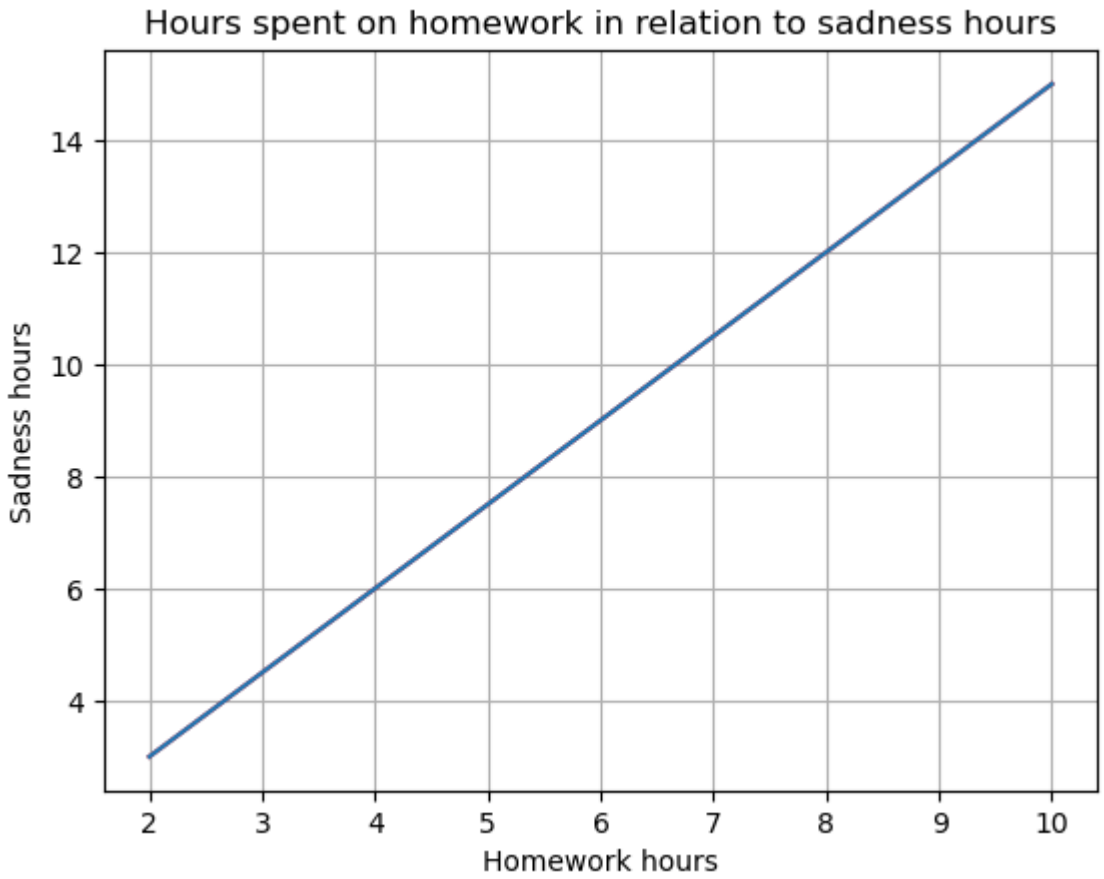
matplotlib homework

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).

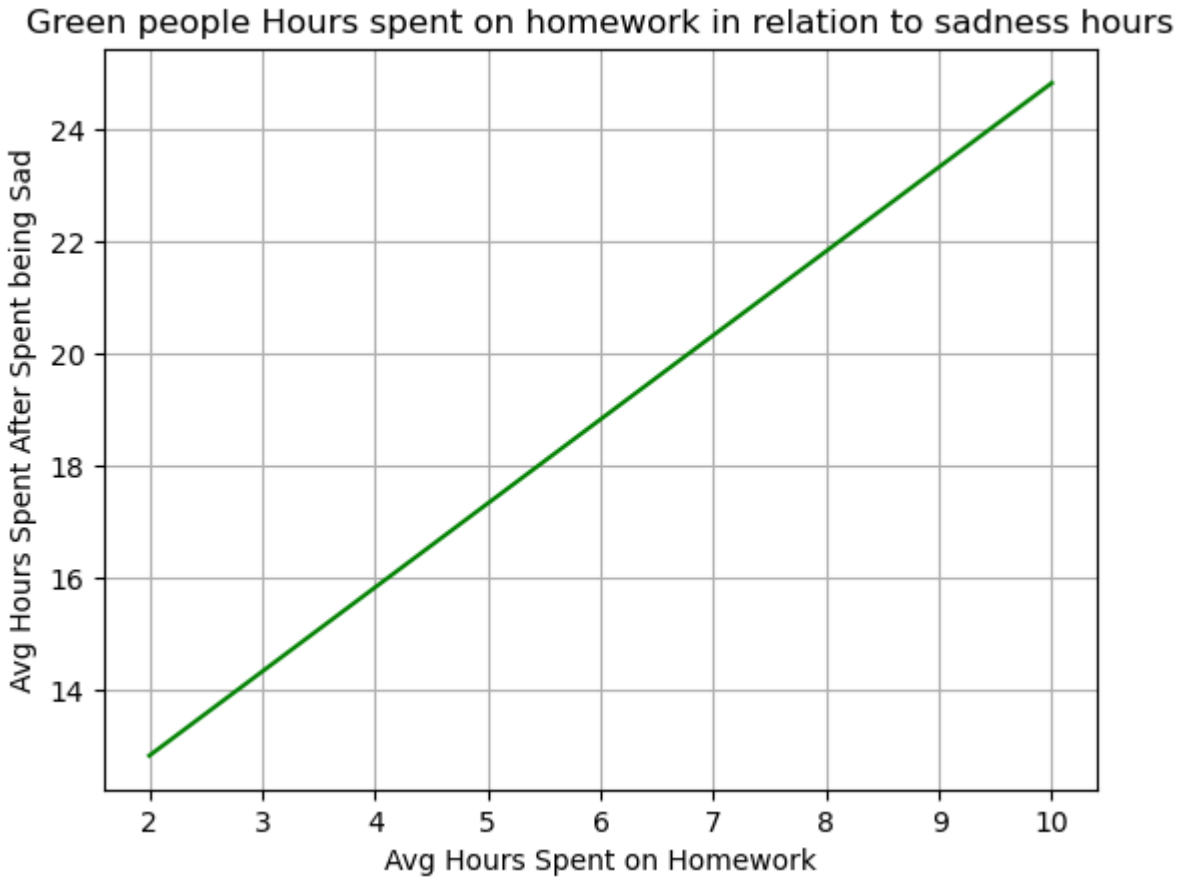
In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [30]:
```
x = np.array([2, 4, 6, 8, 10])
y = np.array([3, 6, 9, 12, 15])

plt.plot(x, y,'b')
m, b = np.polyfit(x, y, 1)

plt.plot(x, m*x+b)

plt.title('Hours spent on homework in relation to sadness hours')
plt.xlabel('Homework hours')
plt.ylabel('Sadness hours')
plt.grid()
plt.show()
```



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

In [27]:
```
b = np.array([2, 4, 6, 8, 10])
c = np.array([3, 6, 9, 12, 15])
noise = np.random.normal(10)
noised = c + noise

plt.plot(b, noised,'green')
m, b = np.polyfit(b, c, 1)

plt.title('Green people Hours spent on homework in relation to sadness hours')
plt.xlabel('Avg Hours Spent on Homework')
plt.ylabel('Avg Hours Spent After Spent being Sad')
plt.grid()
plt.show()
```



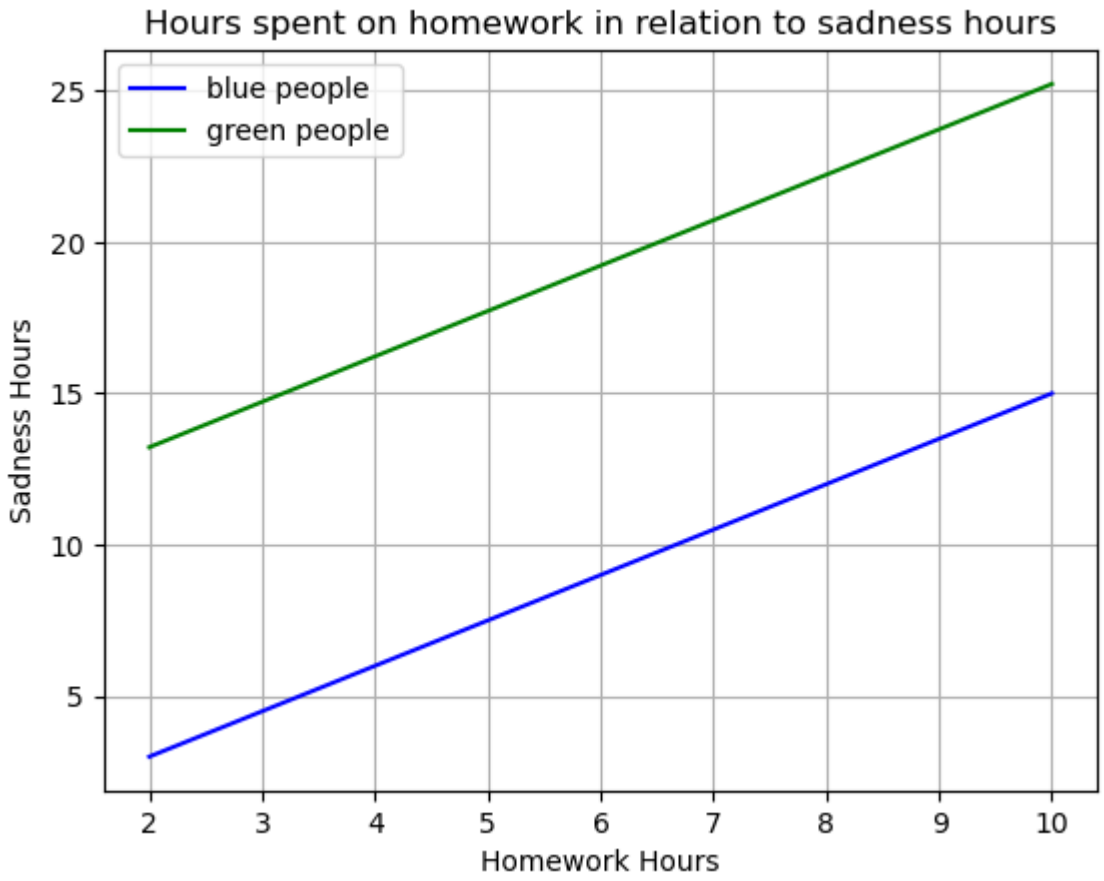
1. Plot the straight line from 1. and the data from 2. on the same graph. Make sure to add the standard annotations, including a legend.

In [ ]:
```
#Just semi roughing it out
x = np.array([2, 4, 6, 8, 10])
y = np.array([3, 6, 9, 12, 15])

plt.plot(x, y,'red')
m, b = np.polyfit(x, y, 1)
```

```
plt.plot(x, m*x+b)

plt.title('Hours spent on homework in relation to sadness hours')
plt.xlabel('Homework hours')
plt.ylabel('Sadness hours')
plt.grid()
plt.show()

b = np.array([2, 4, 6, 8, 10])
c = np.array([3, 6, 9, 12, 15])
noise = np.random.normal(10)
noised = c + noise

plt.plot(b, noised,'green')
m, b = np.polyfit(b, c, 1)

plt.title('Hours spent on homework in relation to sadness hours')
plt.xlabel('Avg Hours Spent on Homework')
plt.ylabel('Avg Hours Spent After Spent being Sad')
plt.grid()
plt.show()
```

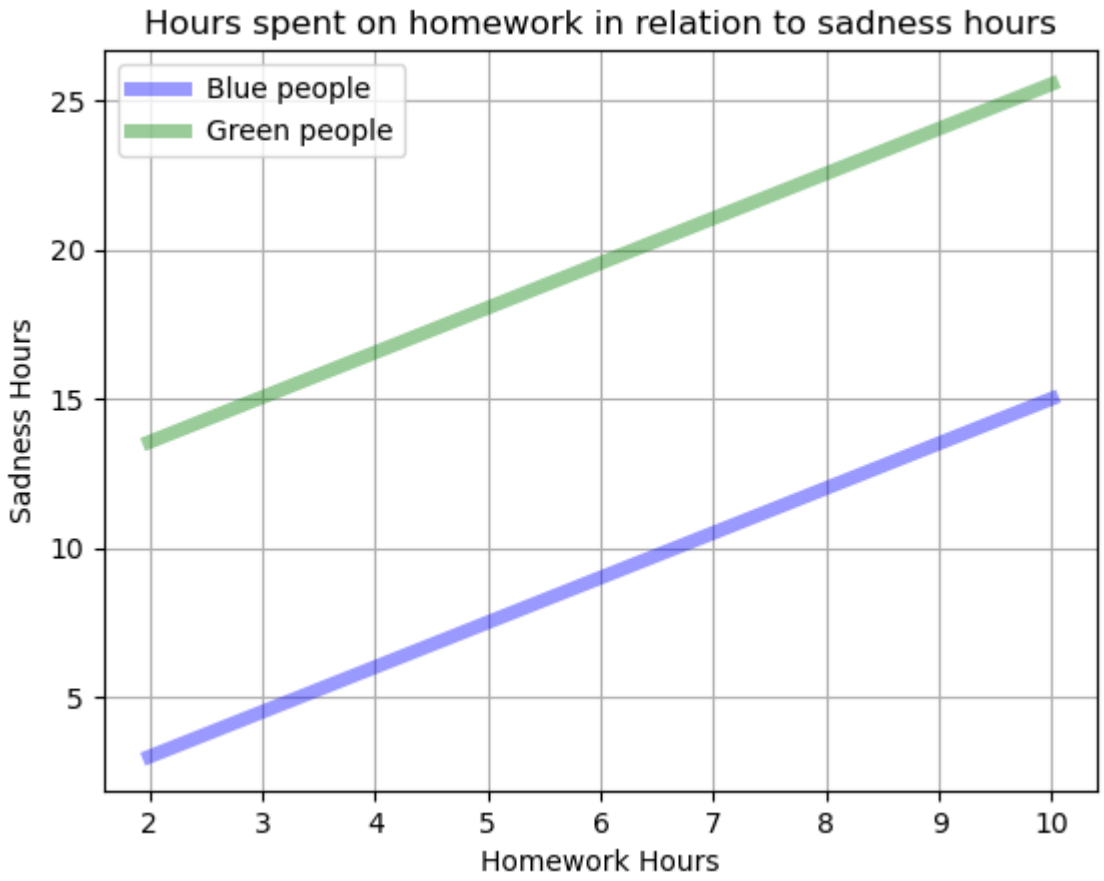
In [46]:

```
#The real chart code
x = np.array([2, 4, 6, 8, 10])
y = np.array([3, 6, 9, 12, 15])


b = np.array([2, 4, 6, 8, 10])
c = np.array([3, 6, 9, 12, 15])
noise = np.random.normal(10)
noised = c + noise

plt.plot(x, y, 'b', label = 'blue people')
plt.plot(b, noised, 'green', label = 'green people')
plt.title('Hours spent on homework in relation to sadness hours')
plt.xlabel('Homework Hours')
plt.ylabel('Sadness Hours')
plt.grid()
plt.legend()
plt.show()
```



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

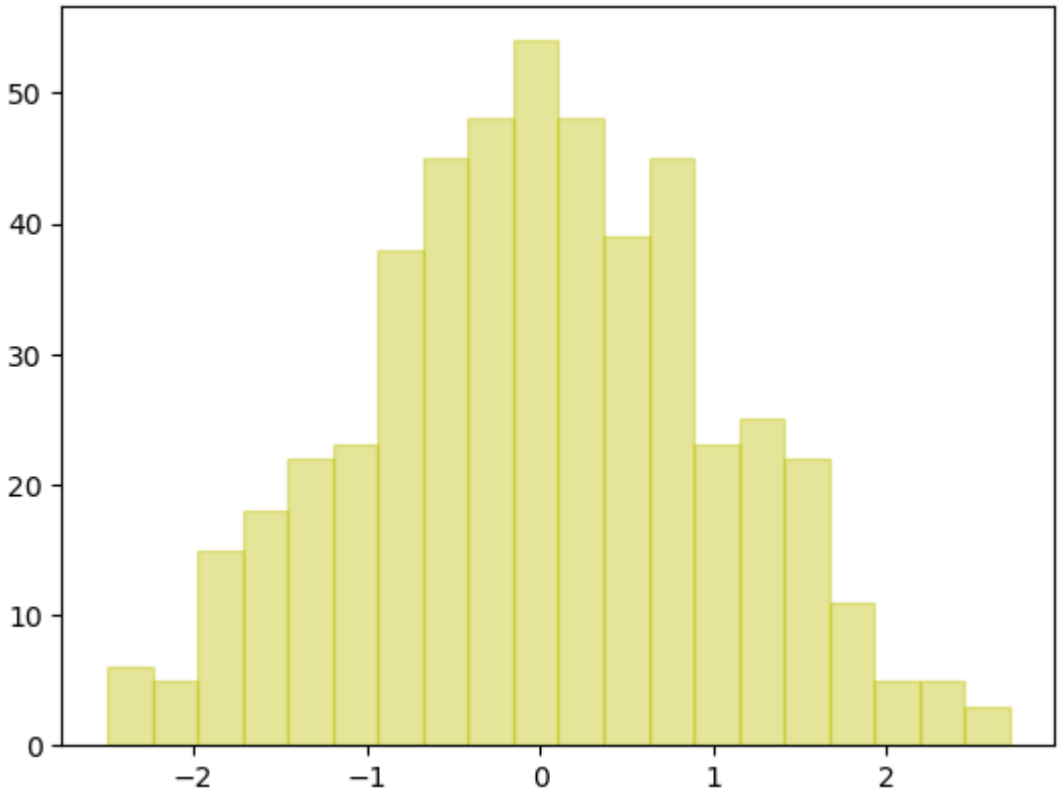
In [48]:
```
x = np.array([2, 4, 6, 8, 10])
y = np.array([3, 6, 9, 12, 15])


b = np.array([2, 4, 6, 8, 10])
c = np.array([3, 6, 9, 12, 15])
noise = np.random.normal(10)
noised = c + noise

plt.plot(x, y, 'b', linewidth = 5, alpha = 0.4, label = 'Blue people')
plt.plot(b, noised, 'green', linewidth = 5, alpha = 0.4, label = 'Green people')
plt.title('Hours spent on homework in relation to sadness hours')
plt.xlabel('Homework Hours')
plt.ylabel('Sadness Hours')
plt.grid()
plt.legend()
plt.show()
```



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

In [51]:
```
bigdata = np.random.randn(500)
plt.hist(bigdata, bins = 20, color = 'y', edgecolor = 'y', alpha = 0.4)
```

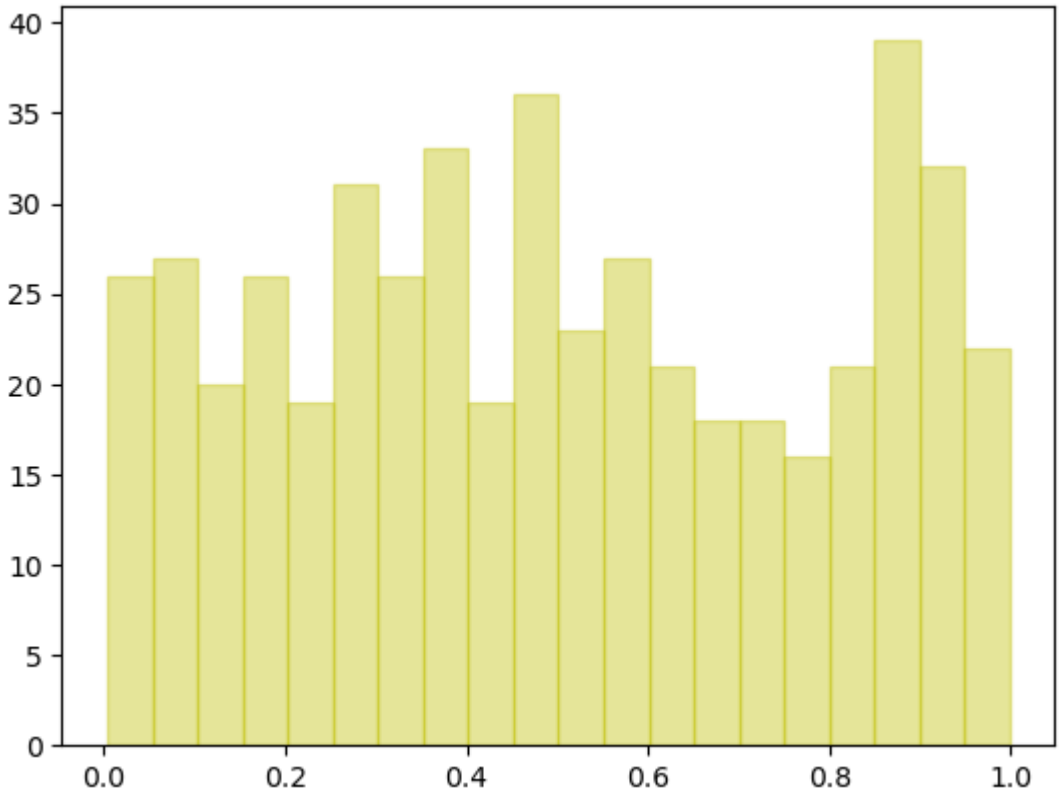
Out[51]:
```
(array([ 6.,  5., 15., 18., 22., 23., 38., 45., 48., 54., 48., 39., 45.,
        23., 25., 22., 11.,  5.,  5.,  3.]),
 array([-2.50009537, -2.23926202, -1.97842868, -1.71759533, -1.45676198,
        -1.19592863, -0.93509529, -0.67426194, -0.41342859, -0.15259524,
         0.10823811,  0.36907145,  0.6299048 ,  0.89073815,  1.1515715 ,
         1.41240484,  1.67323819,  1.93407154,  2.19490489,  2.45573823,
         2.71657158]),
 <BarContainer object of 20 artists>)
```



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

In [53]:
```
bigdata2 = np.random.rand(500)
plt.hist(bigdata2, bins = 20, color = 'y', edgecolor = 'y', alpha = 0.4)
```

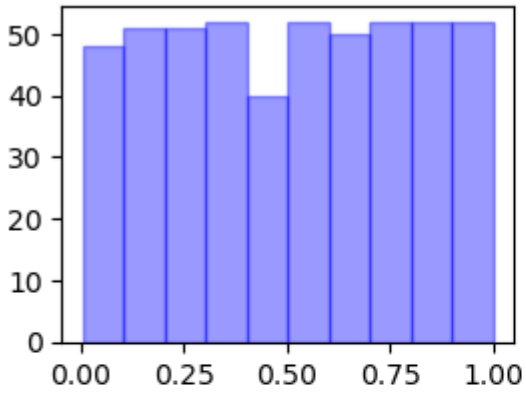
Out[53]:
```
(array([26., 27., 20., 26., 19., 31., 26., 33., 19., 36., 23., 27., 21.,
        18., 18., 16., 21., 39., 32., 22.]),
 array([0.00336974, 0.0531539 , 0.10293806, 0.15272222, 0.20250638,
        0.25229054, 0.3020747 , 0.35185886, 0.40164302, 0.45142718,
        0.50121134, 0.5509955 , 0.60077965, 0.65056381, 0.70034797,
        0.75013213, 0.79991629, 0.84970045, 0.89948461, 0.94926877,
        0.99905293]),
 <BarContainer object of 20 artists>)
```

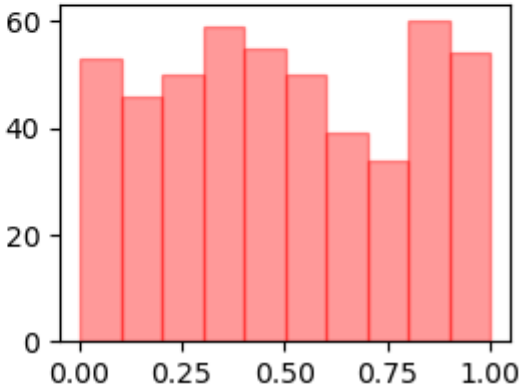


1. Plot the histograms from 5. and 6. in the same axes to compare the two distributions. Tinker around with the `color =` and `alpha =` arguments to `plt.hist()` until you're happy with your figure. Don't forget the axis labels and a legend!

In [75]:
```
plt.subplot(2,2,1)
plt.hist(bigdata, color = 'b', edgecolor = 'b', alpha = 0.4)
plt.subplot(2,2,4)
plt.hist(bigdata2, color = 'r', edgecolor = 'r', alpha = 0.4)
```

Out[75]:
```
(array([53., 46., 50., 59., 55., 50., 39., 34., 60., 54.]),
 array([0.00336974, 0.10293806, 0.20250638, 0.3020747 , 0.40164302,
        0.50121134, 0.60077965, 0.70034797, 0.79991629, 0.89948461,
        0.99905293]),
 <BarContainer object of 10 artists>)
```

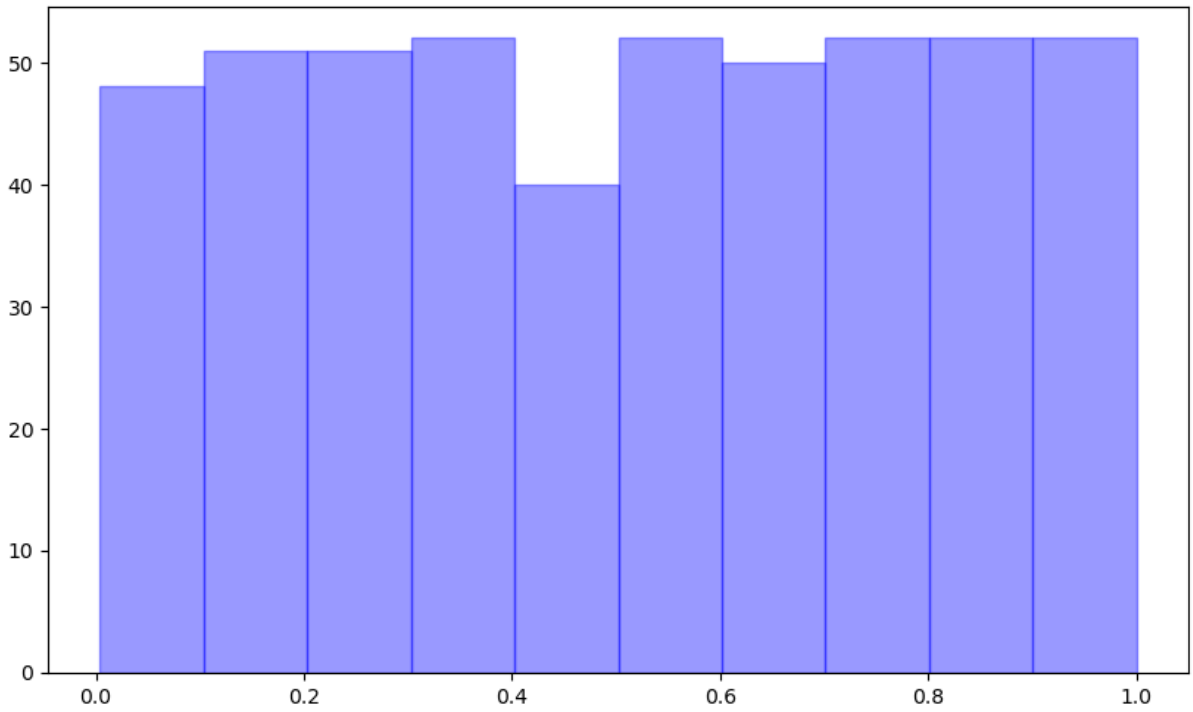


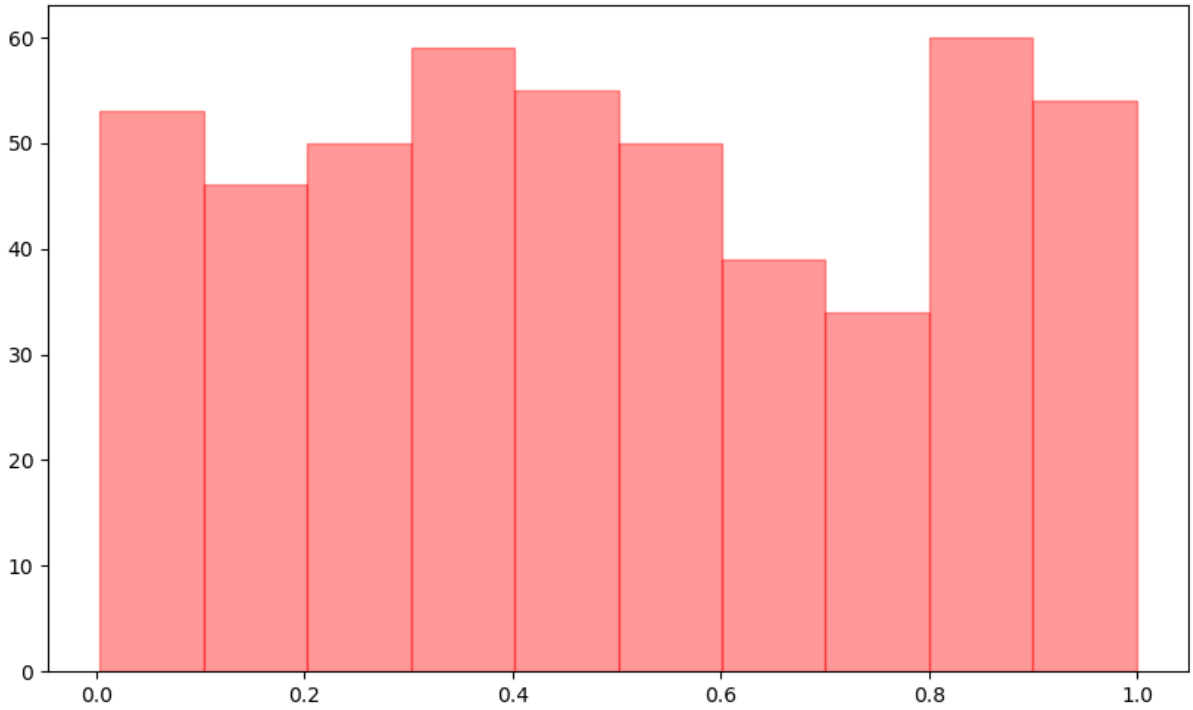


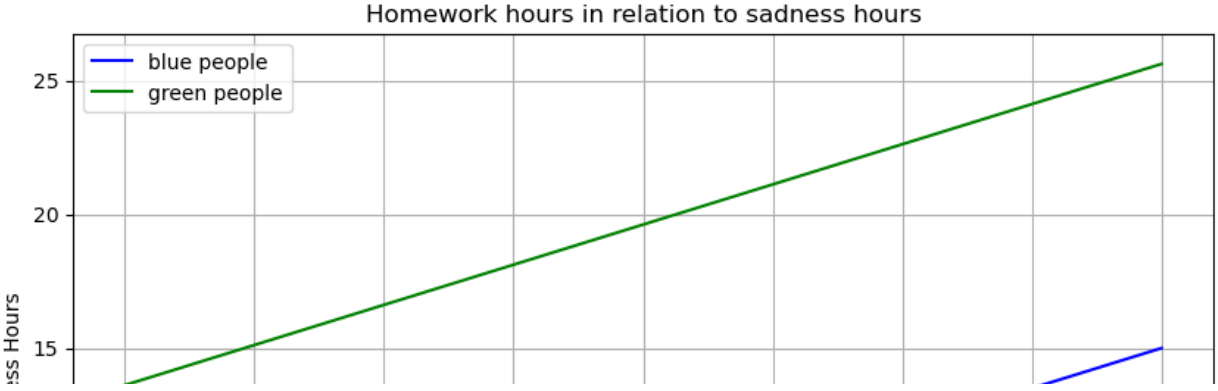
1. Make a figure with 3 subplots, the first containing the plot of the data with a straight line (from 3.), and the second and third containing each of the 2 histograms created in 5. and 6. Try a 3x1 and 1x3 layout and show your favorite.

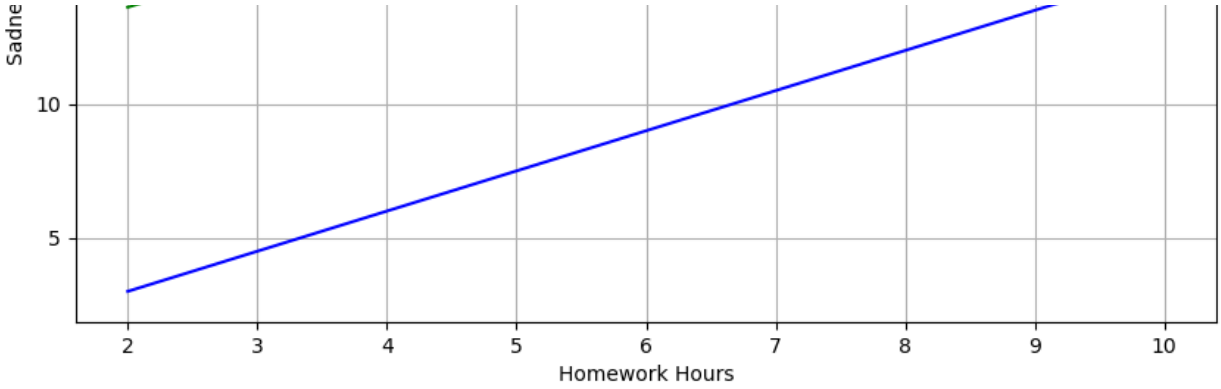
In [90]:
```
fig, axs = plt.subplots(3, 1, figsize=(10, 20))

axs[0].hist(bigdata, color = 'b', edgecolor = 'b', alpha = 0.4)


axs[1].hist(bigdata2, color = 'r', edgecolor = 'r', alpha = 0.4)


axs[2].plot(x, y, 'b', label = 'blue people')
axs[2].plot(b, noised, 'green', label = 'green people')
axs[2].set_title('Homework hours in relation to sadness hours')
axs[2].set_xlabel('Homework Hours')
axs[2].set_ylabel('Sadness Hours')
axs[2].grid()
axs[2].legend()
```

Out[90]:
```
<matplotlib.legend.Legend at 0x1a34d3fe110>
```


50
40
30
20
10
0
0.0
0.2
0.4
0.6
0.8
1.0


60
50
40
30
20
10
0
0.0
0.2
0.4
0.6
0.8
1.0


Homework hours in relation to sadness hours
blue people
green people
25
20
15
ss Hours



In [ ]: