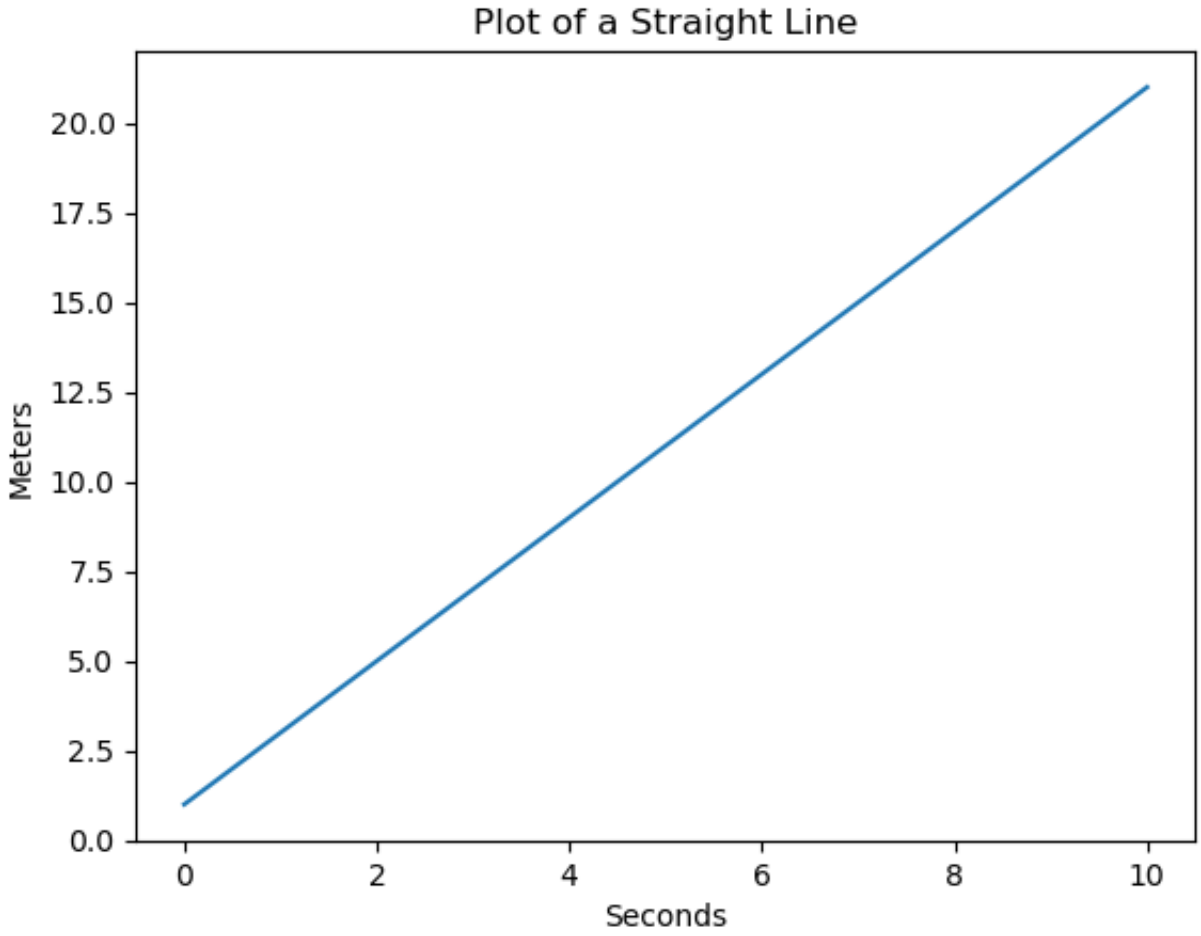
matplotlib homework

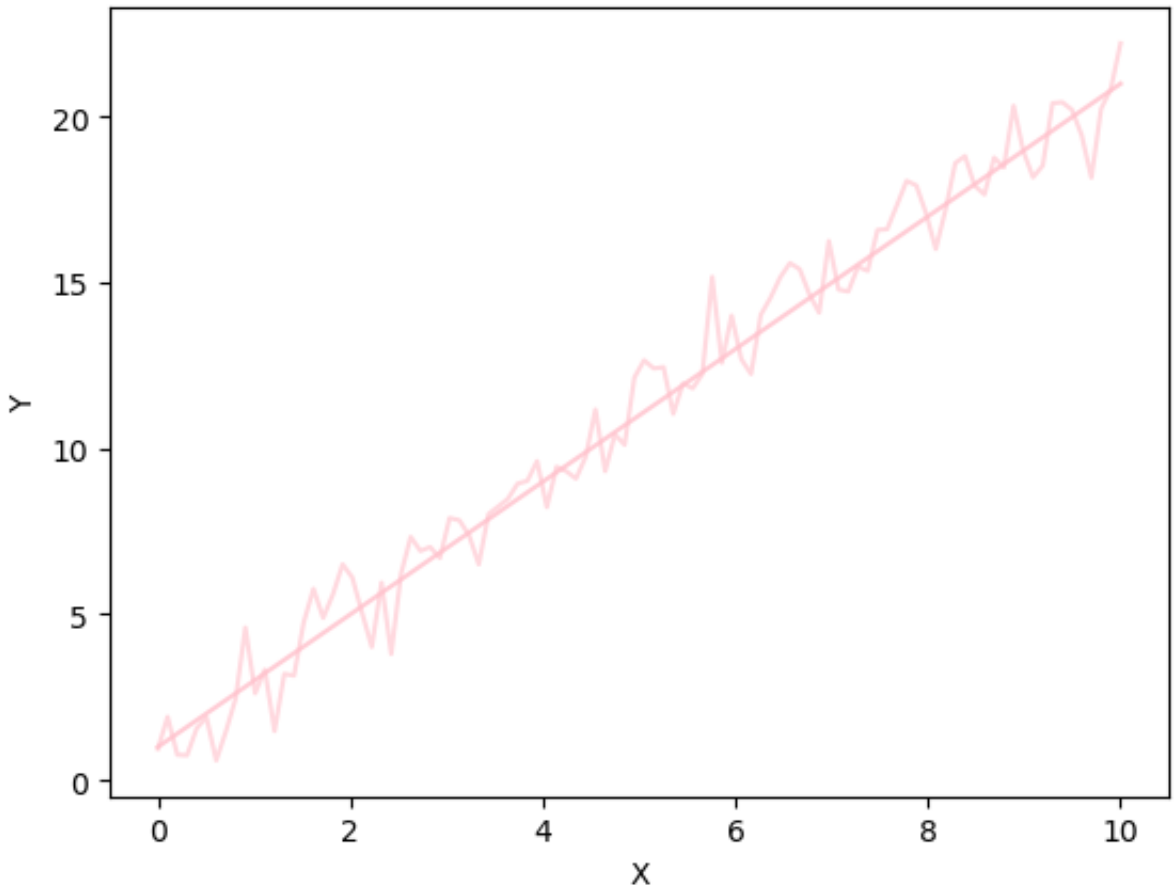
1. Make a plot of a straight line. Use `linspace()` to create the *x* values and the formula of a straight line, $y = a + bx$*, to create the *y* values (use an *a* and *b* of your choosing). You can pretend x and y are anything you like (x = time, y = international piracy or whatever).

In [107]:
```
import numpy as np
import matplotlib.pyplot as plt


a = 1
b = 2
x = np.linspace(0, 10, 100)
y = a + b * x
plt.plot(x, y)
plt.xlabel('Seconds')
plt.ylabel('Meters')
plt.title('Plot of a Straight Line')
plt.show()
```



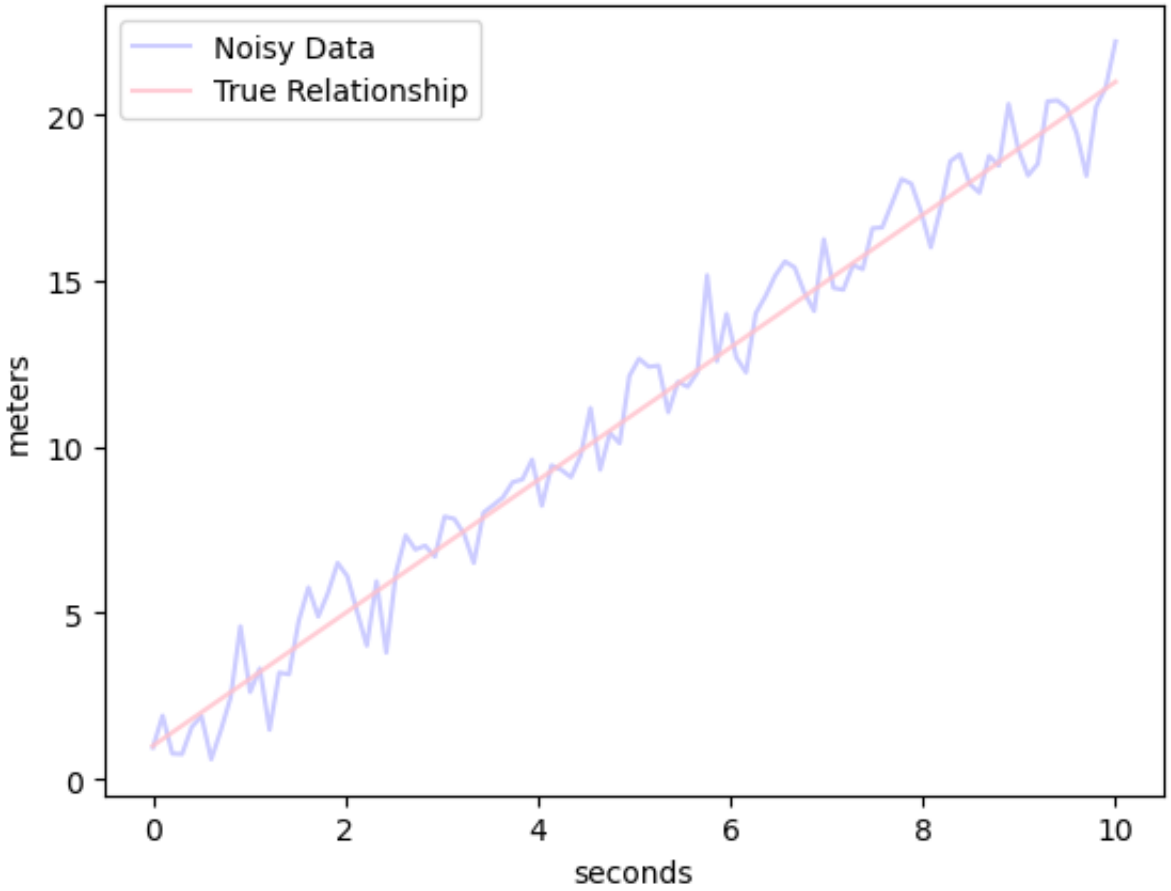
2. Make some data that are straight line values from the same straight line relationship as in 1. plus random noise. Plot these data.

In [108]:
```
x = np.linspace(0, 10, 100)
y = a + b * x + np.random.normal(0, 1, len(x))

plt.plot(x, y, 'pink', alpha = 0.6)
plt.plot(x, a + b * x, 'pink', alpha = 0.8)
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```



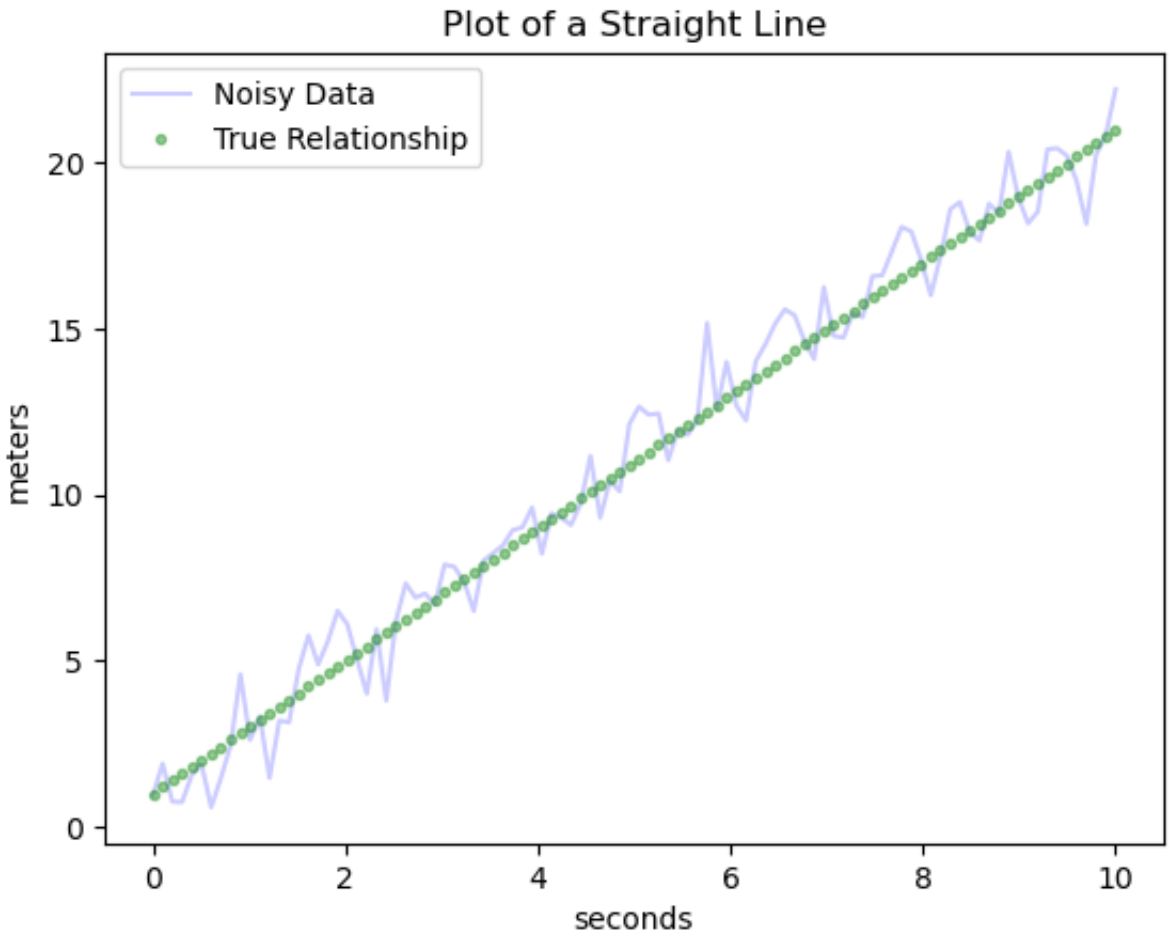
3. Plot the straight line from 1. and the data from 2. on the same graph. Make sure to add the standard annotations, including a legend.

In [110]:
```
plt.plot(x, y, 'blue', alpha=0.2, label='Noisy Data')
plt.plot(x, a + b * x, 'pink', alpha=0.8, label='True Relationship')
plt.xlabel('seconds')
plt.ylabel('meters')
plt.legend()
plt.show()
```



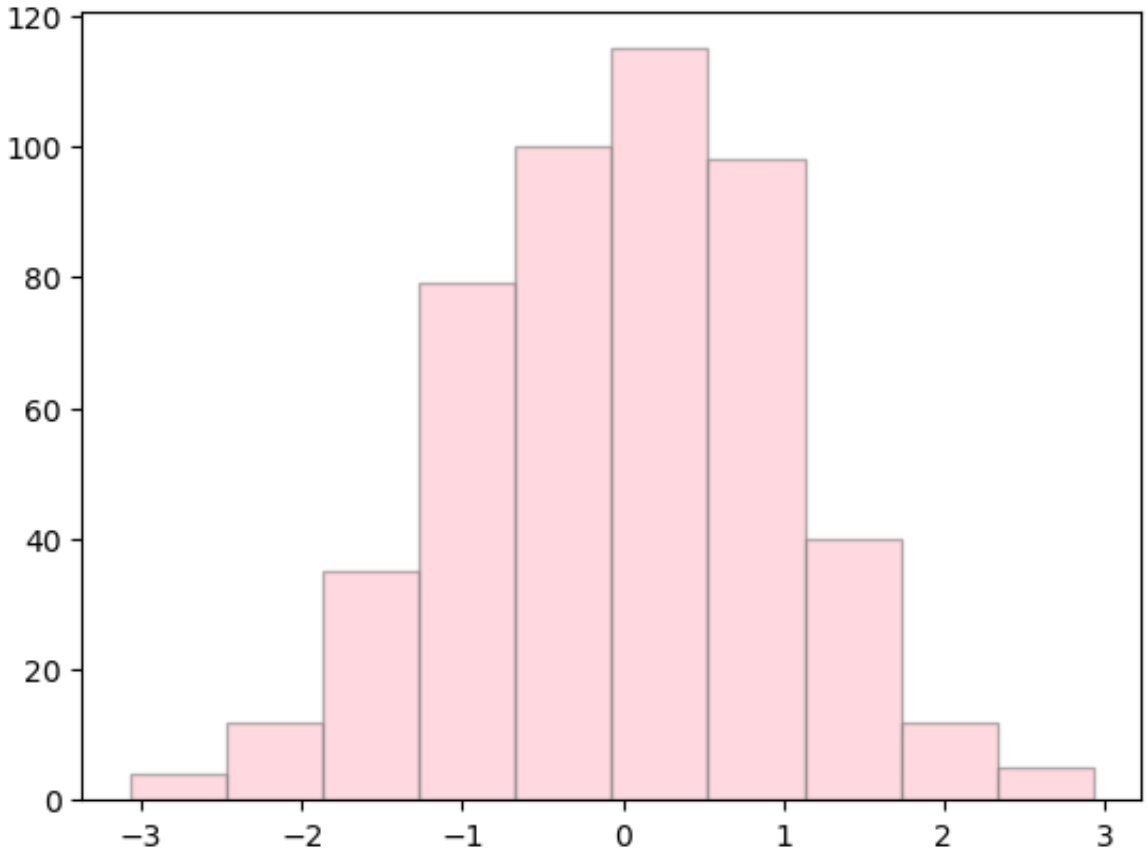
4. Tinker around with your plot (colors, symbols, marker sizes, etc.) until you have a plot you would be happy to use in a presentation.

In [111]:
```
plt.plot(x, y, 'b-', alpha=0.2, label='Noisy Data')
plt.plot(x, a + b * x, 'g.', alpha=0.4, label='True Relationship')
plt.xlabel('seconds')
plt.ylabel('meters')
plt.title('Plot of a Straight Line')
plt.legend()
plt.show()
```



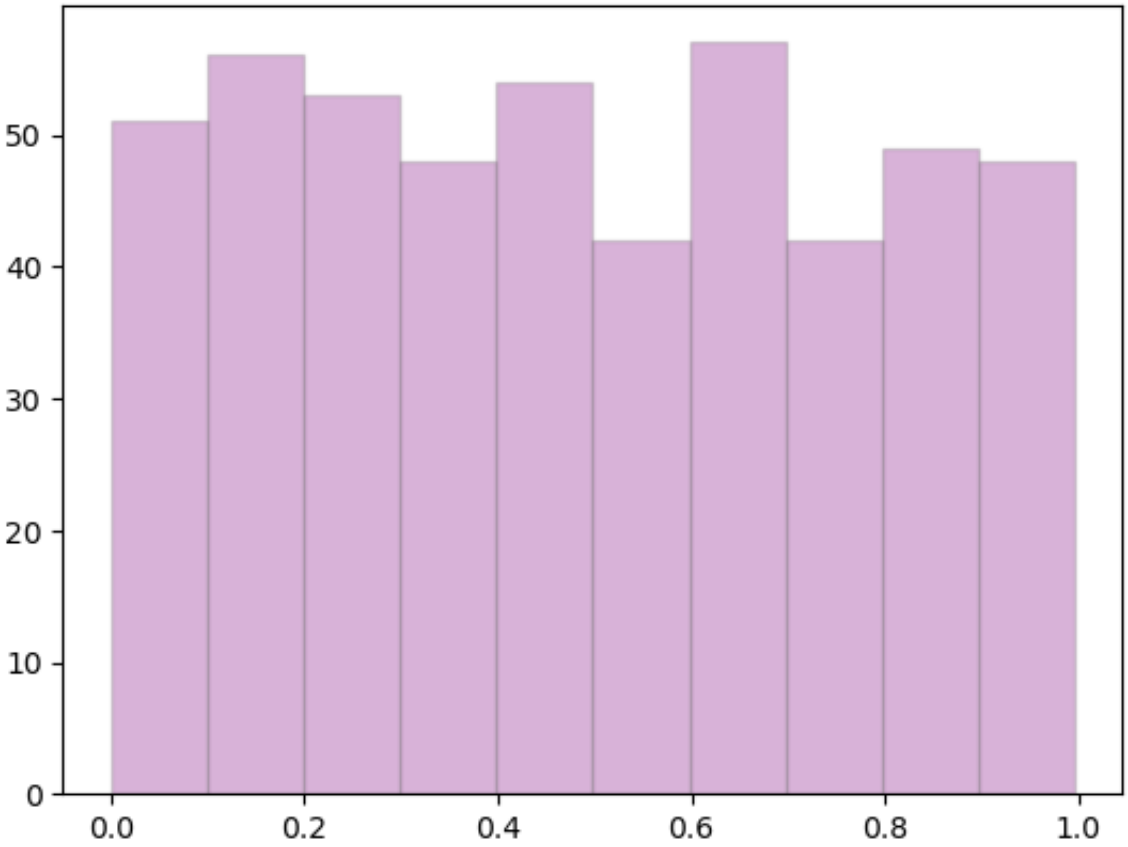
5. Make 500 ***normally*** distributed random numbers and make a histogram of them.

In [66]:
```
random_numbers = np.random.normal(size=500)

plt.hist(random_numbers, bins=10, color='pink', alpha= 0.6, edgecolor=
plt.show()
```



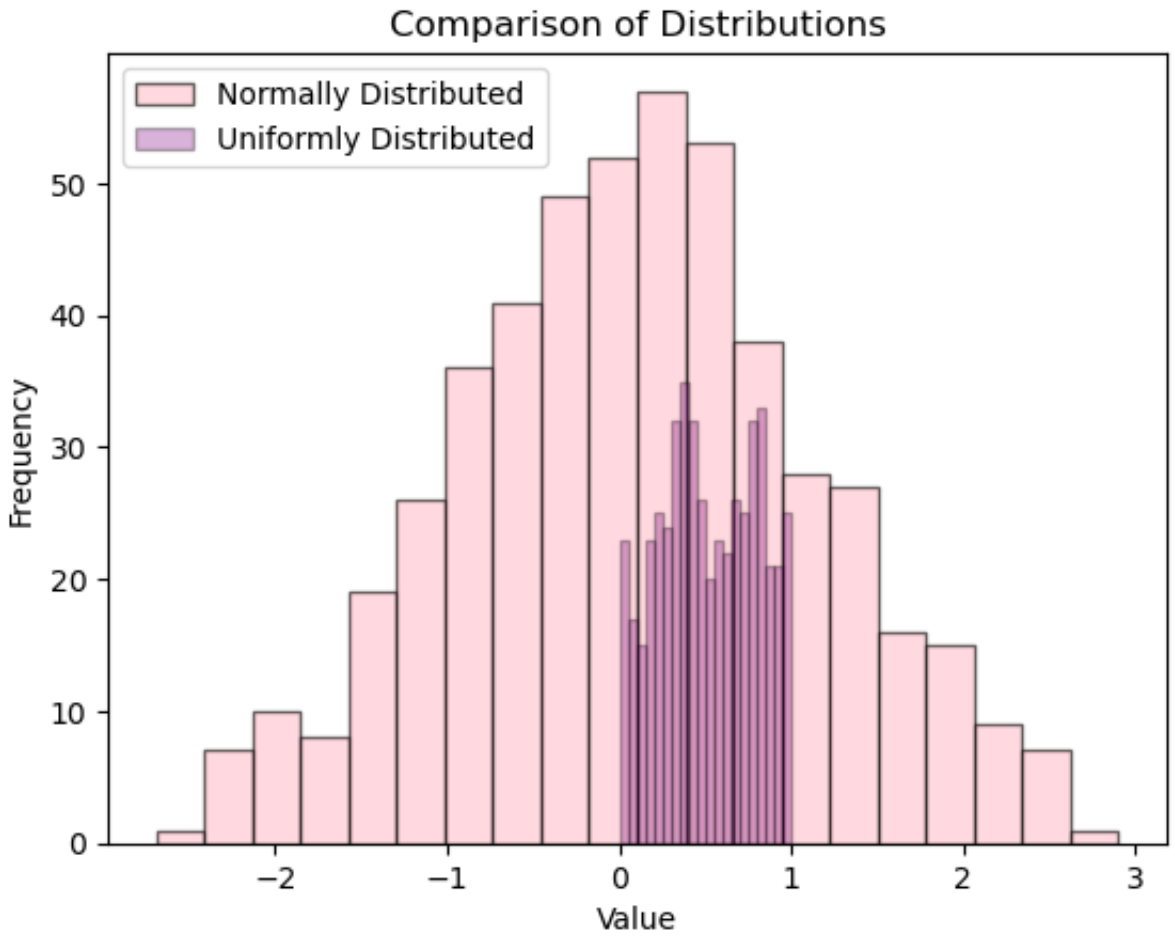
6. Make 500 ***uniformly*** distributed random numbers (use `...rand()` instead of `...randn()`) and make a histogram of them.

In [65]:
```
random_numbers = np.random.rand(500)

plt.hist(random_numbers, bins=10, color='purple', alpha= 0.3, edgecolo
plt.show()
```



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!

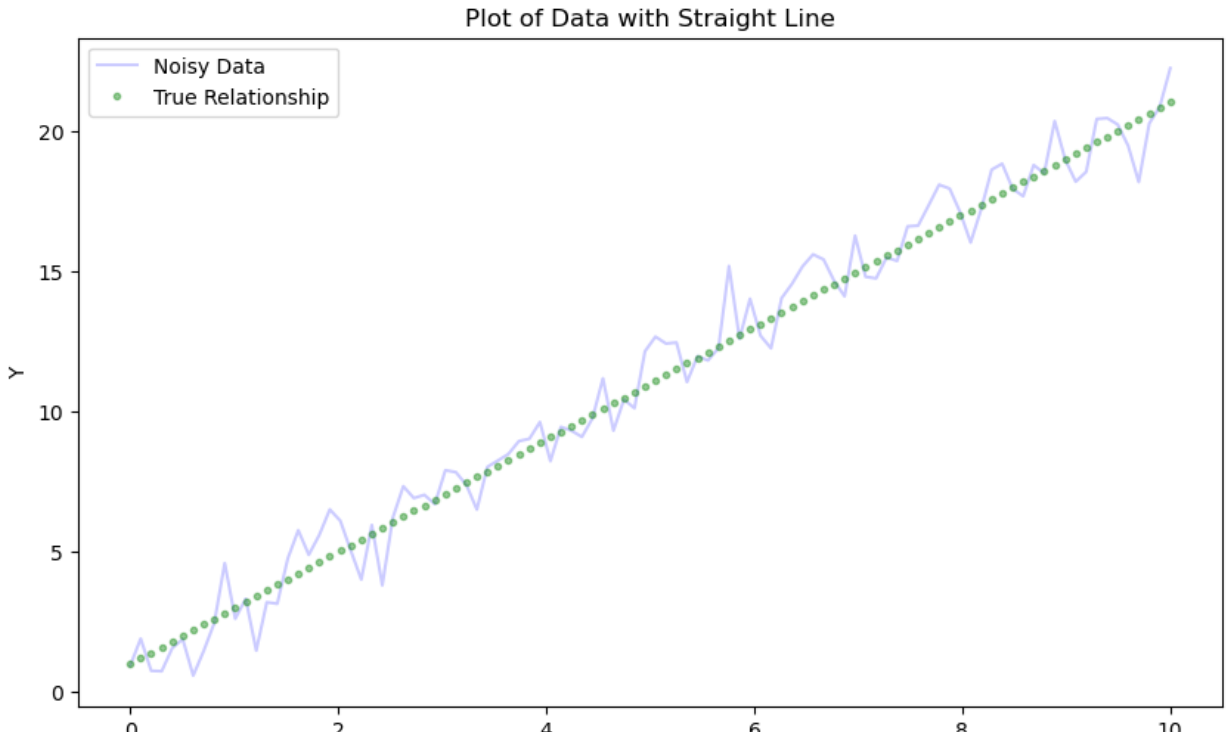
In [81]:
```
plt.hist(random_numbers_normal, bins=20, color='pink', edgecolor='blac
plt.hist(random_numbers_uniform, bins=20, color='purple', edgecolor='b
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Comparison of Distributions')
plt.legend()
plt.show()
```

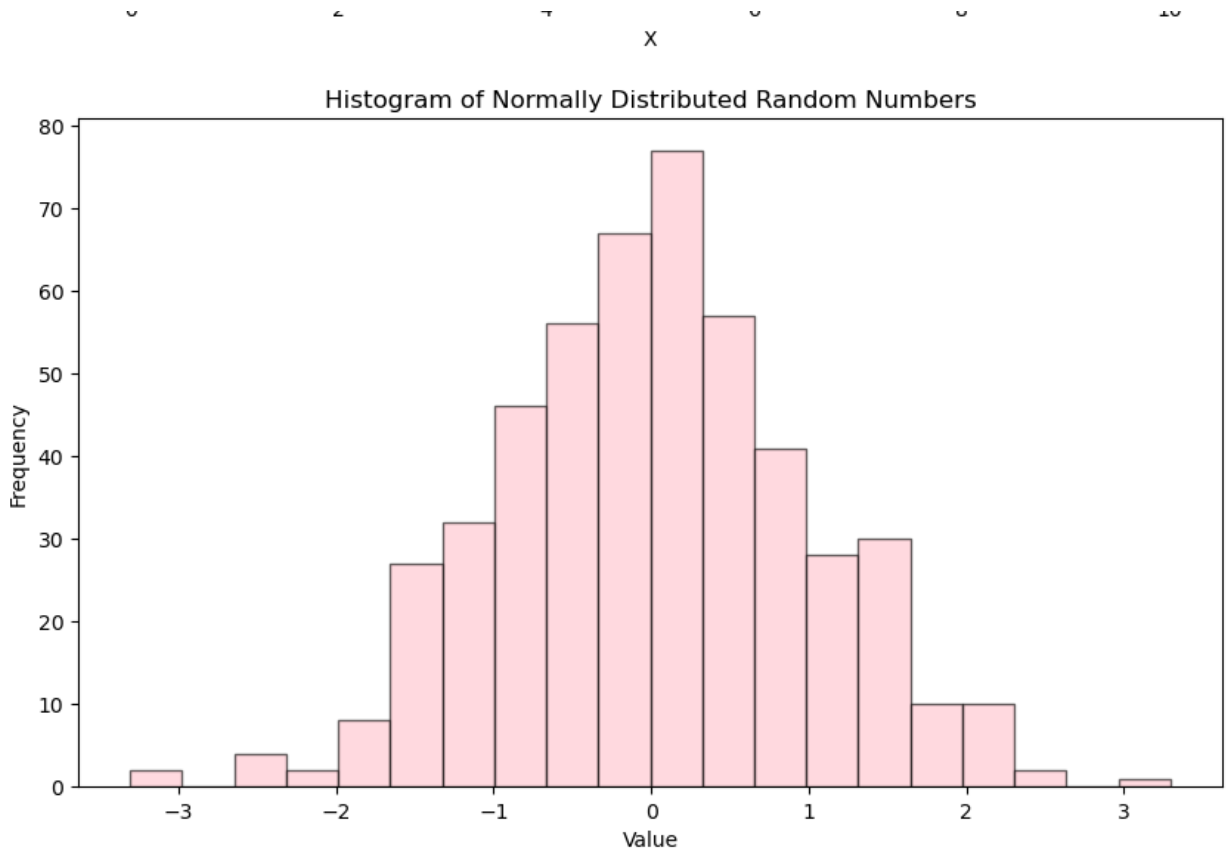


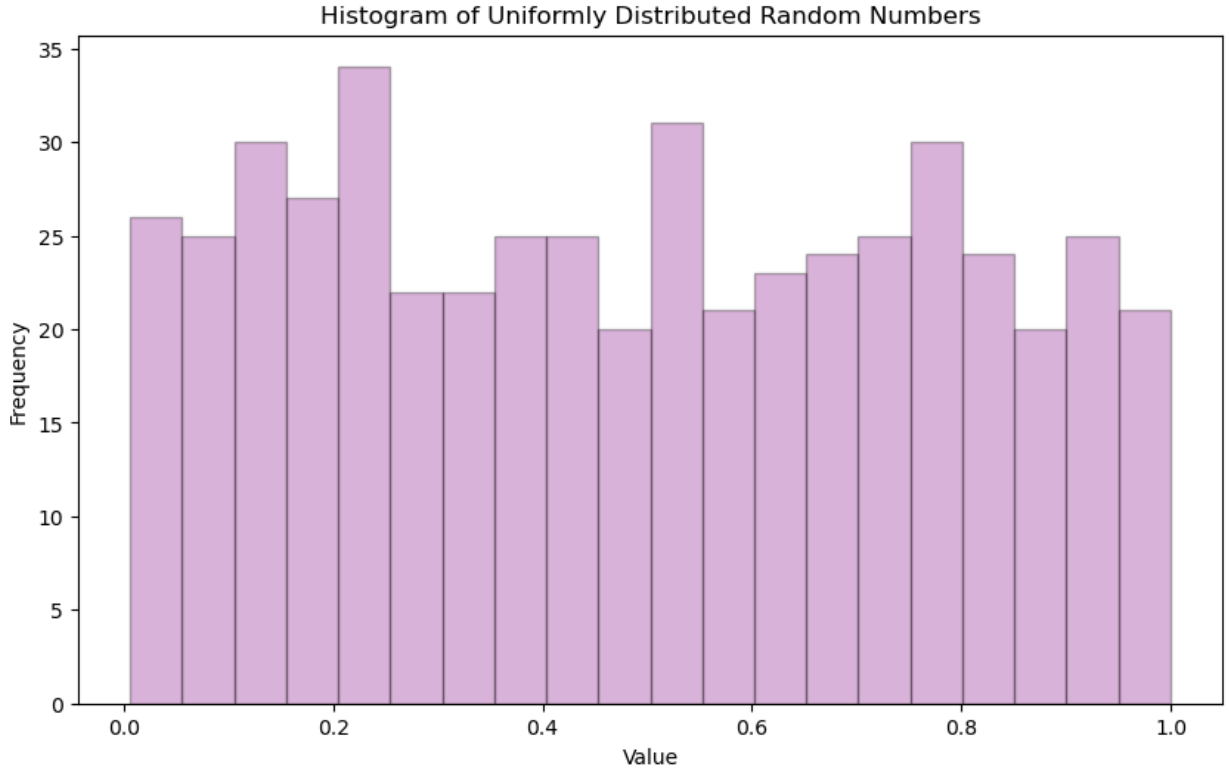
8. Make a figure with 3 subplots, the first containing the plot of the data with a straight line (from 3.), and the second and third containing each of the 2 histograms created in 5. and 6. Try a 3x1 and 1x3 layout and show your favorite.

In [112]:
```
```

```
plt.figure(figsize=(10, 20))


# Straight Line Plot
plt.subplot(3, 1, 1)
plt.plot(x, y, 'b-', alpha=0.2, label='Noisy Data')
plt.plot(x, a + b * x, 'g.', alpha=0.4, label='True Relationship')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Plot of Data with Straight Line')
plt.legend()

#Normal Plot
plt.subplot(3, 1, 2)
plt.hist(random_numbers_normal, bins=20, color='pink', edgecolor='blac

plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram of Normally Distributed Random Numbers')

#Uniform Plot
plt.subplot(3, 1, 3)
plt.hist(random_numbers_uniform, bins=20, color='purple', edgecolor='b
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram of Uniformly Distributed Random Numbers')


plt.show()
```




X
Histogram of Normally Distributed Random Numbers
Frequency
Value
0
10
20
30
40
50
60
70
80
−3
−2
−1
0
1
2
3


Histogram of Uniformly Distributed Random Numbers
Frequency
Value
0
5
10
15
20
25
30
35
0.0
0.2
0.4
0.6
0.8
1.0