matplotlib homework

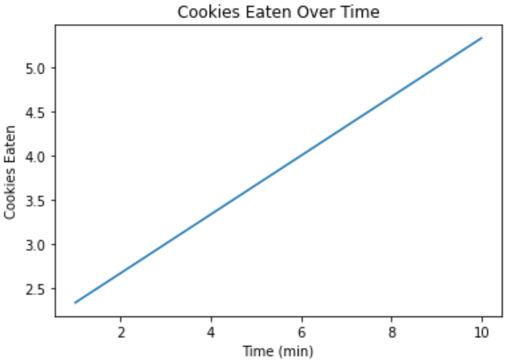
1. Make a plot of a straight line. Use `linspace()` to create the *x* values and the formula of a straight line, *y* = *a* + *b***x*, to create the *y* values (use an *a* and *b* of your choosing). You can pretend x and y are anything you like (x = time, y = international piracy or whatever).

In [9]:

```python
import numpy as np
import matplotlib.pyplot as plt

# make the data
x = np.linspace(1,10,10)
y = 2 + (1/3)*x

# make the plot
plt.plot(x,y)

# annotate
plt.title('Cookies Eaten Over Time')
plt.xlabel('Time (min)')
plt.ylabel('Cookies Eaten')
```

Out[9]:

```
Text(0, 0.5, 'Cookies Eaten')
```



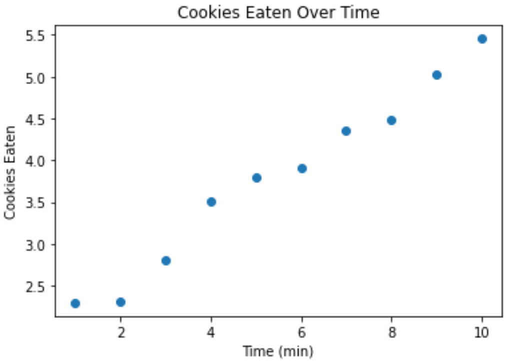
2. Make some data that are straight line values from the same straight line relationship as in 1. plus random noise. Plot these data.

In [17]:

```python
# add noise to the data
noise = .2*np.random.randn(len(y))
y2 = y + noise

# make the plot
plt.plot(x, y2, 'o')

# annotate
plt.title('Cookies Eaten Over Time')
plt.xlabel('Time (min)')
plt.ylabel('Cookies Eaten')
```

Out[17]:

```
Text(0, 0.5, 'Cookies Eaten')
```



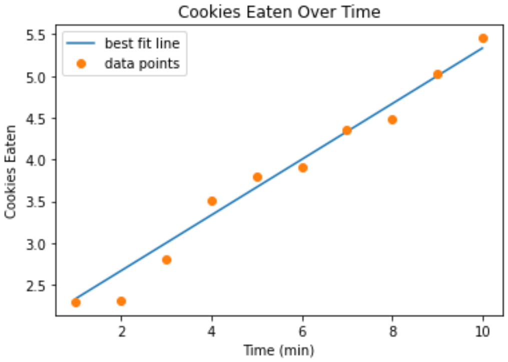
3. Plot the straight line from 1. and the data from 2. on the same graph. Make sure to add the standard annotations, including a legend.

In [18]:

```python
# make the plot
plt.plot(x,y, label = 'best fit line')
plt.plot(x, y2, 'o', label = 'data points')

# annotate
plt.title('Cookies Eaten Over Time')
plt.xlabel('Time (min)')
plt.ylabel('Cookies Eaten')
plt.legend()
```

Out[18]:

```
<matplotlib.legend.Legend at 0x21eb9eeaa90>
```



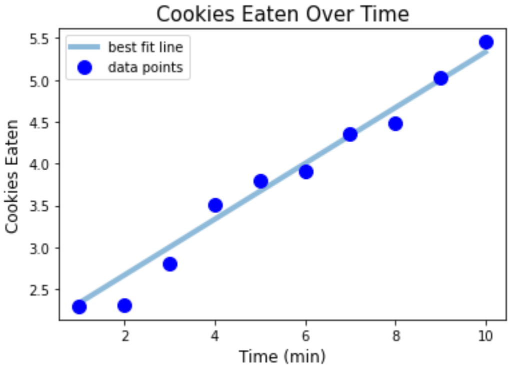
4. Tinker around with your plot (colors, symbols, marker sizes, etc.) until you have a plot you would be happy to use in a presentation.

In [30]:

```python
# make the plot
plt.plot(x, y, linewidth = 4, alpha = .5, label = 'best fit line')
plt.plot(x, y2, 'bo', markersize = 10, label = 'data points')

# annotate
plt.title('Cookies Eaten Over Time', fontsize=15)
plt.xlabel('Time (min)', fontsize = 12)
plt.ylabel('Cookies Eaten', fontsize = 12)
plt.legend()
```

Out[30]:

```
<matplotlib.legend.Legend at 0x21ebb311910>
```



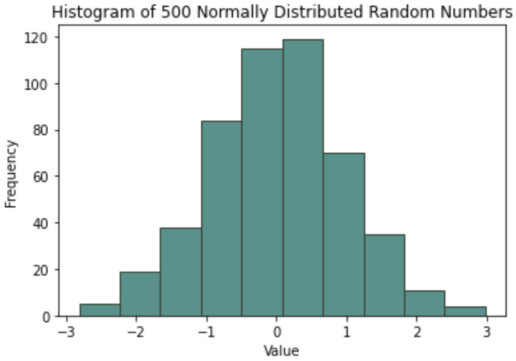
5. Make 500 ***normally*** distributed random numbers and make a histogram of them.

In [37]:

```
# data
normal_data = np.random.randn(500)

# plot
plt.hist(normal_data, color = '#5a918a', edgecolor = '#364236');

# annotate
plt.title('Histogram of 500 Normally Distributed Random Numbers')
plt.xlabel('Value')
plt.ylabel('Frequency')
```

Out[37]:

```
Text(0, 0.5, 'Frequency')
```



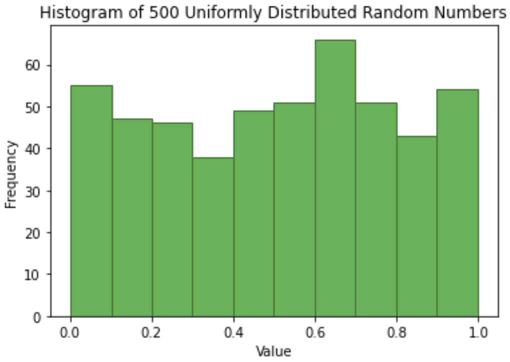
6. Make 500 ***uniformly*** distributed random numbers (use `...rand()` instead of `...randn()`) and make a histogram of them.

In [39]:

```
# data
uniform_numbers = np.random.rand(500)

# plot
plt.hist(uniform_numbers, color = '#69B259', edgecolor = '#437238');

# annotate
plt.title('Histogram of 500 Uniformly Distributed Random Numbers')
plt.xlabel('Value')
plt.ylabel('Frequency')
```

Out[39]:

```
Text(0, 0.5, 'Frequency')
```



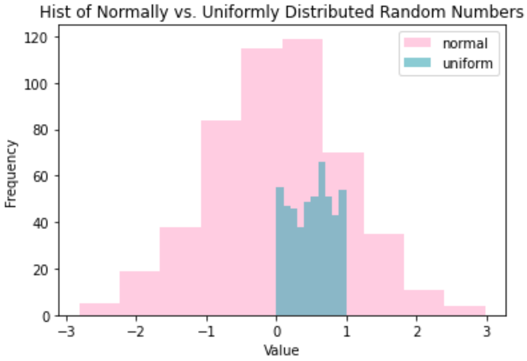
7. Plot the histograms from 5. and 6. in the same axes to compare the two distributions. Tinker around with the `color =` and `alpha =` arguments to `plt.hist()` until you're happy with your figure. Don't forget the axis labels and a legend!

In [67]:

```python
# plot
plt.hist(normal_data, color = '#FF569E', alpha = .3, label = 'normal', histtype = 'stepfilled', linewidth = 3);
plt.hist(uniform_numbers, color = '#3CA9B6', alpha = .6, label = 'uniform', histtype = 'stepfilled', linewidth = 3);

# annotate
plt.title('Hist of Normally vs. Uniformly Distributed Random Numbers')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.legend()
```

Out[67]:

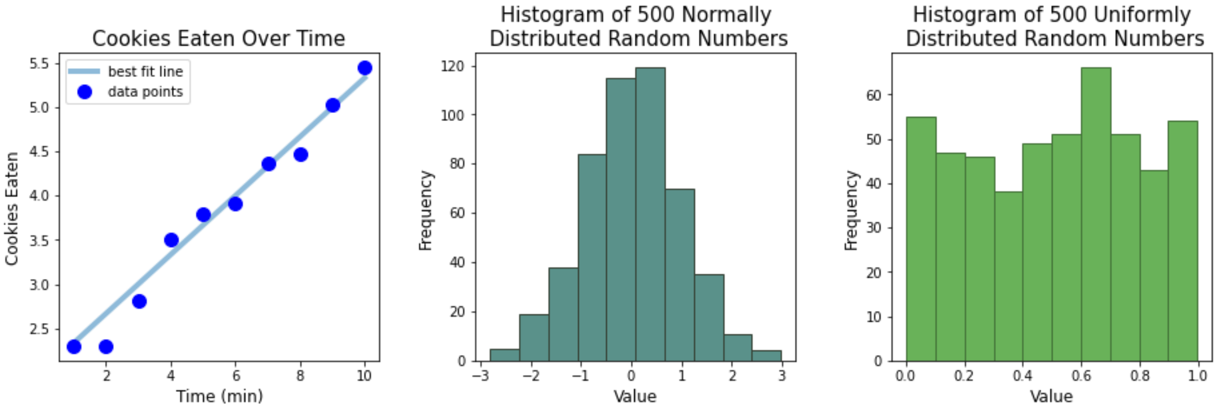
```
<matplotlib.legend.Legend at 0x21ebcd22790>
```



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 [88]:

```python
# altering figsize of the subplots
plt.subplots(1,3,figsize=(15,4))

plt.subplot(1,3,1)
# make the plot
plt.plot(x, y, linewidth = 4, alpha = .5, label = 'best fit line')
plt.plot(x, y2, 'bo', markersize = 10, label = 'data points')
# annotate
plt.title('Cookies Eaten Over Time', fontsize=15)
plt.xlabel('Time (min)', fontsize = 12)
plt.ylabel('Cookies Eaten', fontsize = 12)
plt.legend()

plt.subplot(1,3,2)
# plot
plt.hist(normal_data, color = '#5a918a', edgecolor = '#364236');
# annotate
plt.title('Histogram of 500 Normally\n Distributed Random Numbers', fontsize=15)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

plt.subplot(1,3,3)
# plot
plt.hist(uniform_numbers, color = '#69B259', edgecolor = '#437238');
# annotate
plt.title('Histogram of 500 Uniformly\n Distributed Random Numbers', fontsize=15)
plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

# adjusting the spacing between subplots so everything is readable
plt.subplots_adjust(left=0.1,
                    bottom=0.1,
                    right=0.9,
                    top=0.9,
                    wspace=0.3,
                    hspace=0.5)
```


Cookies Eaten Over Time
best fit line
data points
Cookies Eaten
Time (min)
Histogram of 500 Normally
Distributed Random Numbers
Frequency
Value
Histogram of 500 Uniformly
Distributed Random Numbers
Frequency
Value