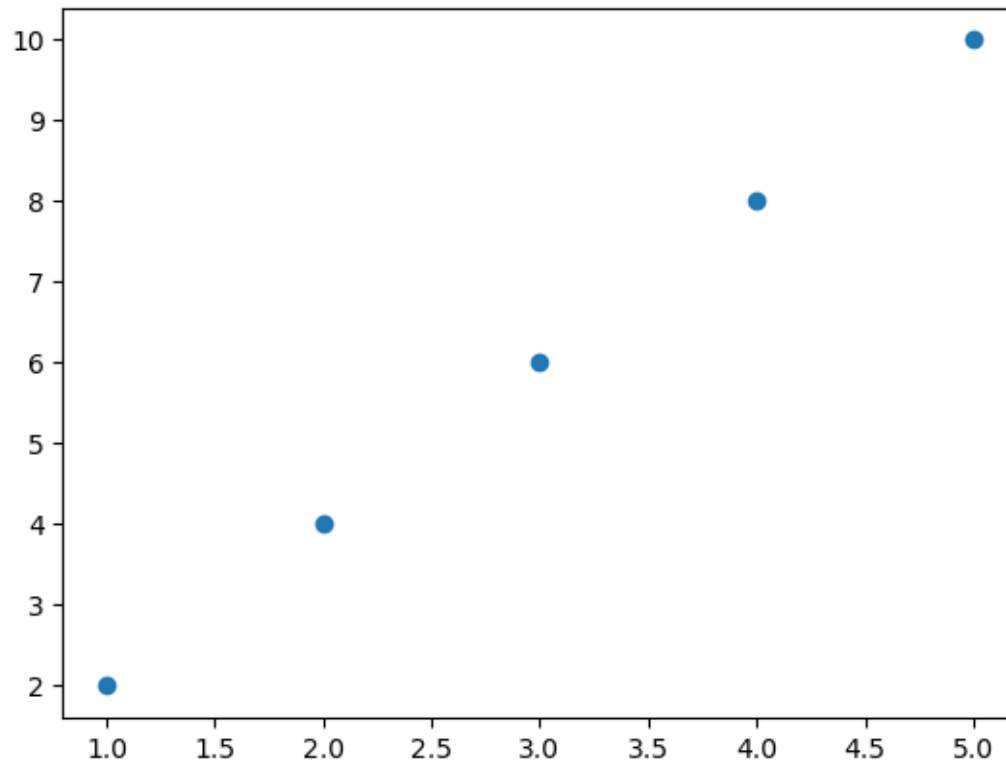
1. Linear Interpolation
2. Cubic Interpolation With Scipy
3. Polynomial Interpolation

0.2 1. Linear Interpolation

```
[ ]: import numpy as np
      x=np.array([1,2,3,4,5])
      y=np.array([2,4,6,8,10])
```

```
[ ]: import matplotlib.pyplot as plt
      plt.scatter(x,y)
```

```
[ ]: <matplotlib.collections.PathCollection at 0x7f3224dc63b0>
```



```
[ ]: ## interpolate the data using linear interpolation  
x_new=np.linspace(1,5,10) ## create new x values  
y_interp=np.interp(x_new,x,y)
```

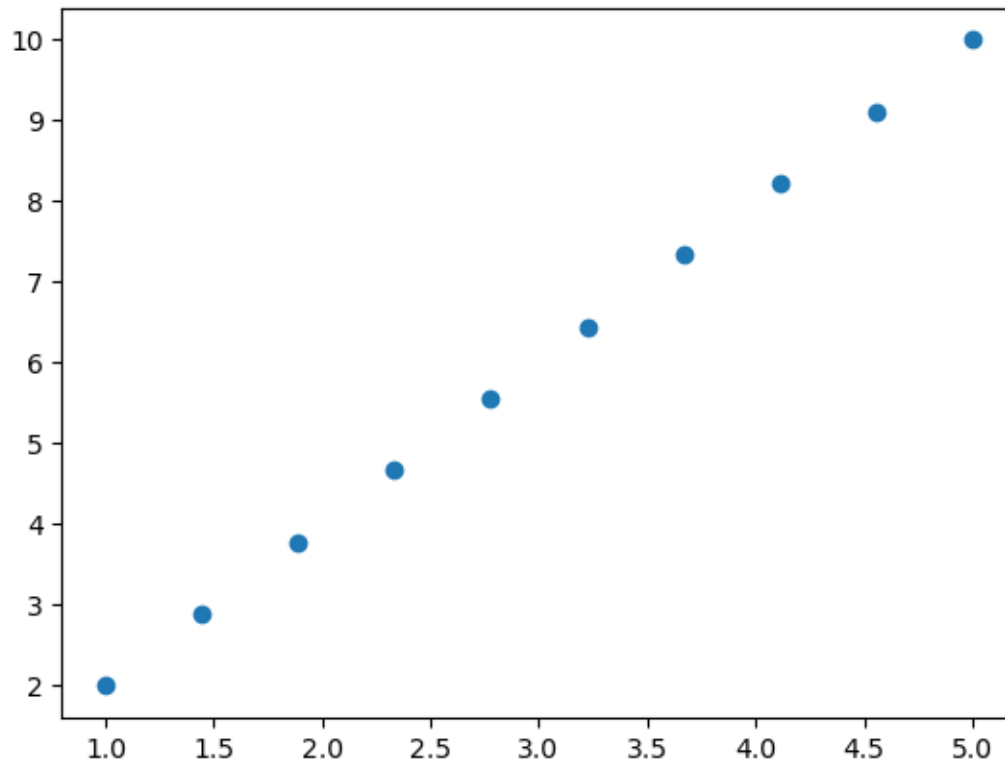
```
[ ]:
```

```
[ ]: y_interp
```

```
[ ]: array([ 2.          ,  2.88888889,  3.77777778,  4.66666667,  5.55555556,  
          6.44444444,  7.33333333,  8.22222222,  9.11111111, 10.          ])
```

```
[ ]: plt.scatter(x_new,y_interp)
```

```
[ ]: <matplotlib.collections.PathCollection at 0x7f321ca85b40>
```



0.3 2. Cubic Interpolation With Scipy

```
[ ]: import numpy as np
x=np.array([1,2,3,4,5])
y=np.array([1,8,27,64,125])
```

```
[ ]: from scipy.interpolate import interp1d
```

```
[ ]: ##create a cubic interpolation function
f=interp1d(x,y,kind='cubic')
```

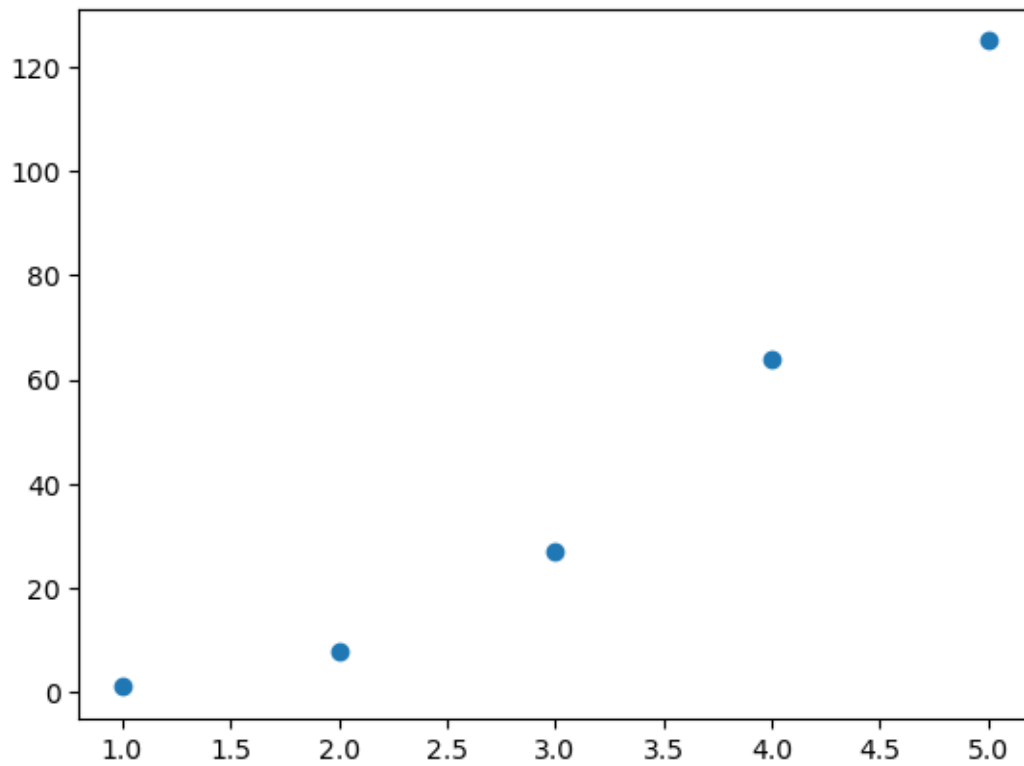
```
[ ]: # interpolate the data
x_new = np.linspace(1, 5, 10)
y_interp=f(x_new)
```

```
[ ]: y_interp
```

```
[ ]: array([ 1.          ,  3.01371742,  6.739369   , 12.7037037 ,
          21.43347051,  33.45541838,  49.2962963  , 69.48285322,
          94.54183813, 125.          ])
```

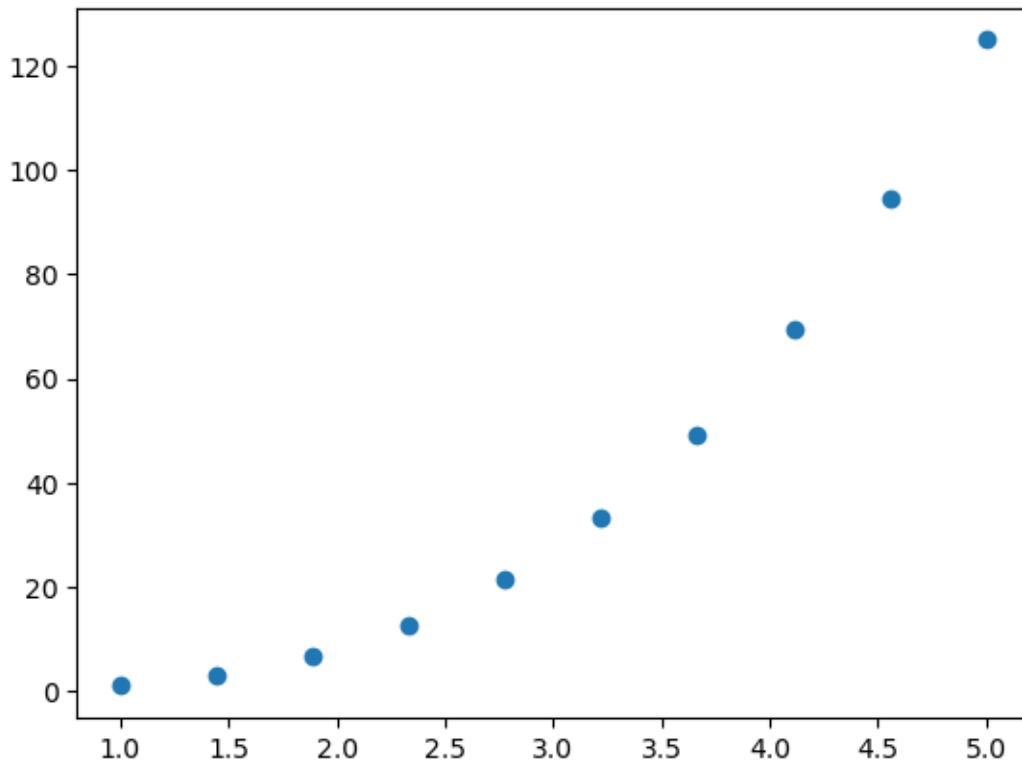
```
[ ]: plt.scatter(x,y)
```

```
[ ]: <matplotlib.collections.PathCollection at 0x7f321951a200>
```



```
[ ]: plt.scatter(x_new,y_interp)
```

```
[ ]: <matplotlib.collections.PathCollection at 0x7f32195e3640>
```



0.4 3. Polynomial Interpolation

```
[ ]: import numpy as np

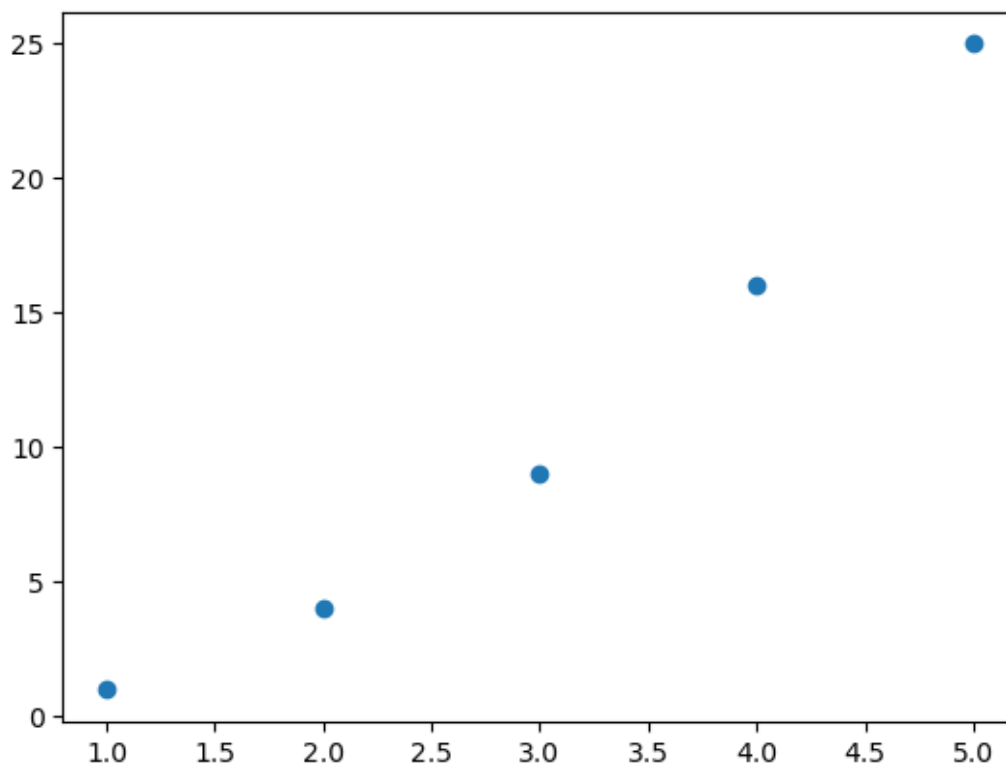
      # create some sample data
      x = np.array([1, 2, 3, 4, 5])
      y = np.array([1, 4, 9, 16, 25])

[ ]: # interpolate the data using polynomial interpolation
      p=np.polyfit(x,y,2)

[ ]: x_new = np.linspace(1, 5, 10) # create new x values
      y_interp = np.polyval(p, x_new) # interpolate y values

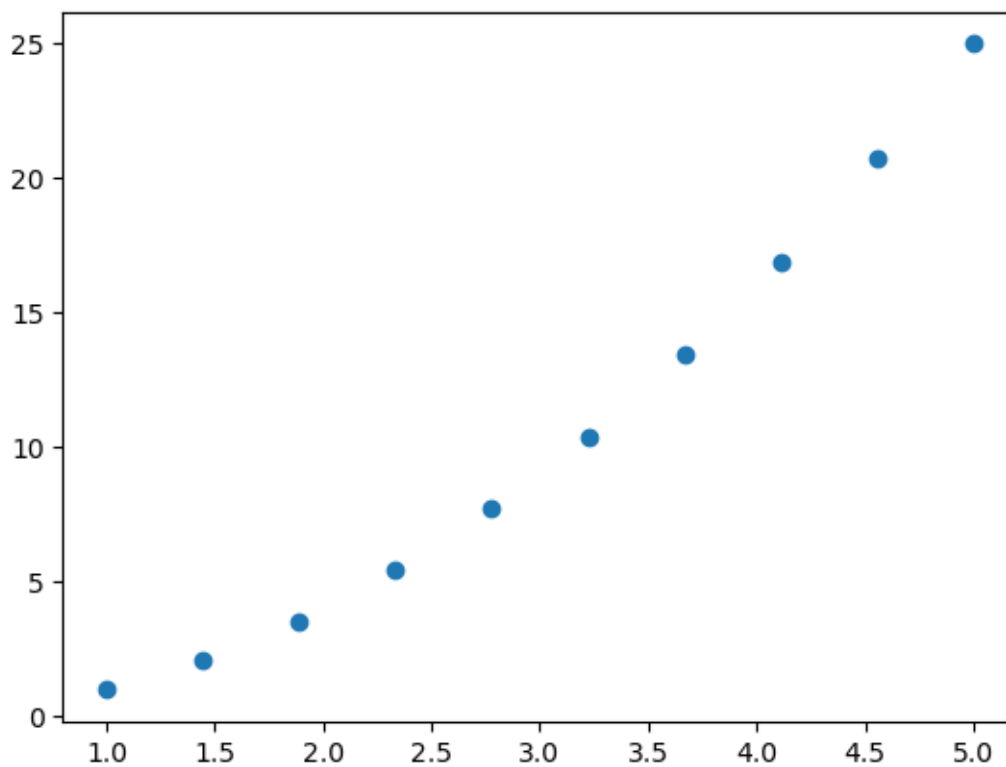
[ ]: plt.scatter(x,y)

[ ]: <matplotlib.collections.PathCollection at 0x7f32195b3370>
```



```
[ ]: plt.scatter(x_new,y_interp)
```

```
[ ]: <matplotlib.collections.PathCollection at 0x7f32191f4550>
```



[]:

[]:

[]: