matplotlib homework

In [210…
```
import numpy as np
import matplotlib.pyplot as plt
```

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 [216…
```
x = np.linspace(0, 10) # time in hours
y = 50 + 5*x

plt.plot(x,y)
plt.xlabel('time studied (hours)')
plt.ylabel('grade on test')
plt.title('Hours studied vs. grades on test')
```

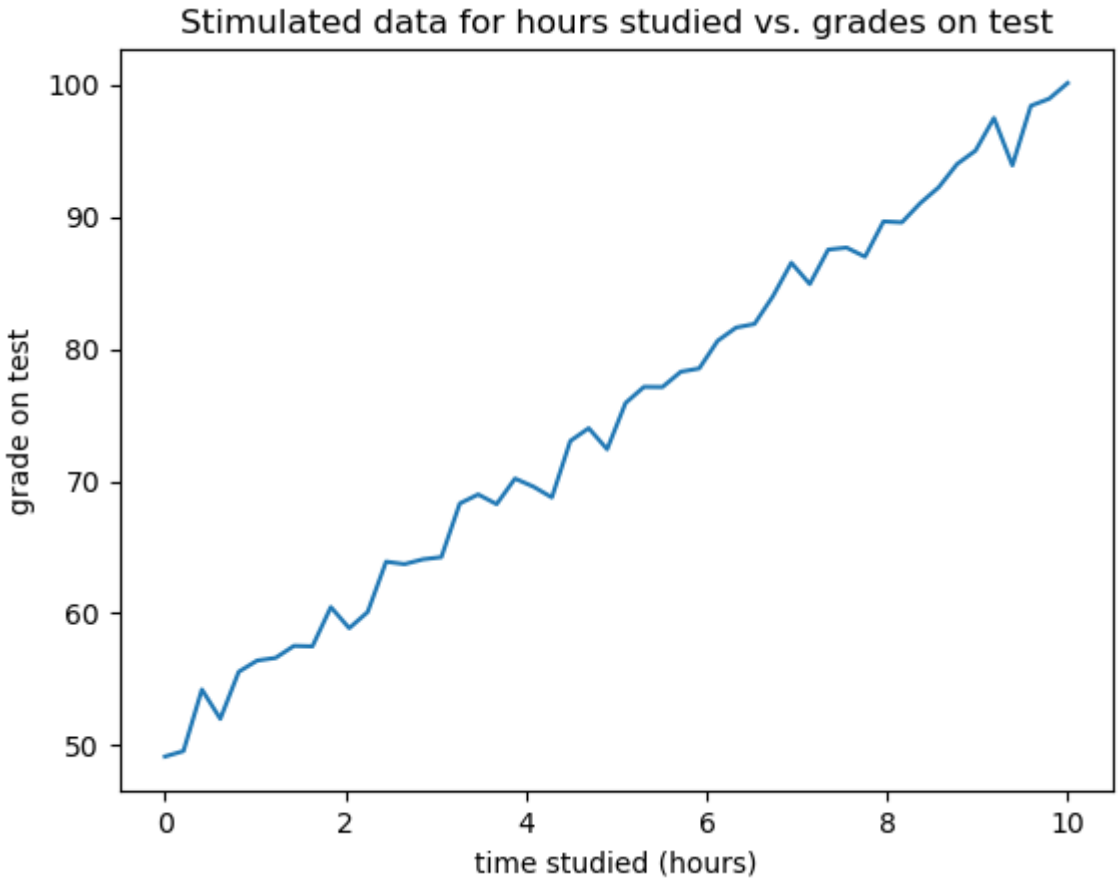
Out[216]:
```
Text(0.5, 1.0, 'Hours studied vs. grades on test')
```

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 [217…
```
x1 = np.linspace(0, 10) # time in hours
y1 = 50 + 5*x

noise = np.random.normal(0,1, size=len(x1))
```

```
y_with_noise = y1 + noise

plt.plot(x1,y_with_noise)
plt.xlabel('time studied (hours)')
plt.ylabel('grade on test')
plt.title('Stimulated data for hours studied vs. grades on test')
```

Out[217]:
```
Text(0.5, 1.0, 'Stimulated data for hours studied vs. grades on test')
```



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 [218…
```
# original line
x = np.linspace(0, 10) # time in hours
y = 50 + 5*x

# simulated line
x1 = np.linspace(0, 10) # time in hours
y1 = 50 + 5*x

noise = np.random.normal(0,1, size=len(x1))
y_with_noise = y1 + noise

# plots
plt.plot(x,y, label = 'original line')
plt.plot(x1,y_with_noise, label = 'stimulated line')
plt.xlabel('time studied (hours)')
plt.ylabel('grade for class')
plt.title('Hours studied vs. grades on test')
plt.legend()
```

Out[218]:
```
<matplotlib.legend.Legend at 0x143bc3490>
```

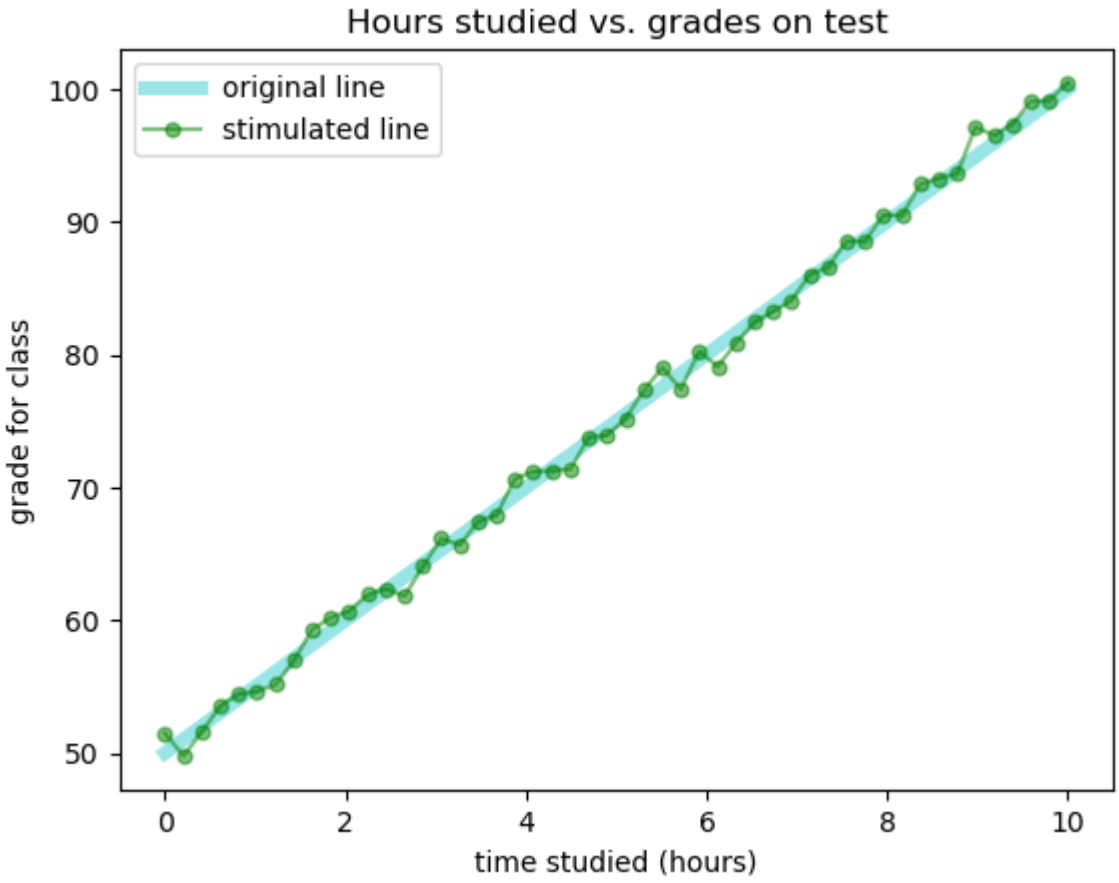
1. Tinker around with your plot (colors, symbols, marker sizes, etc.) until you have a plot you would be happy to use in a presentation.

In [219…

```
# original line
x = np.linspace(0, 10) # time in hours
y = 50 + 5*x

# line that is randomly generated
x1 = np.linspace(0, 10) # time in hours
y1 = 50 + 5*x

noise = np.random.normal(0,1, size=len(x1))
y_with_noise = y1 + noise


# plot the two plots
plt.plot(x,y, 'c-',linewidth = 5, alpha = 0.4, label = 'original line')
plt.plot(x1,y_with_noise, 'go-', alpha = 0.5, markersize = 5, label = 'stimu
plt.xlabel('time studied (hours)')
plt.ylabel('grade for class')
plt.title('Hours studied vs. grades on test')
plt.legend()
```

Out[219]:

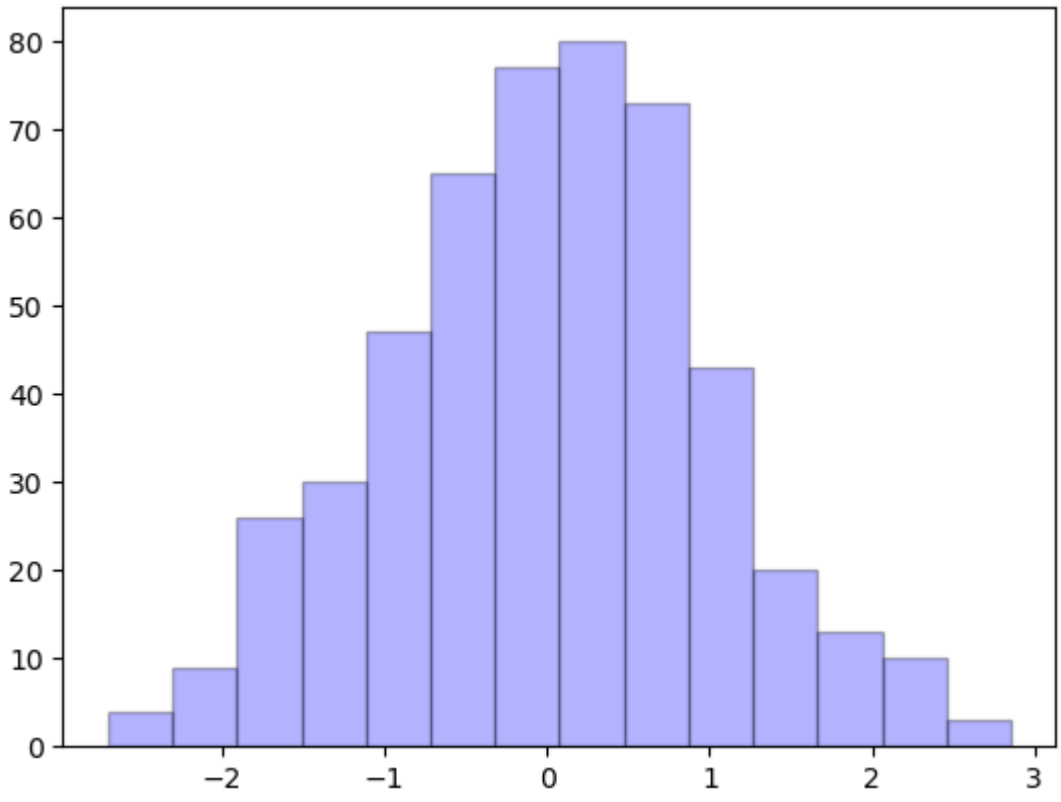
```
<matplotlib.legend.Legend at 0x143c71f90>
```



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

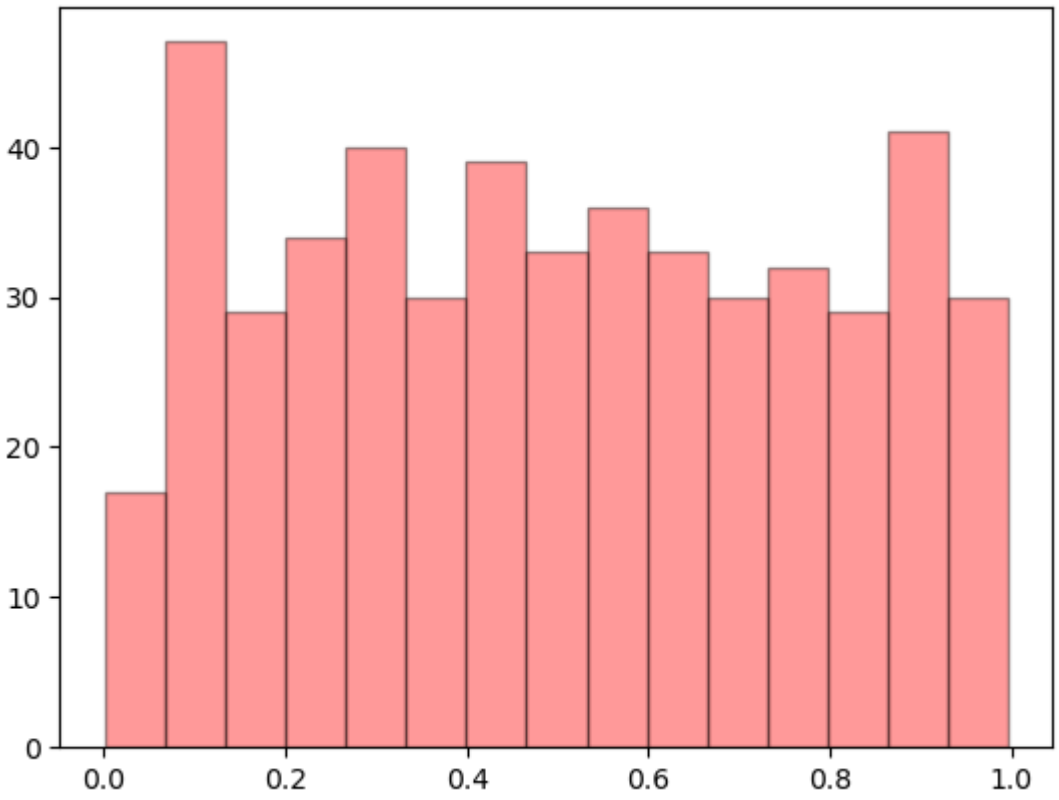
In [220…

```
normal_dist = np.random.normal(0, 1, 500)

plt.hist(normal_dist, bins = 14, color = 'b', edgecolor = 'k', alpha = 0.3);
```



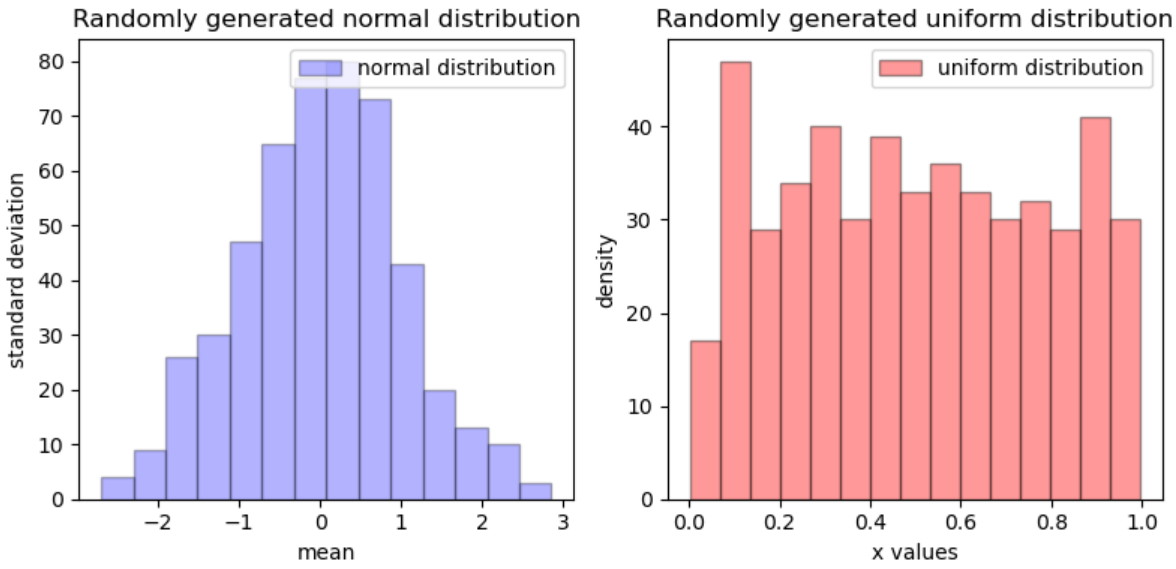
1. Make 500 ***uniformly*** distributed random numbers (use `...rand()` instead of `...randn()`) and make a histogram of them.

In [222…
```
uniform_dist = np.random.rand(500)

plt.hist(uniform_dist, bins = 15, color = 'r', edgecolor = 'k', alpha = 0.4
```



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!

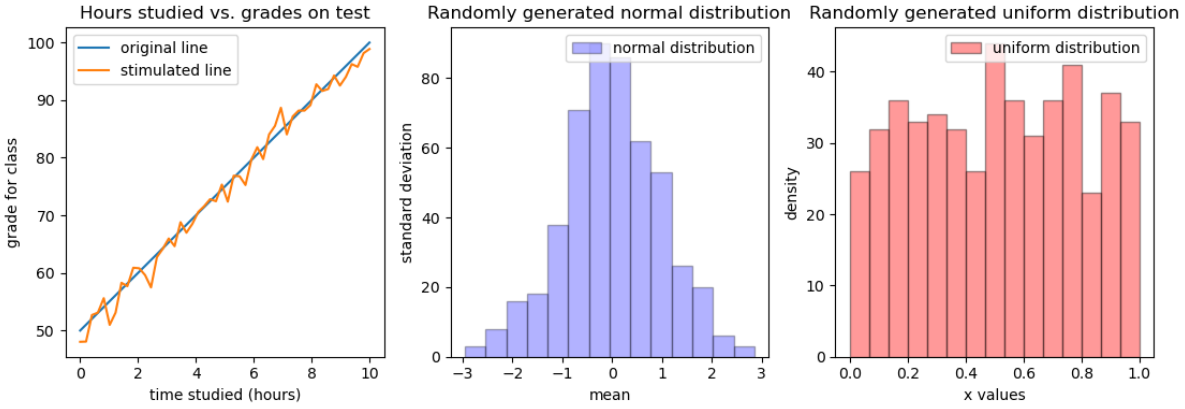
In [226…
```
# normal distribution
fig = plt.figure(figsize = (8,4))
ax = fig.add_subplot(1,2,1)
ax.hist(normal_dist, bins = 14, color = 'b', edgecolor = 'k', alpha = 0.3,
plt.xlabel('mean')
plt.ylabel('standard deviation')
plt.title('Randomly generated normal distribution')
plt.legend()

# uniform distribution
ax1 = fig.add_subplot(1,2,2)
ax1.hist(uniform_dist, bins = 15, color = 'r', edgecolor = 'k', alpha = 0.4
plt.xlabel('x values')
plt.ylabel('density')
plt.title('Randomly generated uniform distribution')
plt.legend()

plt.tight_layout()
```



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.

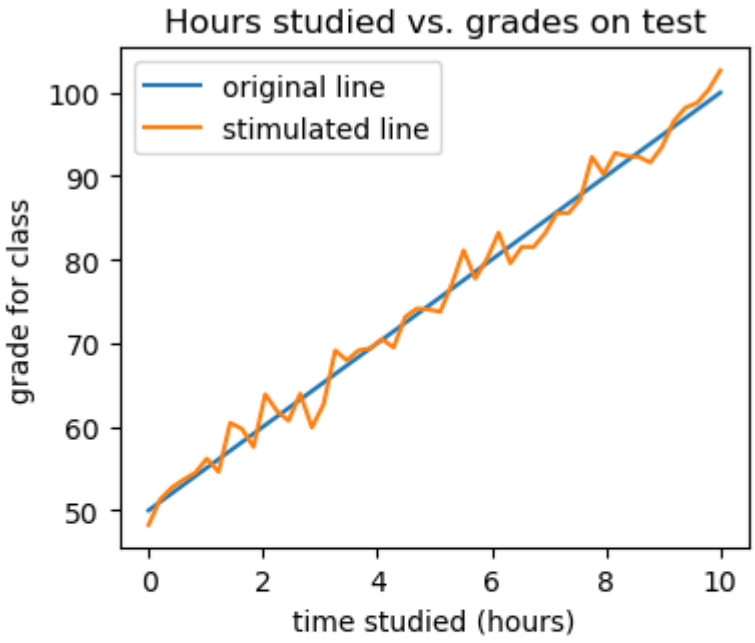
In [201…

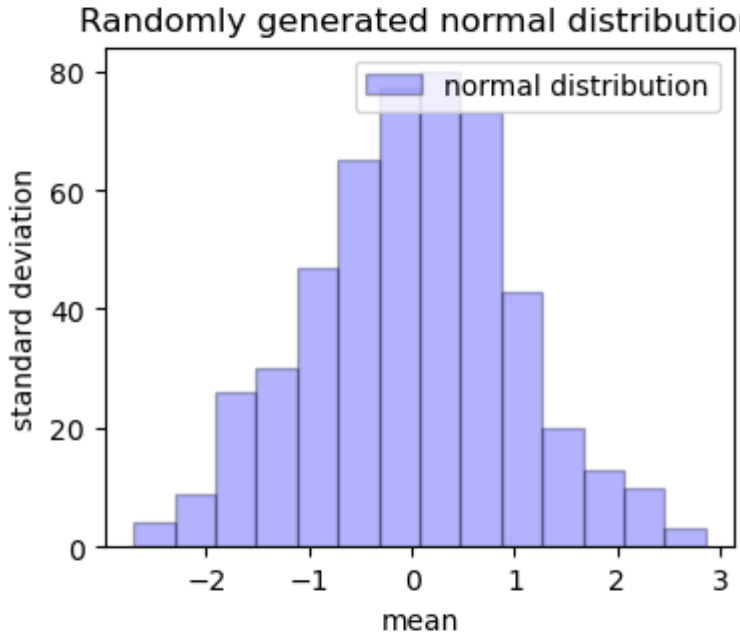
```
# 1x3 layout, my favorite!!

plt.figure(figsize = (11,4))

# straight line
x = np.linspace(0, 10) # time in hours
y = 50 + 5*x

x1 = np.linspace(0, 10) # time in hours
y1 = 50 + 5*x

noise = np.random.normal(scale=2, size=len(x))  # Adjust the scale according
y_with_noise = y1 + noise

plt.subplot(1,3,1)
plt.plot(x,y, label = 'original line')
plt.plot(x1,y_with_noise, label = 'stimulated line')
plt.xlabel('time studied (hours)')
plt.ylabel('grade for class')
plt.title('Hours studied vs. grades on test')
plt.legend()

# normal distribution histogram
plt.subplot(1,3,2)
plt.hist(normal_dist, bins = 14, color = 'b', edgecolor = 'k', alpha = 0.3,
plt.xlabel('mean')
plt.ylabel('standard deviation')
plt.title('Randomly generated normal distribution')
plt.legend()

# uniform distribution histogram
plt.subplot(1,3,3)
plt.hist(uniform_dist, bins = 15, color = 'r', edgecolor = 'k', alpha = 0.4
plt.xlabel('x values')
plt.ylabel('density')
plt.title('Randomly generated uniform distribution')
plt.legend()

plt.tight_layout()
```

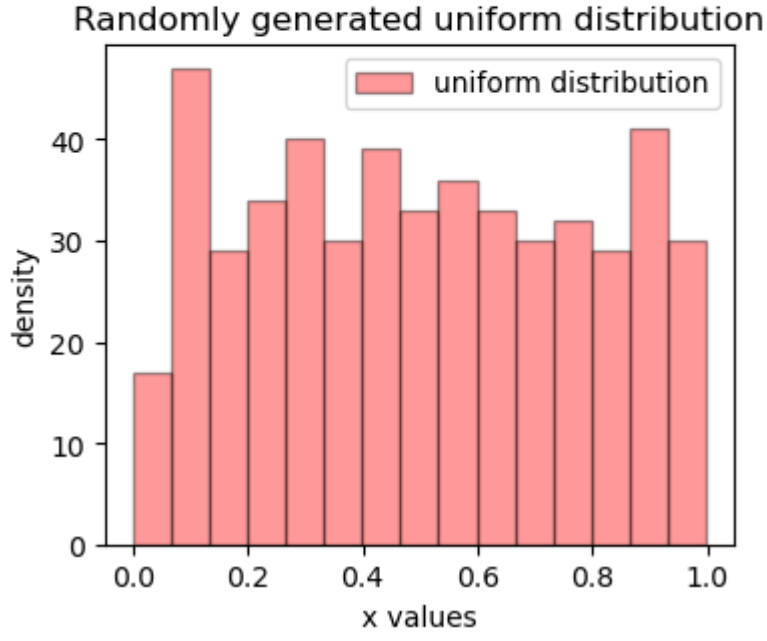


In [229…

```
# 3x1 layout, meh

plt.figure(figsize = (4,10))

# straight line
x = np.linspace(0, 10) # time in hours
y = 50 + 5*x

x1 = np.linspace(0, 10) # time in hours
y1 = 50 + 5*x

noise = np.random.normal(scale=2, size=len(x))  # Adjust the scale according
y_with_noise = y1 + noise

plt.subplot(3,1,1)
plt.plot(x,y, label = 'original line')
plt.plot(x1,y_with_noise, label = 'stimulated line')
plt.xlabel('time studied (hours)')
plt.ylabel('grade for class')
plt.title('Hours studied vs. grades on test')
plt.legend()

# normal distribution histogram
plt.subplot(3,1,2)
plt.hist(normal_dist, bins = 14, color = 'b', edgecolor = 'k', alpha = 0.3,
plt.xlabel('mean')
plt.ylabel('standard deviation')
plt.title('Randomly generated normal distribution')
plt.legend()

# uniform distribution histogram
plt.subplot(3,1,3)
plt.hist(uniform_dist, bins = 15, color = 'r', edgecolor = 'k', alpha = 0.4,
plt.xlabel('x values')
plt.ylabel('density')
plt.title('Randomly generated uniform distribution')
plt.legend()

plt.tight_layout()
```


Hours studied vs. grades on test
original line
stimulated line
grade for class
time studied (hours)
100
90
80
70
60
50
0
2
4
6
8
10


Randomly generated normal distribution
normal distribution
standard deviation
mean
80
60
40
20
0
−2
−1
0
1
2
3


Randomly generated uniform distribution
uniform distribution
density
x values
40
30
20
10
0
0.0
0.2
0.4
0.6
0.8
1.0