

# tu08\_re\_MatPlotLib\_HW

February 23, 2023

## 1 matplotlib homework

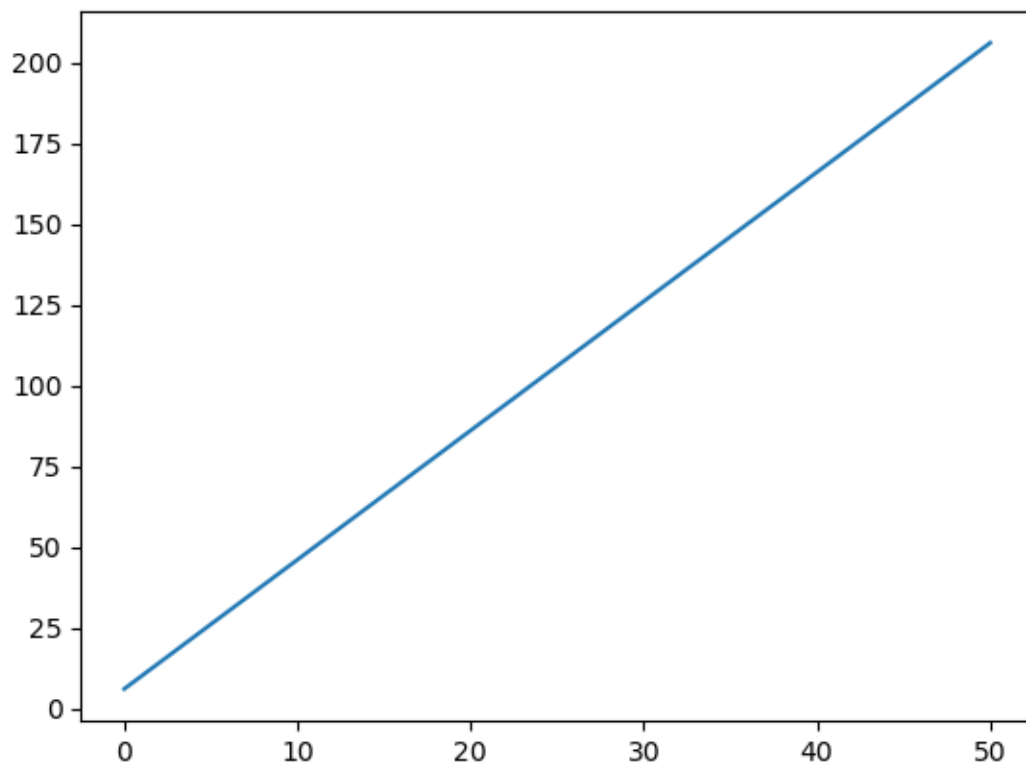
1. Make a plot of a straight line. Use `linspace()` to create the  $x$  values and the formula of a straight line,  $y = a + bx$ , to create the  $y^*$  values (use an  $a$  and  $b$  of your choosing). You can pretend  $x$  and  $y$  are anything you like ( $x$  = time,  $y$  = international piracy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

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

```
[13]: x = np.linspace(0,50,25)
y = 4*x + 6

plt.plot(x,y)
```

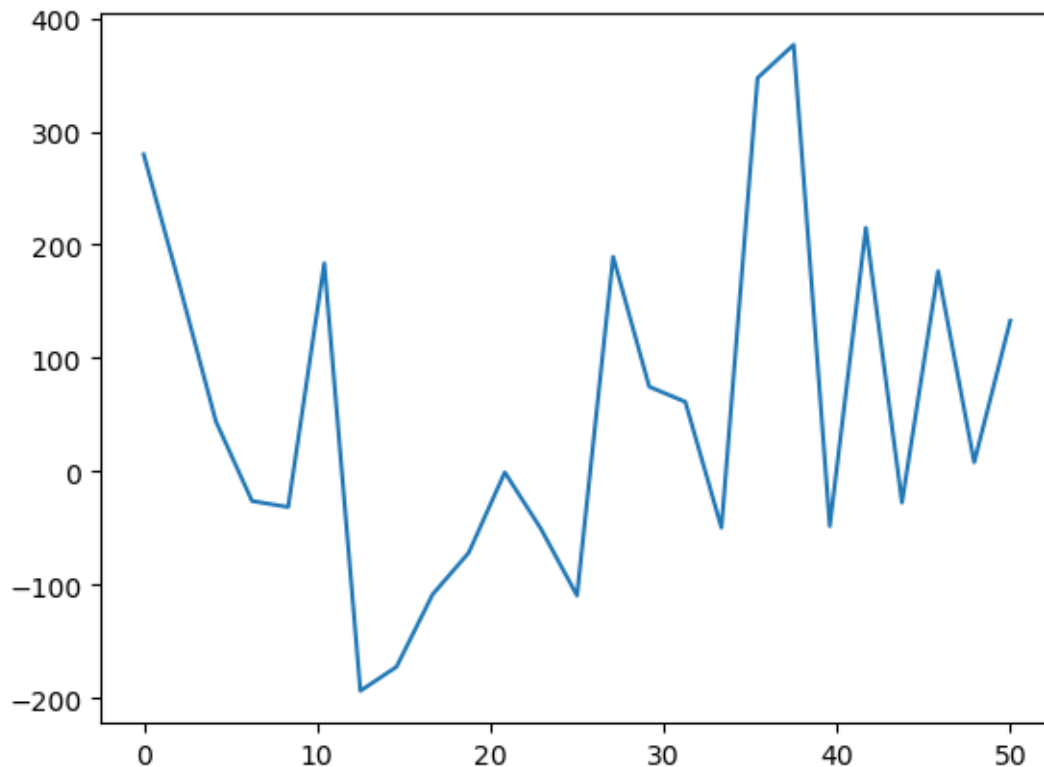
```
[13]: [<matplotlib.lines.Line2D at 0x7fecb111b5e0>]
```



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

```
[20]: y2 = np.random.randn(len(x))*150  
plt.plot(x,y2)
```

```
[20]: [<matplotlib.lines.Line2D at 0x7fecb14b15e0>]
```

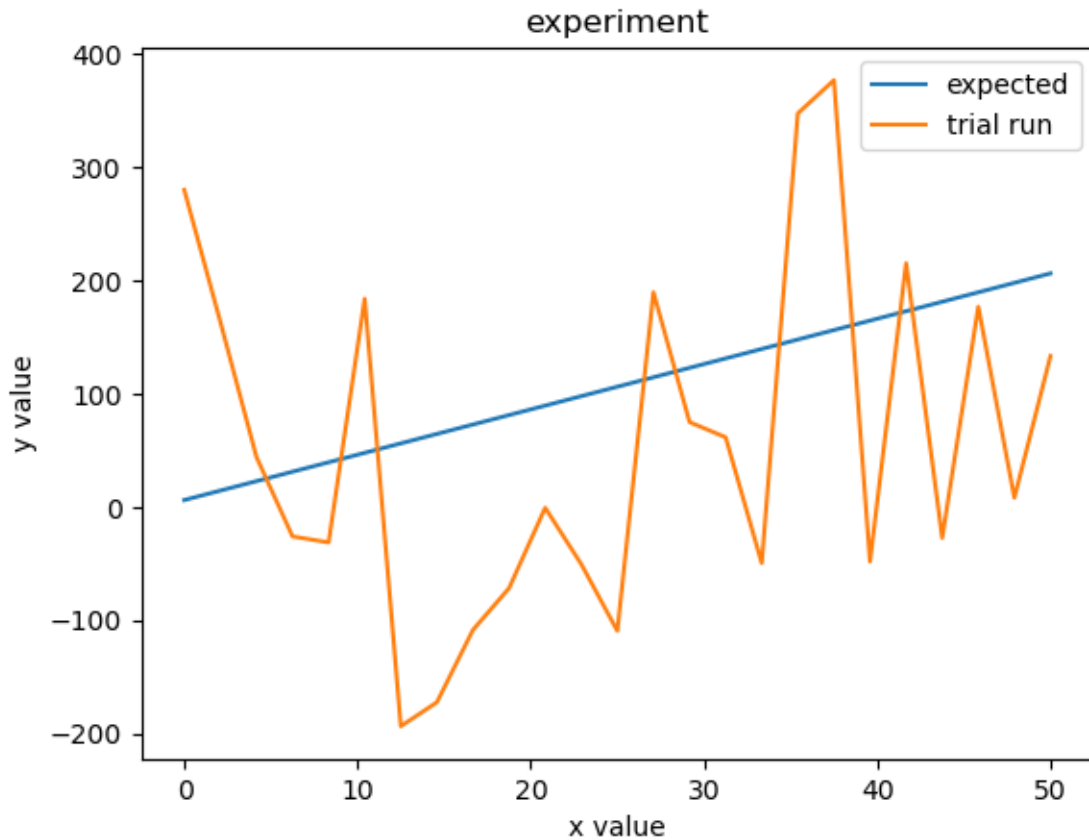


3. Plot the straight line from 1. and the data from 2. on the same graph. Make sure to add the standard annotations, including a legend.

```
[104]: # plots  
plt.plot(x,y)  
plt.plot(x,y2)  
  
# labels  
plt.xlabel('x value')  
plt.ylabel('y value')  
plt.title('experiment')
```

```
# legend
plt.legend(['expected', 'trial run'])
```

[104]: <matplotlib.legend.Legend at 0x7fecb43f1c40>



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.

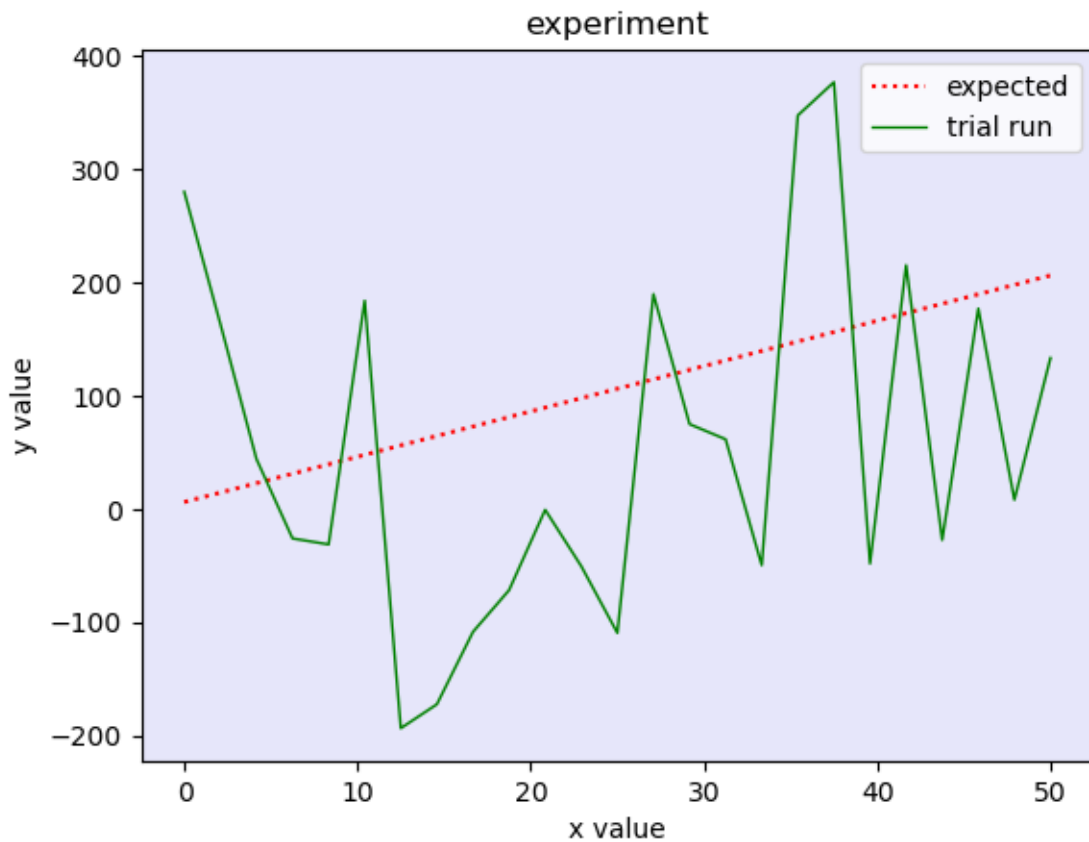
```
[56]: ax = plt.axes()
ax.set_facecolor('lavender')

# plots
plt.plot(x,y, 'r:', linewidth = 1.5)
plt.plot(x,y2, 'g', linewidth = 1)

# labels
plt.xlabel('x value')
plt.ylabel('y value')
plt.title('experiment')
```

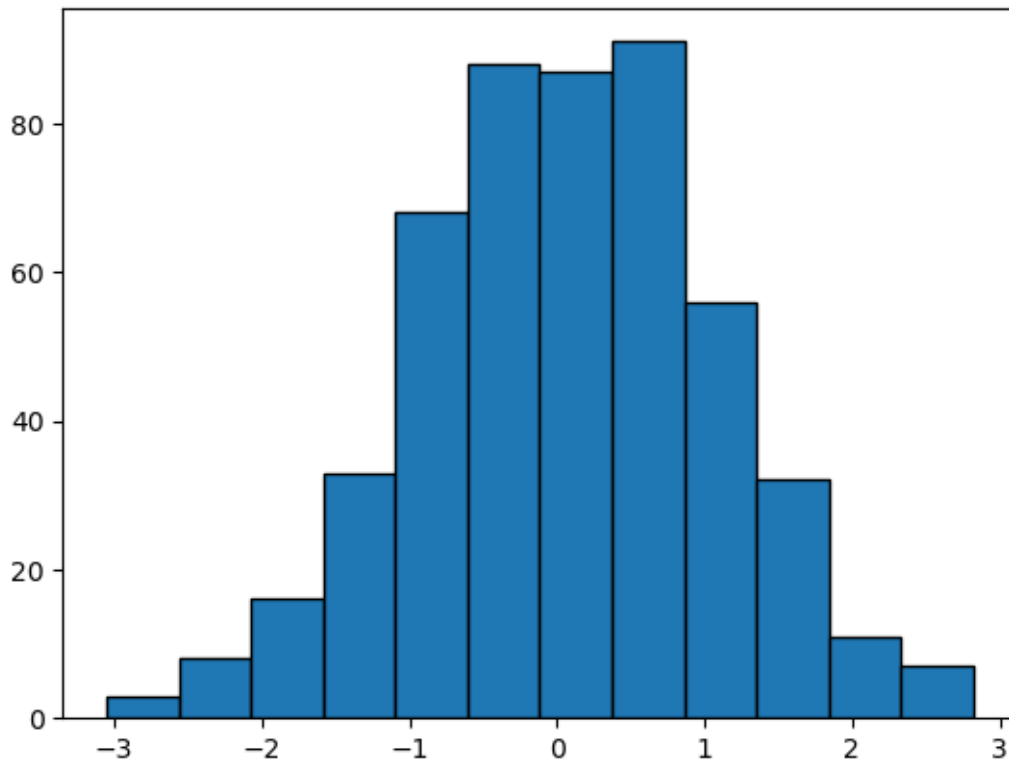
```
# legend
plt.legend(['expected', 'trial run'])
```

[56]: <matplotlib.legend.Legend at 0x7fec80906f40>



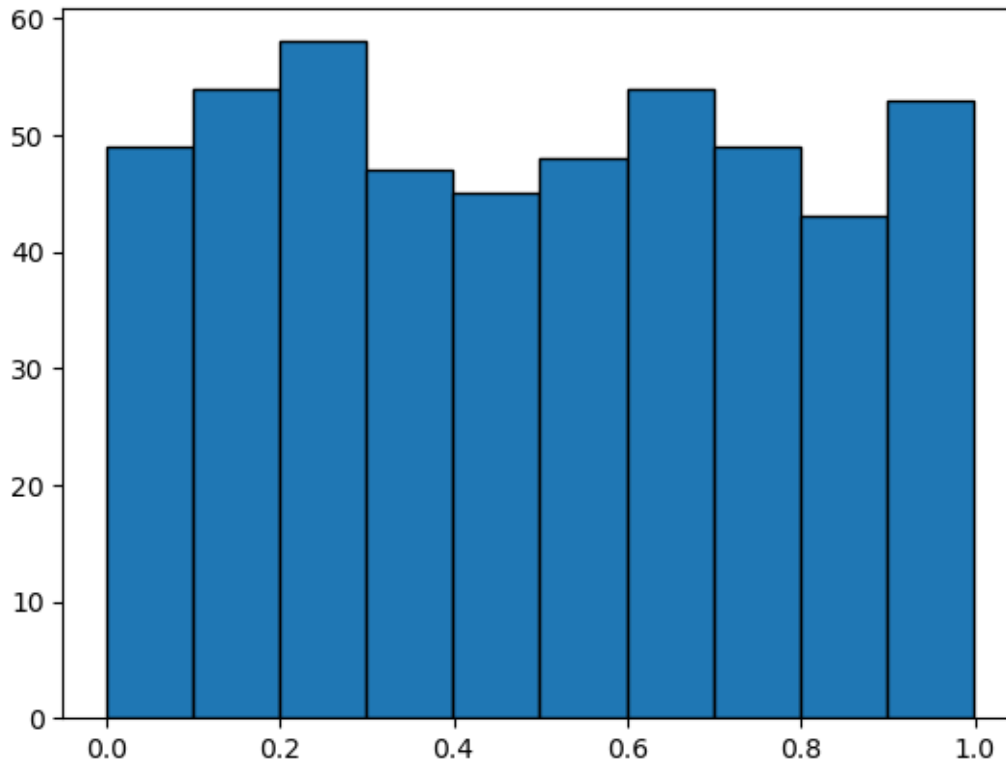
5. Make 500 *normally* distributed random numbers and make a histogram of them.

```
[93]: norm = np.random.randn(500)
plt.hist(norm, bins = 12, edgecolor = 'k');
```



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

```
[92]: uni = np.random.rand(500)
      plt.hist(uni, bins = 10, edgecolor = 'k');
```

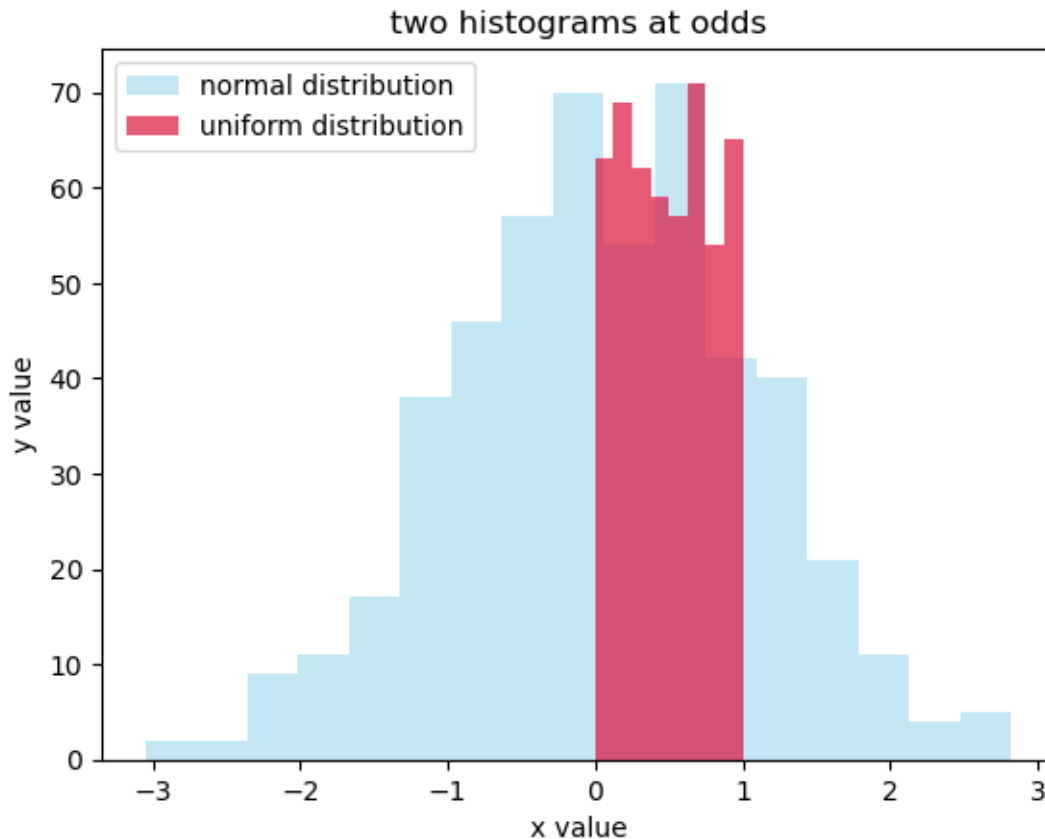


7. Plot the histograms from 5. and 6. in the same axes to compare the two distributions. Tinker around with the `color =` and `alpha =` arguments to `plt.hist()` until you're happy with your figure. Don't forget the axis labels and a legend!

```
[115]: plt.hist(norm, bins = 17, color = 'skyblue', alpha = 0.5, label = 'normal_
      ↪distribution')
plt.hist(uni, bins = 8, color = 'crimson', alpha = 0.7, label = 'uniform_
      ↪distribution')

# labels
plt.title('two histograms at odds')
plt.xlabel('x value')
plt.ylabel('y value')
plt.legend()
```

```
[115]: <matplotlib.legend.Legend at 0x7feca4dc3490>
```



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.

```
[126]: plt.subplot(3,1,1)
plt.hist(norm, bins = 17, color = 'skyblue', alpha = 0.5, edgecolor = 'grey',
        label = 'normal distribution');

plt.subplot(3,1,3)
plt.hist(uni, bins = 8, color = 'crimson', alpha = 0.7, edgecolor = 'k', label=
        label = 'uniform distribution');

plt.subplot(3,1,2)
plt.plot(x,y, 'g--', linewidth = 1);
```

