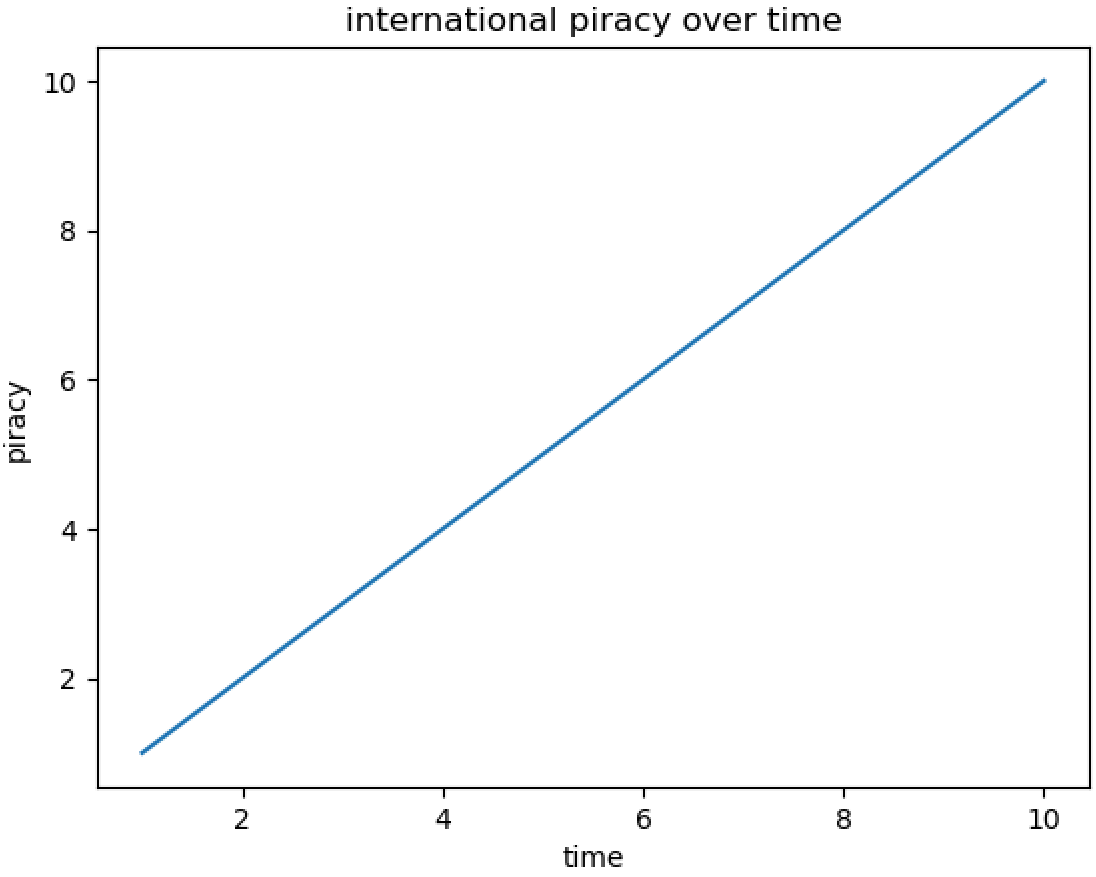
matplotlib homework

1. Make a plot of a straight line. Use `linspace()` to create the *x* values and the formula of a straight line, *y = a + bx, to create the* y* values (use an *a* and *b* of your choosing). You can pretend x and y are anything you like (x = time, y = international piracy or whatever).

In [7]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [10]:
```
x = np.linspace(1,10)
plt.plot(x,x)
plt.title('international piracy over time')
plt.xlabel('time')
plt.ylabel('piracy')
```

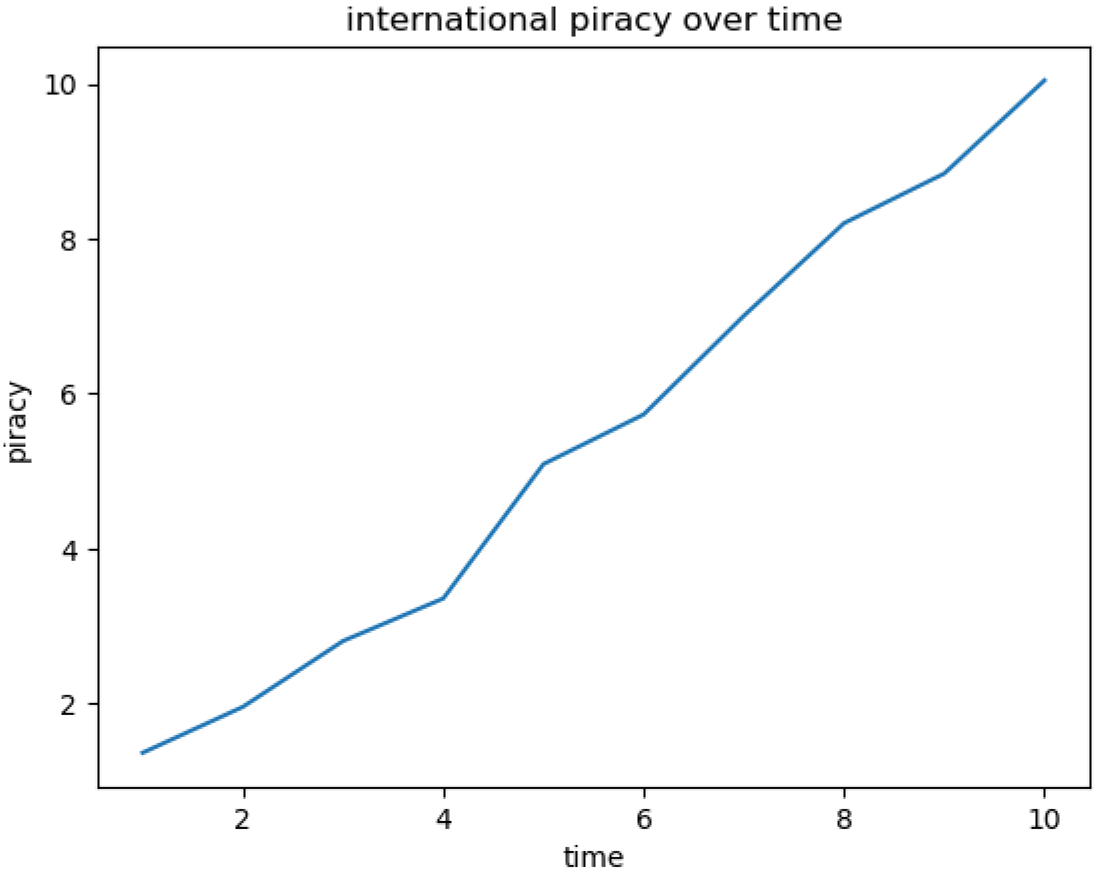
Out[10]:
```
Text(0, 0.5, 'piracy')
```



2. Make some data that are straight line values from the same straight line relationship as in 1. plus random noise. Plot these data.

```
In [15]: x2 = [1,2,3,4,5,6,7,8,9,10]
         y2 = x2 + 0.25*np.random.randn(len(x2))

         plt.plot(x2,y2)
         plt.title('international piracy over time')
         plt.xlabel('time')
         plt.ylabel('piracy')
```

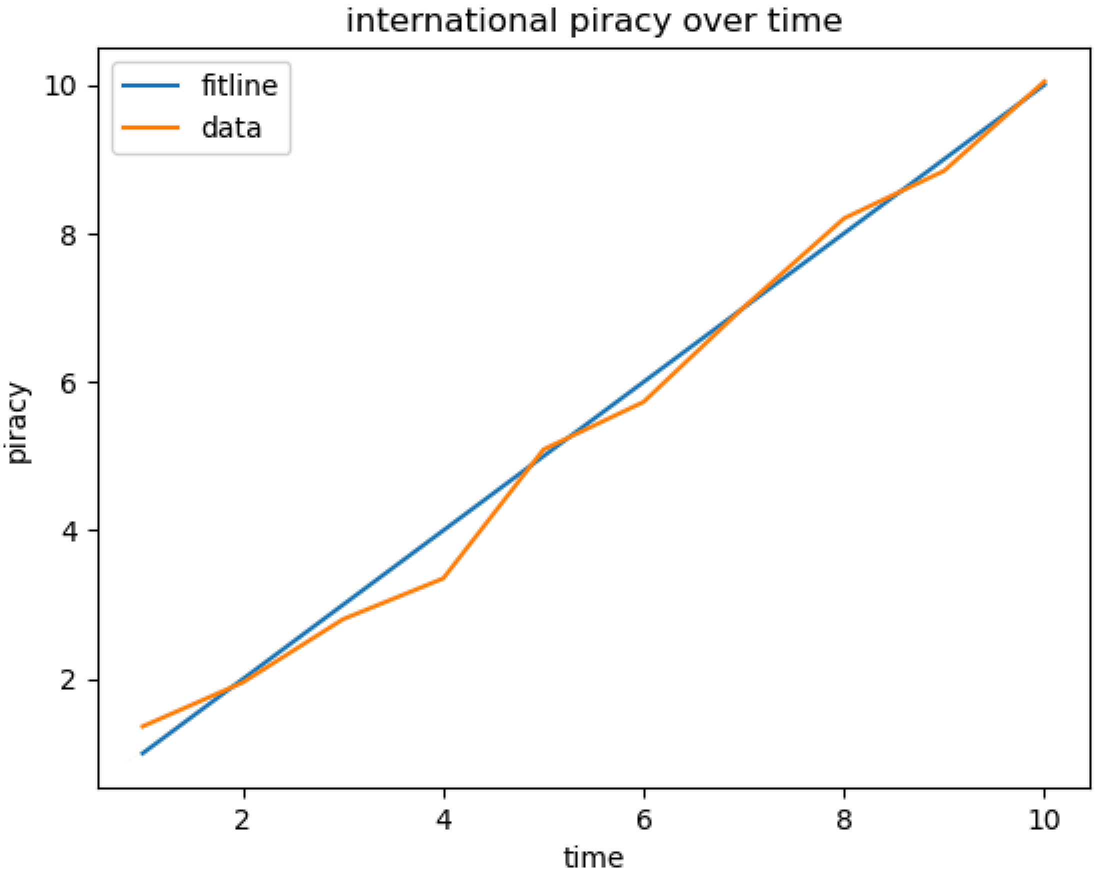
```
Out[15]: Text(0, 0.5, 'piracy')
```



3. Plot the straight line from 1. and the data from 2. on the same graph. Make sure to add the standard annotations, including a legend.

```
In [18]: plt.plot(x,x, label = 'fitline')
         plt.plot(x2,y2, label = 'data')

         plt.title('international piracy over time')
         plt.xlabel('time')
         plt.ylabel('piracy')
         plt.legend()
```

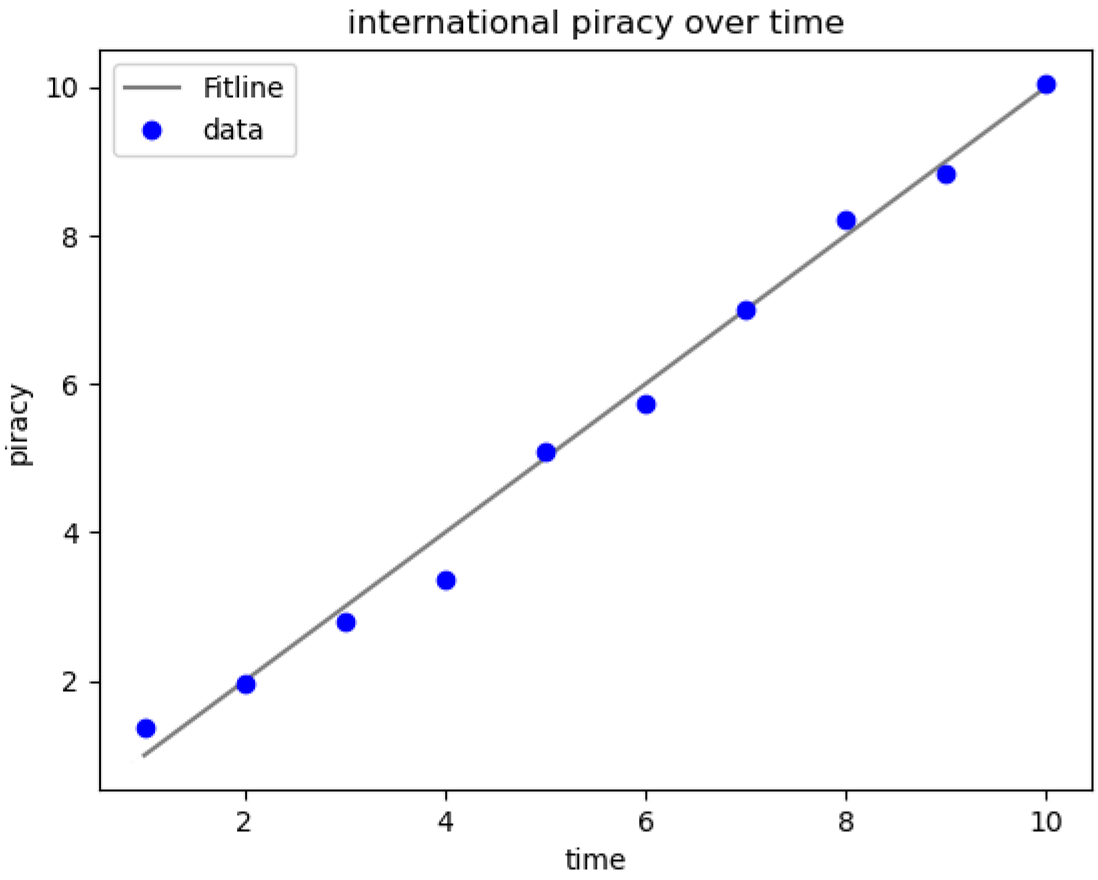
```
Out[18]: <matplotlib.legend.Legend at 0x126270a10>
```



4. Tinker around with your plot (colors, symbols, marker sizes, etc.) until you have a plot you would be happy to use in a presentation.

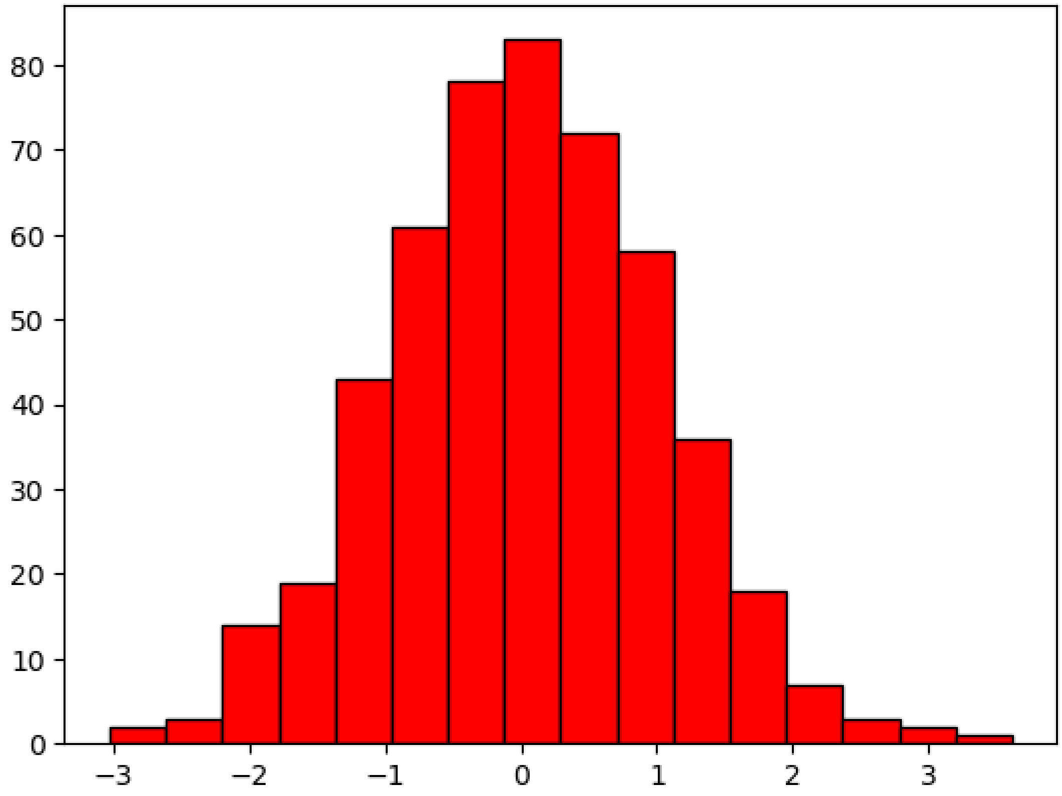
```
In [27]: plt.plot(x,x, label = 'Fitline', color = 'k', alpha = 0.5)
         plt.plot(x2,y2, 'bo', label = 'data')

         plt.title('international piracy over time')
         plt.xlabel('time')
         plt.ylabel('piracy')
         plt.legend()
```

```
Out[27]: <matplotlib.legend.Legend at 0x1266d5750>
```



5. Make 500 ***normally*** distributed random numbers and make a histogram of them.

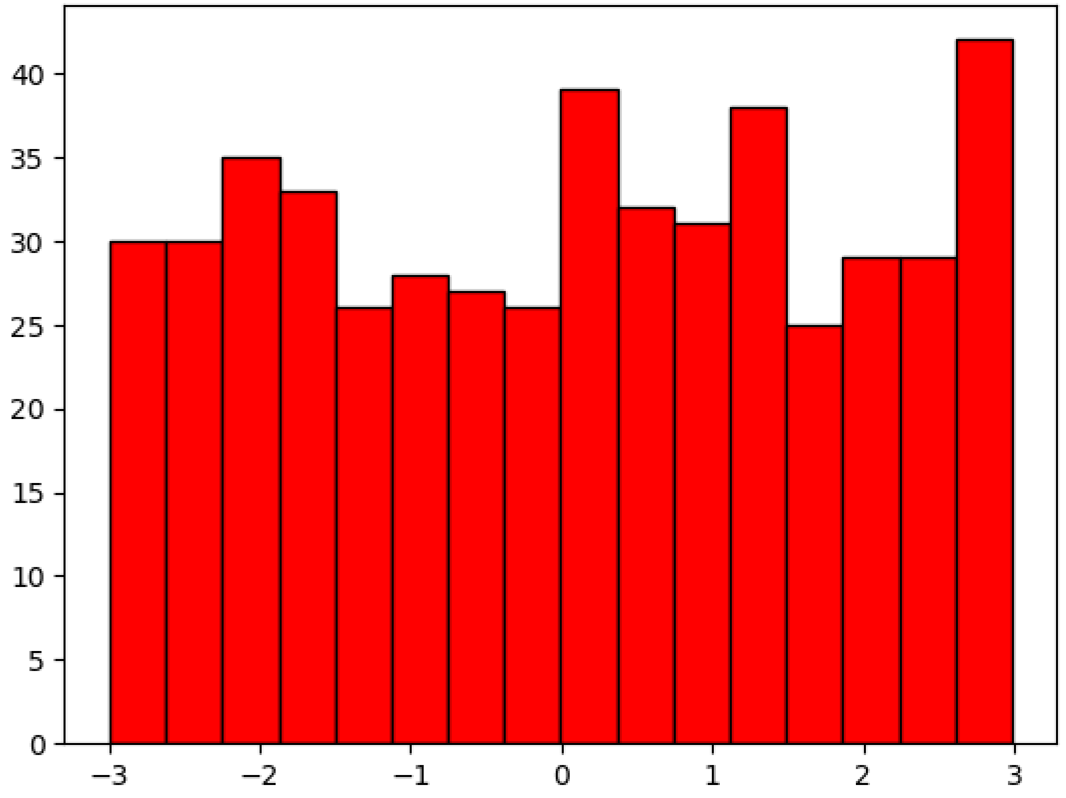
```
In [31]: data = np.random.randn(500)

         plt.hist(data, bins = 16, color = 'r', edgecolor = 'k');
```



6. Make 500 ***uniformly*** distributed random numbers (use `...rand()` instead of `...randn()`) and make a histogram of them.

In [51]:

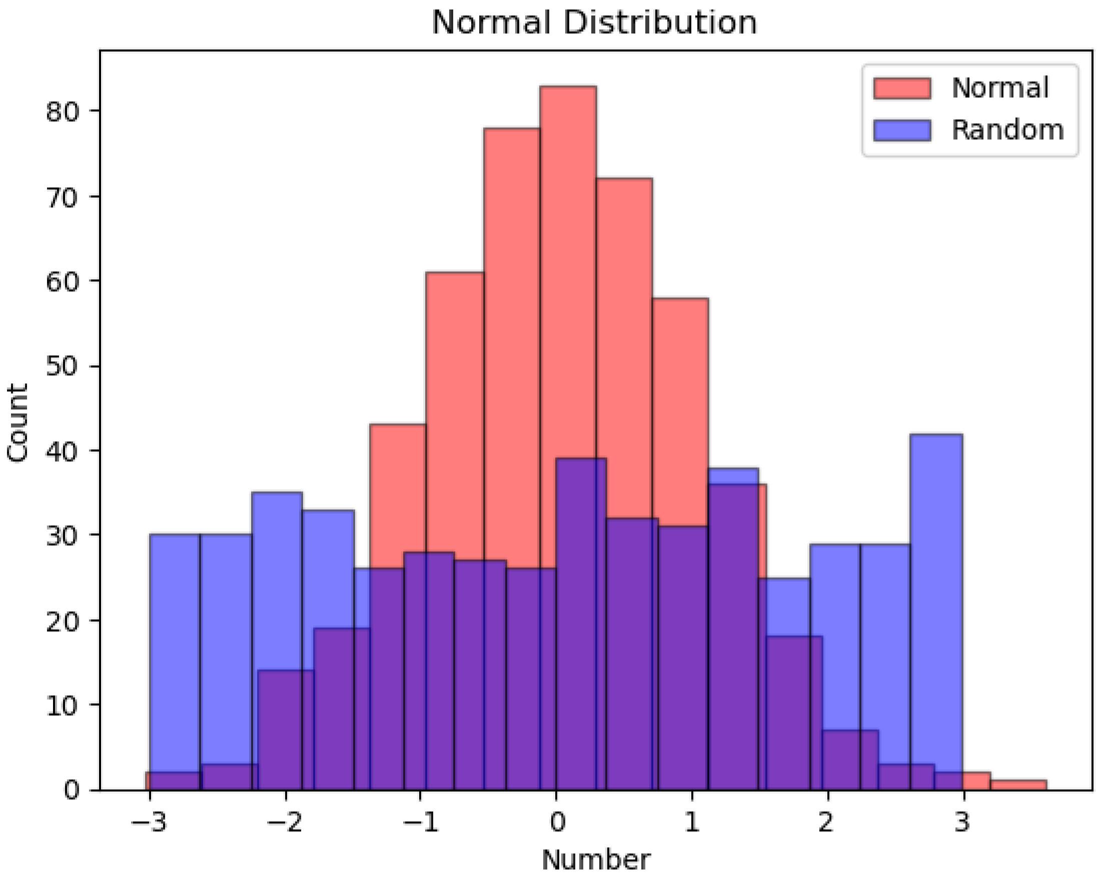
```
data2 = 6 * np.random.rand(500) - 3

plt.hist(data2, bins = 16, color = 'r', edgecolor = 'k');
```



7. Plot the histograms from 5. and 6. in the same axes to compare the two distributions. Tinker around with the `color =` and `alpha =` arguments to `plt.hist()` until you're happy with your figure. Don't forget the axis labels and a legend!

```
In [55]: plt.hist(data, bins = 16, color = 'r', edgecolor = 'k', alpha = 0.5, label =
         plt.hist(data2, bins = 16, color = 'b', edgecolor = 'k', alpha = 0.5, label


         plt.title('Normal Distribution')
         plt.xlabel('Number')
         plt.ylabel('Count')
         plt.legend()
```

```
Out[55]: <matplotlib.legend.Legend at 0x14579ed10>
```

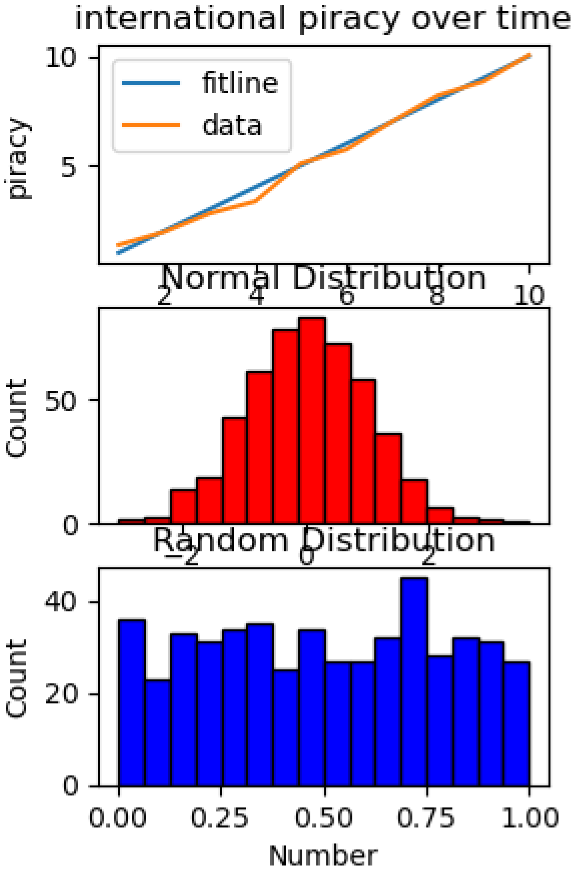


8. Make a figure with 3 subplots, the first containing the plot of the data with a straight line (from 3.), and the second and third containing each of the 2 histograms created in 5. and 6. Try a 3x1 and 1x3 layout and show your favorite.

In [42]:

```
#decided to use a 3x2 layout as to prefent the amount of stretch distortion,

plt.subplot(3,2,1)
plt.plot(x,x, label = 'fitline')
plt.plot(x2,y2, label = 'data')

plt.title('international piracy over time')
plt.xlabel('time')
plt.ylabel('piracy')
plt.legend()

plt.subplot(3,2,3)
plt.hist(data, bins = 16, color = 'r', edgecolor = 'k');
plt.title('Normal Distribution')
plt.xlabel('Number')
plt.ylabel('Count')
plt.subplot(3,2,5)

plt.hist(data2, bins = 16, color = 'b', edgecolor = 'k');
plt.title('Random Distribution')
plt.xlabel('Number')
plt.ylabel('Count')
```

```
Out[42]:  Text(0, 0.5, 'Count')
```


international piracy over time
piracy
fitline
data
Normal Distribution
Count
Random Distribution
Count
Number