

# Plotting with matplotlib

```
In [2]: import numpy as np  
import matplotlib.pyplot as plt
```

```
In [3]: t = np.arange(0,20,1)
t
```

```
Out[3]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
              17, 18, 19])
```

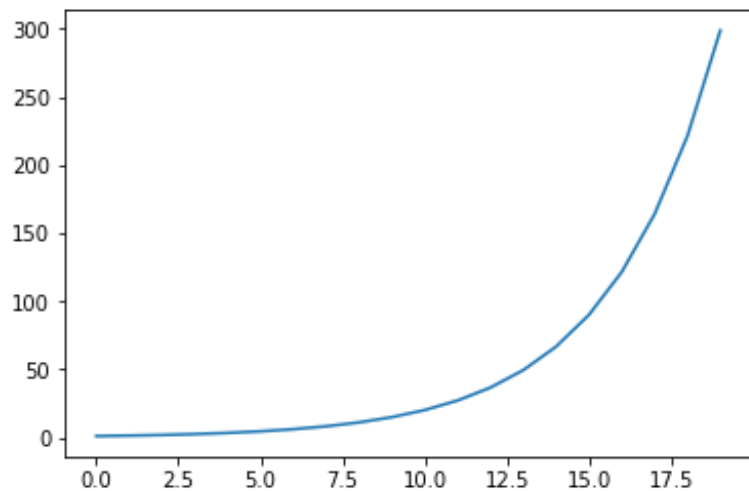
```
In [4]: a = 0.3
y = np.exp(a*t)

print('y: ',y)
```

```
y:  [  1.          1.34985881  1.8221188   2.45960311  3.32011692
      4.48168907  6.04964746  8.16616991 11.02317638 14.87973172
     20.08553692 27.11263892 36.59823444 49.40244911 66.68633104
     90.0171313 121.51041752 164.0219073  221.4064162 298.86740097]
```

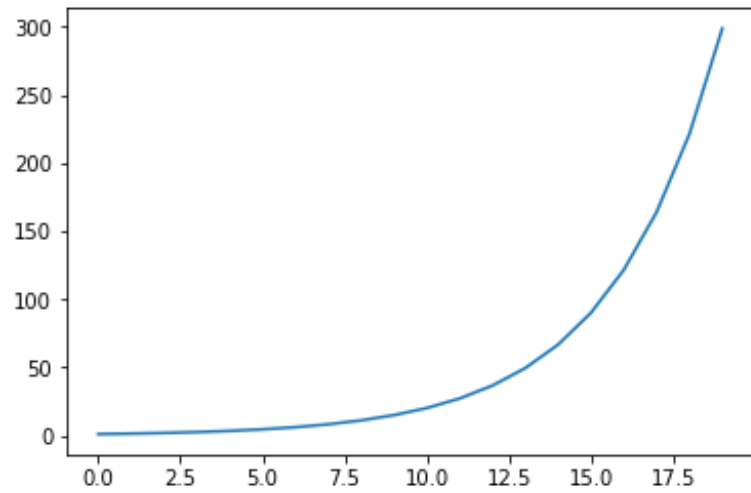
```
In [5]: plt.plot(y)
```

```
Out[5]: [<matplotlib.lines.Line2D at 0x11c0f0748>]
```

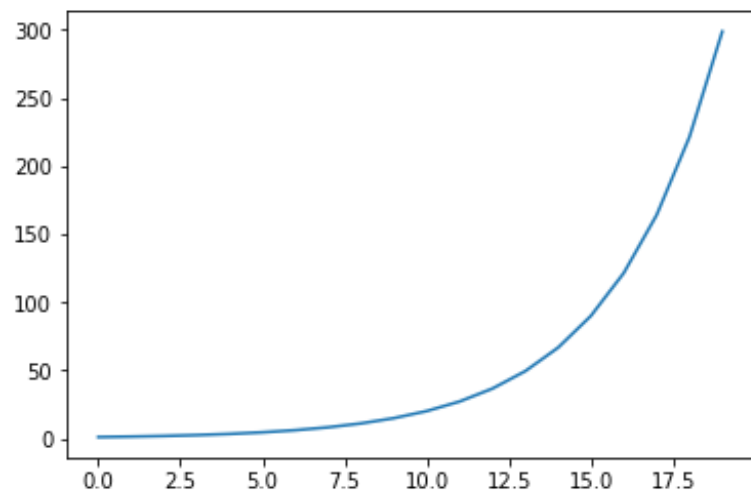


```
In [6]: plt.plot(t,y)
```

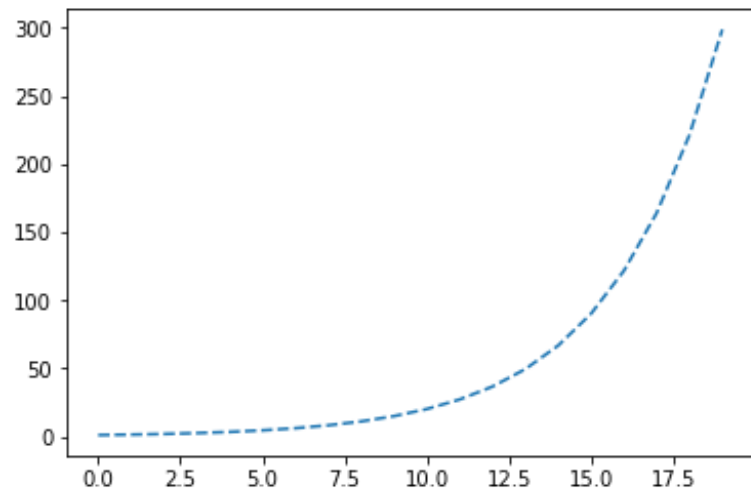
```
Out[6]: [<matplotlib.lines.Line2D at 0x11c220b00>]
```



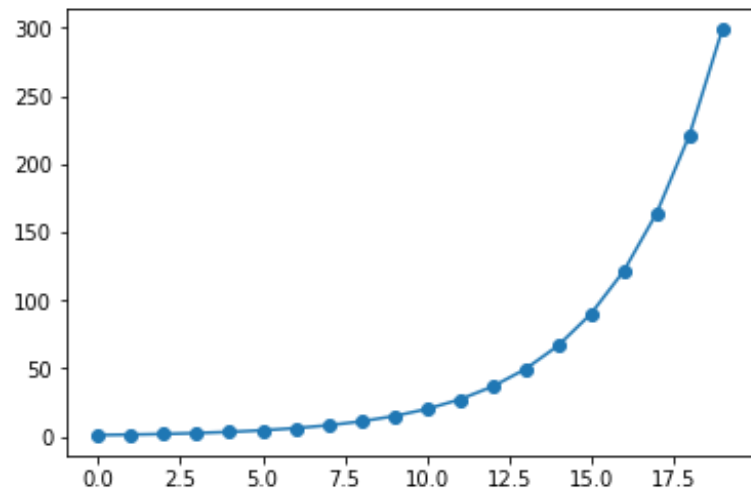
```
In [7]: plt.plot(t,y)  
plt.show()
```



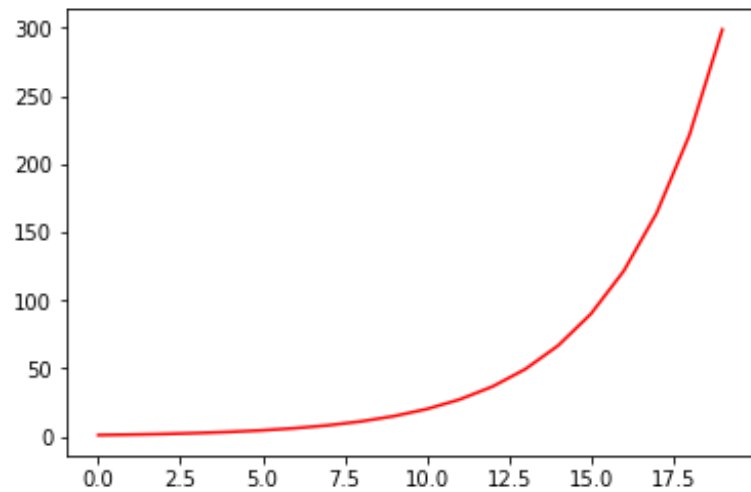
```
In [8]: plt.plot(t,y,'--')  
plt.show()
```



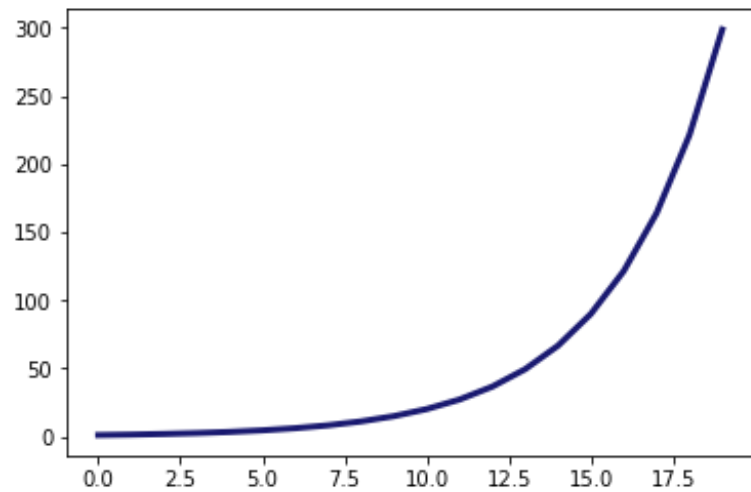
```
In [9]: plt.plot(t,y,'o-')  
plt.show()
```



```
In [10]: plt.plot(t,y,'-',color='r')  
plt.show()
```

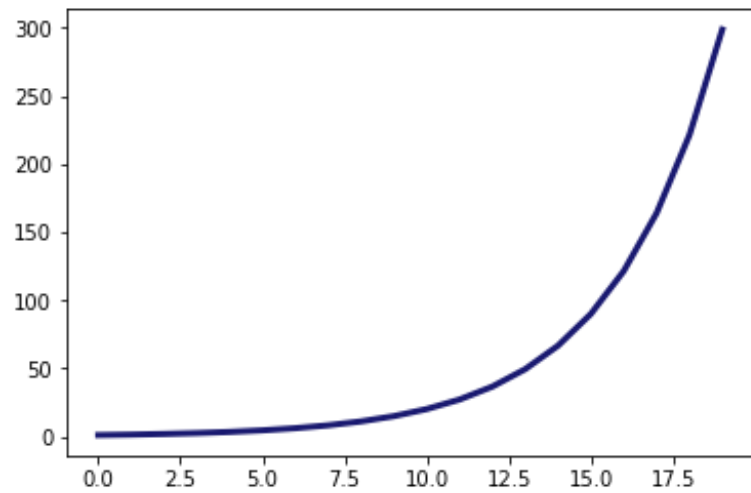


```
In [11]: plt.plot(t,y,'-',color='midnightblue', linewidth=3)  
plt.show()
```



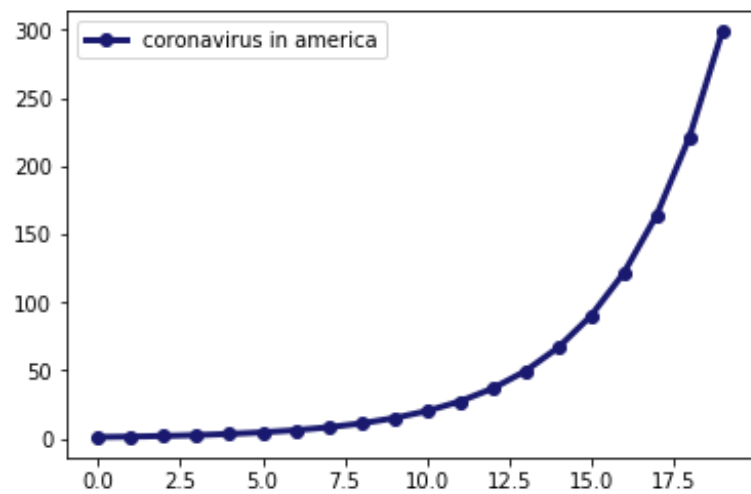


```
In [12]: plt.plot(t,y,'-',color='midnightblue', linewidth=3,label='coronavirus')  
plt.show()
```



no legend has appeared?

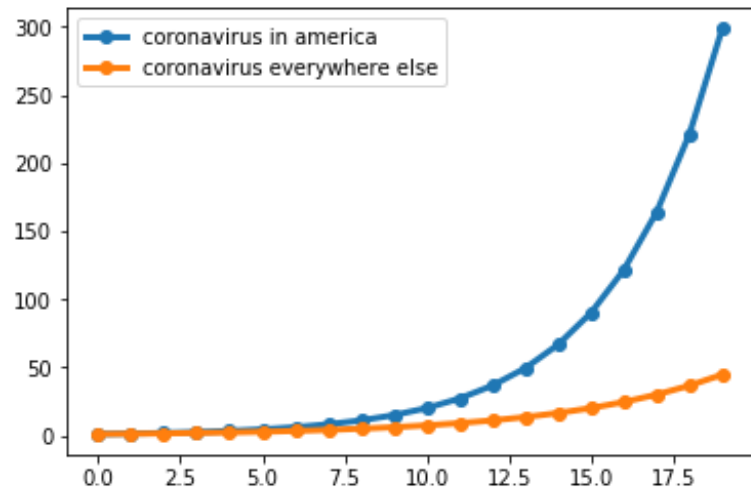
```
In [13]: plt.plot(t,y,'o-',color='midnightblue', linewidth=3,label='coronavirus in america')
plt.legend()
plt.show()
```



```
In [14]: a = 0.3
b = 0.2
y = np.exp(a*t)
y1 = np.exp(b*t)

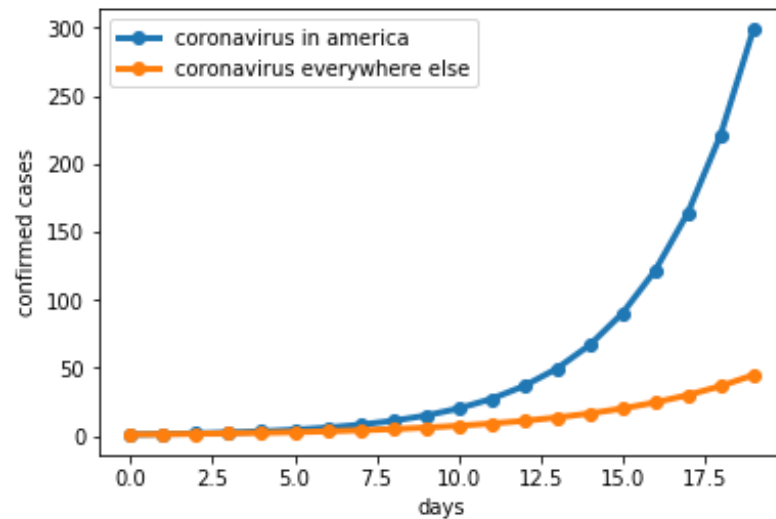
plt.plot(t,y,'o-', linewidth=3,label='coronavirus in america')
plt.plot(t,y1,'o-', linewidth=3,label='coronavirus everywhere else')

plt.legend()
plt.show()
```



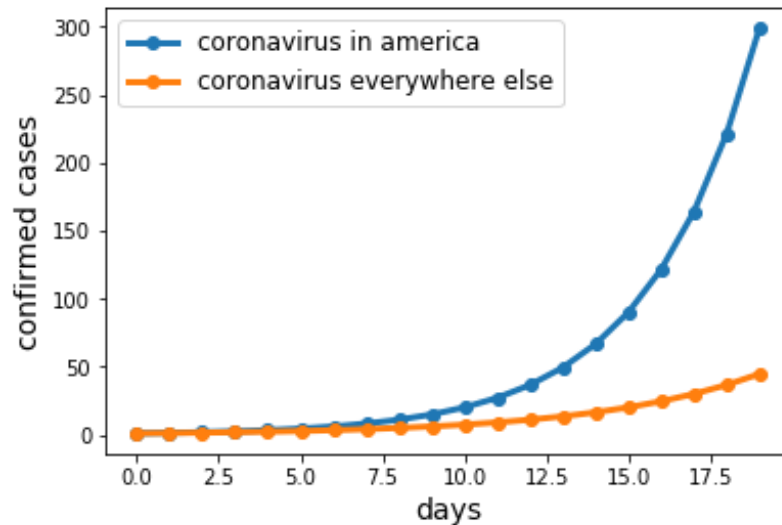
```
In [15]: plt.plot(t,y,'o-', linewidth=3,label='coronavirus in america')
plt.plot(t,y1,'o-', linewidth=3,label='coronavirus everywhere else')

plt.xlabel('days') # include xlabel
plt.ylabel('confirmed cases') # include ylabel
plt.legend()
plt.show()
```



```
In [16]: plt.plot(t,y,'o-', linewidth=3,label='coronavirus in america')
plt.plot(t,y1,'o-', linewidth=3,label='coronavirus everywhere else')

plt.xlabel('days',fontsize=14) # include font size
plt.ylabel('confirmed cases',fontsize=14)
plt.legend(fontsize=12)
plt.show()
```

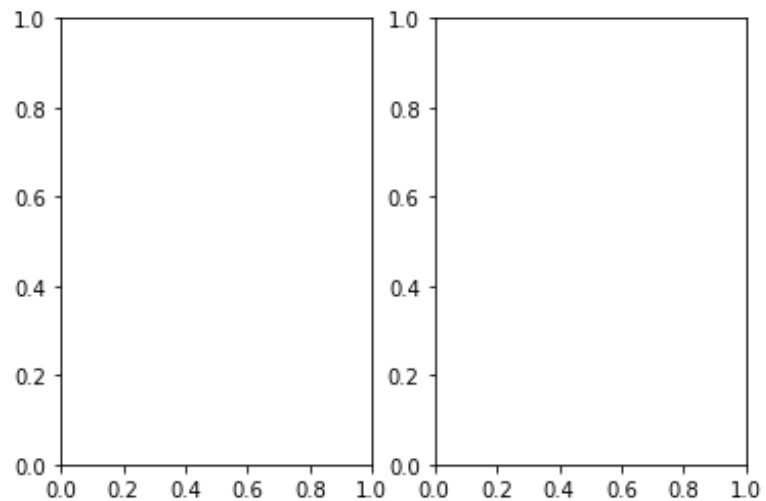


# Figures and Axes

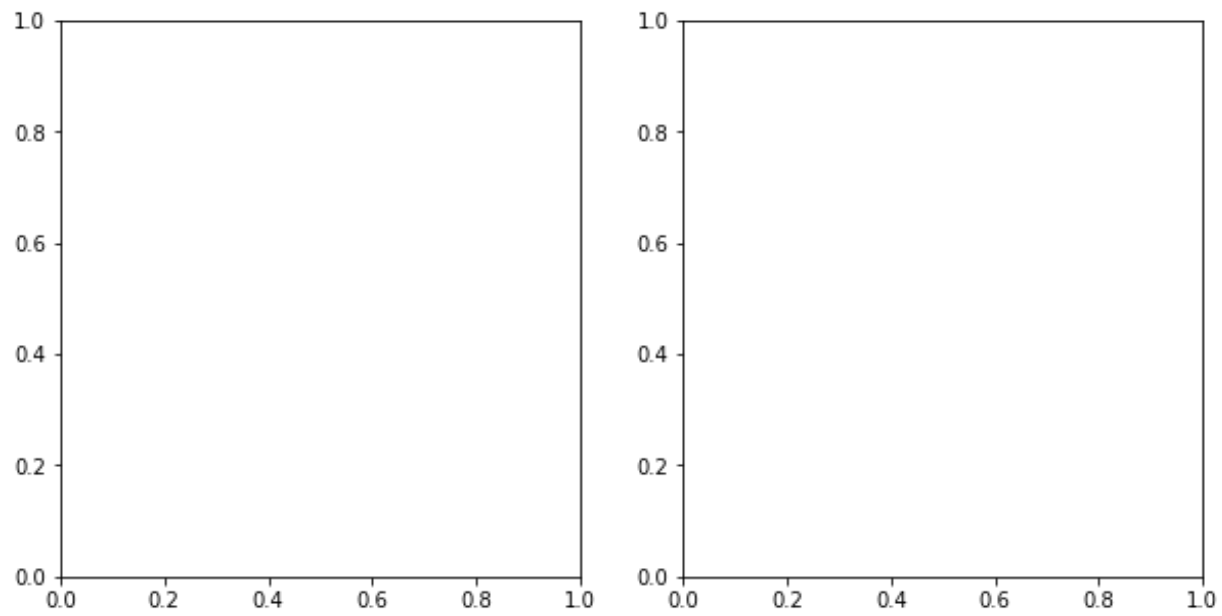
```
In [17]: fig = plt.figure()
```

```
<Figure size 432x288 with 0 Axes>
```

```
In [18]: fig, ax = plt.subplots(1, 2)
```



```
In [19]: fig, ax = plt.subplots(1, 2, figsize=(10, 5))
```



In [20]:

```
ax
```

Out[20]: array([<matplotlib.axes.\_subplots.AxesSubplot object at 0x11c3207b8>,  
                  <matplotlib.axes.\_subplots.AxesSubplot object at 0x11c8502b0>],  
              dtype=object)



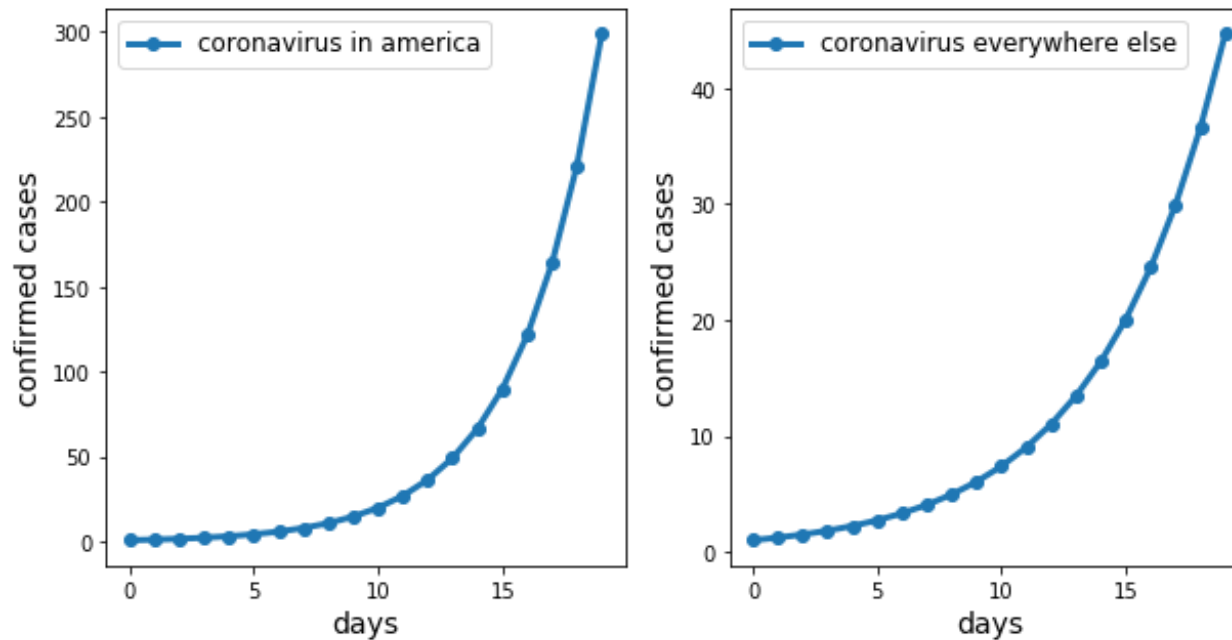
```

In [21]: fig,ax = plt.subplots(1,2,figsize=(10,5))

ax[0].plot(t,y,'o-', linewidth=3,label='coronavirus in america')
ax[1].plot(t,y1,'o-', linewidth=3,label='coronavirus everywhere else')

for a in ax:
    a.set_xlabel('days',fontsize=14) # include font size
    a.set_ylabel('confirmed cases',fontsize=14)
    a.legend(fontsize=12)
plt.show()

```



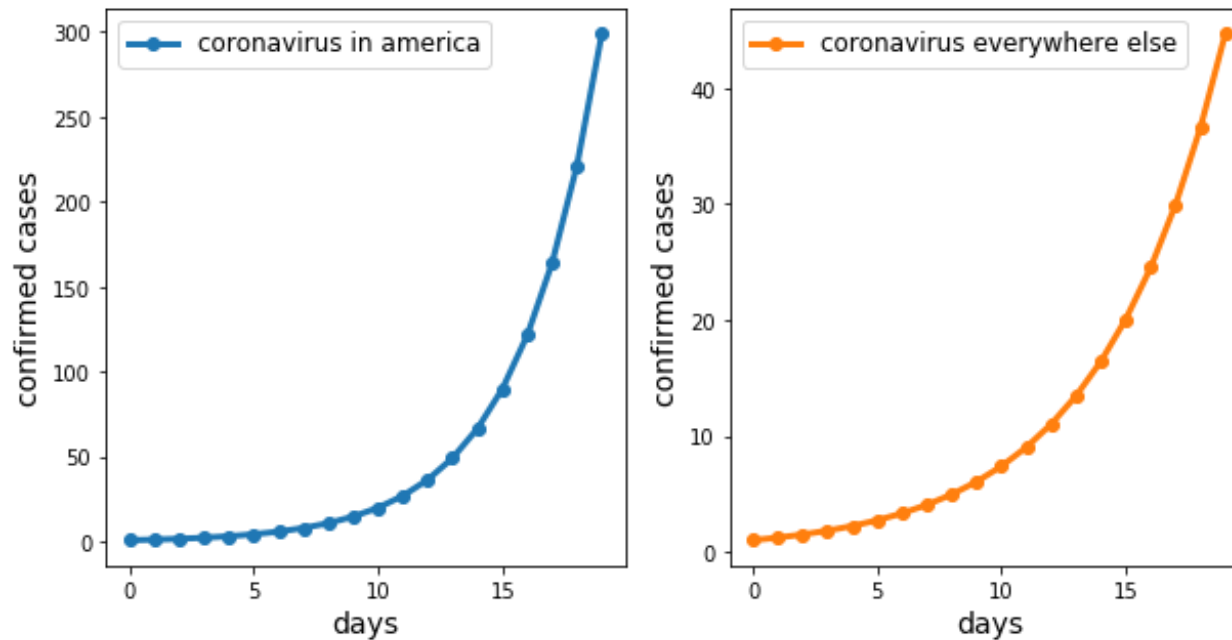
```

In [22]: fig,ax = plt.subplots(1,2,figsize=(10,5))

ax[0].plot(t,y,'o-',color='C0', linewidth=3,label='coronavirus in america')
ax[1].plot(t,y1,'o-', color='C1',linewidth=3,label='coronavirus everywhere else')

for a in ax:
    a.set_xlabel('days',fontsize=14) # include font size
    a.set_ylabel('confirmed cases',fontsize=14)
    a.legend(fontsize=12)
plt.show()

```



```

In [23]: fig,ax = plt.subplots(1,2,figsize=(10,5))

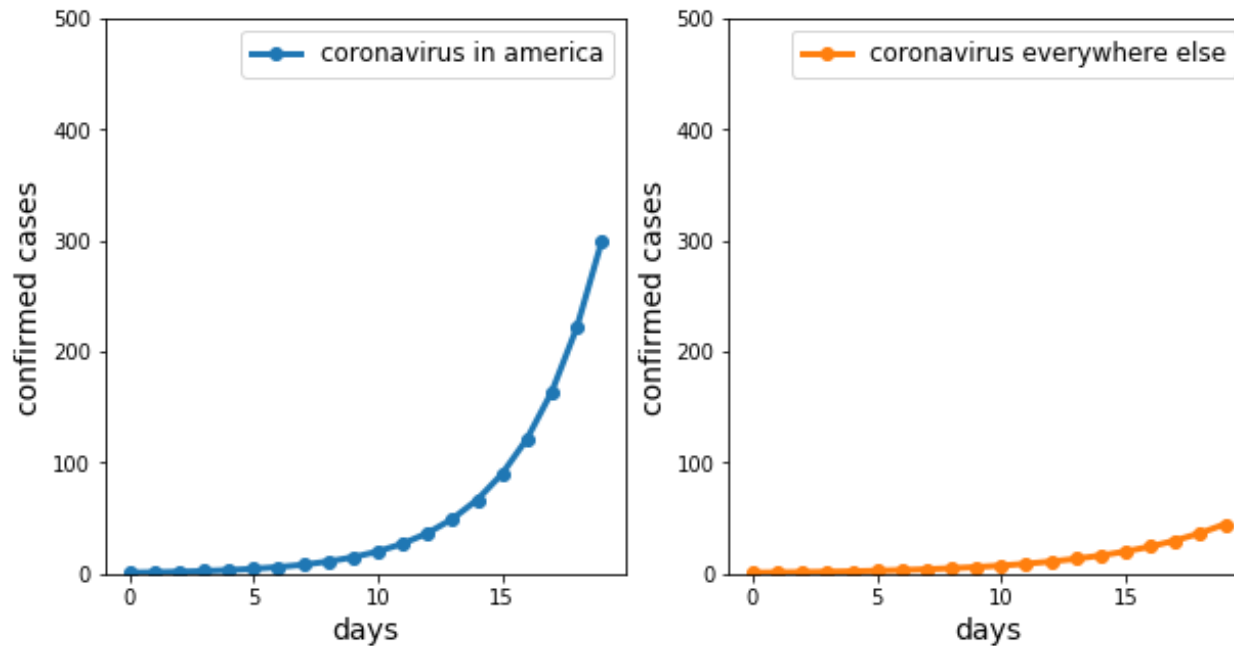
ax[0].plot(t,y,'o-',color='C0', linewidth=3,label='coronavirus in america')
ax[1].plot(t,y1,'o-', color='C1',linewidth=3,label='coronavirus everywhere else')

for a in ax:

    a.set_ylim(0,500) # constrain the y axis limits

    a.set_xlabel('days',fontsize=14)
    a.set_ylabel('confirmed cases',fontsize=14)
    a.legend(fontsize=12)
plt.show()

```

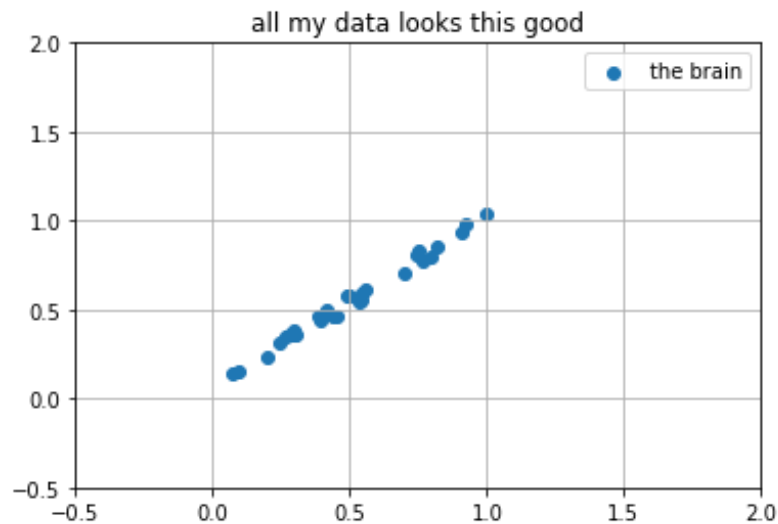


# scatter plots

```
In [24]: x = np.random.rand(30)
m = x + 0.1*np.random.rand(30)
plt.scatter(x,m)

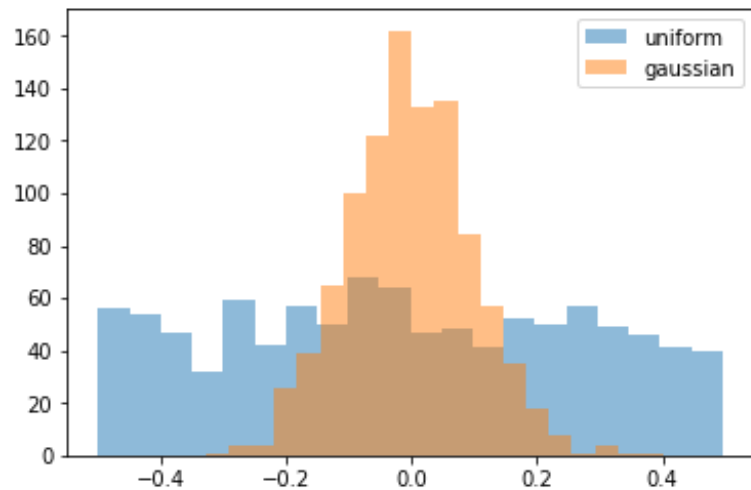
plt.xlim(-.5,2)
plt.ylim(-.5,2)

plt.grid(which='major')
plt.legend(['the brain'])
plt.title('all my data looks this good')
plt.show()
```

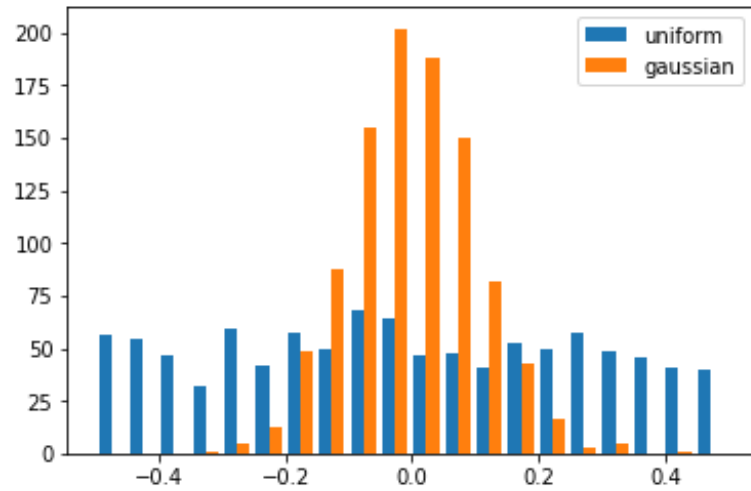


# Histograms

```
In [25]: x_a = np.random.rand(1000)-0.5  
x_b = np.random.randn(1000)*0.1  
  
plt.hist(x_a, bins=20, alpha=0.5, label='uniform')  
plt.hist(x_b, bins=20, alpha=0.5, label='gaussian')  
plt.legend()  
plt.show()
```



```
In [26]: plt.hist([x_a,x_b],bins=20,label=['uniform','gaussian'])  
plt.legend()  
plt.show()
```



**bar plots**

```

In [27]: labels = ['Carolina', 'Black-Capped', 'Boreal', 'Grey-Headed', 'Mountain']
male_means = [20, 34, 30, 35, 27]
female_means = [25, 32, 34, 20, 25]

x = np.arange(len(labels)) # the label locations
width = 0.35 # the width of the bars

fig, ax = plt.subplots(figsize=(10,5))
rects1 = ax.bar(x - width/2, male_means, width, label='Male Chickadees')
rects2 = ax.bar(x + width/2, female_means, width, label='Female Chickadees')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Scores')
ax.set_title('Awesomeness Scores by Group and Gender')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()

def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')

autolabel(rects1)
autolabel(rects2)

fig.tight_layout()

```

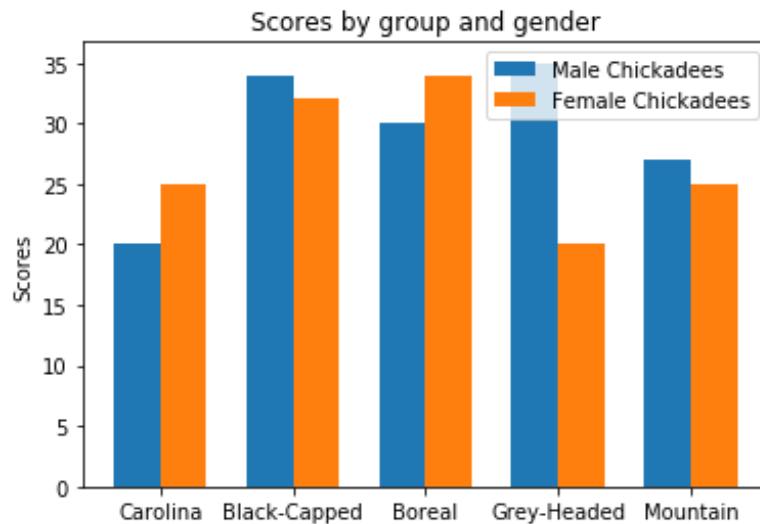


```
In [28]: labels = ['Carolina', 'Black-Capped', 'Boreal', 'Grey-Headed', 'Mountain']  
male_means = [20, 34, 30, 35, 27]  
female_means = [25, 32, 34, 20, 25]  
  
x = np.arange(len(labels)) # the label locations  
width = 0.35 # the width of the bars
```

```
In [29]: fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, male_means, width, label='Male Chickadees')
rects2 = ax.bar(x + width/2, female_means, width, label='Female Chickadees')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Scores')
ax.set_title('Scores by group and gender')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()
```

Out[29]: <matplotlib.legend.Legend at 0x11c53f4a8>



```
In [30]: def autolabel(rects):  
        """Attach a text label above each bar in *rects*, displaying its height."""  
        for rect in rects:  
            height = rect.get_height()  
            ax.annotate('{}' .format(height),  
                        xy=(rect.get_x() + rect.get_width() / 2, height),  
                        xytext=(0, 3), # 3 points vertical offset  
                        textcoords="offset points",  
                        ha='center', va='bottom')
```

```

In [31]: labels = ['Carolina', 'Black-Capped', 'Boreal', 'Grey-Headed', 'Mountain']
male_means = [20, 34, 30, 35, 27]
female_means = [25, 32, 34, 20, 25]

x = np.arange(len(labels)) # the label locations
width = 0.35 # the width of the bars

fig, ax = plt.subplots(figsize=(10,5))
rects1 = ax.bar(x - width/2, male_means, width, label='Male Chickadees')
rects2 = ax.bar(x + width/2, female_means, width, label='Female Chickadees')

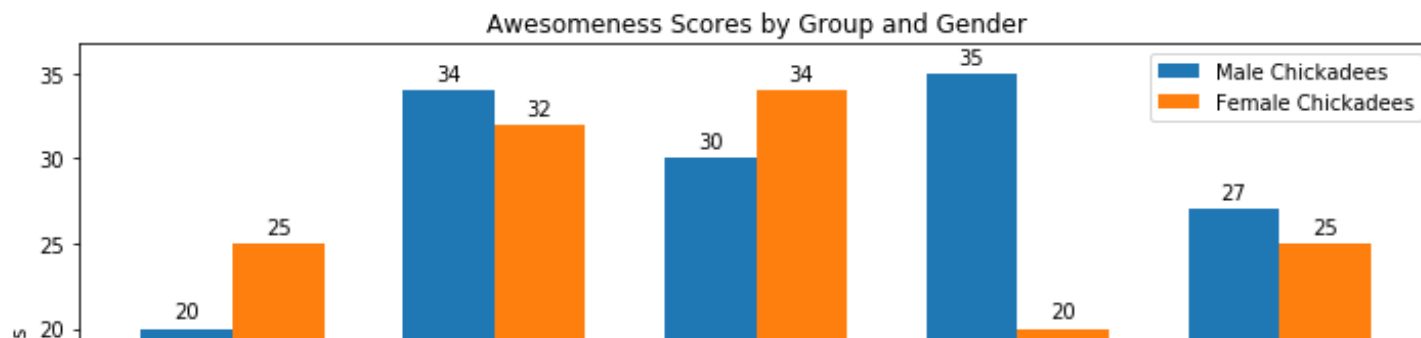
# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Scores')
ax.set_title('Awesomeness Scores by Group and Gender')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()

autolabel(rects1)
autolabel(rects2)

fig.tight_layout()

plt.show()

```



# Images

```
In [32]: import sklearn.datasets  
digits = sklearn.datasets.load_digits()
```

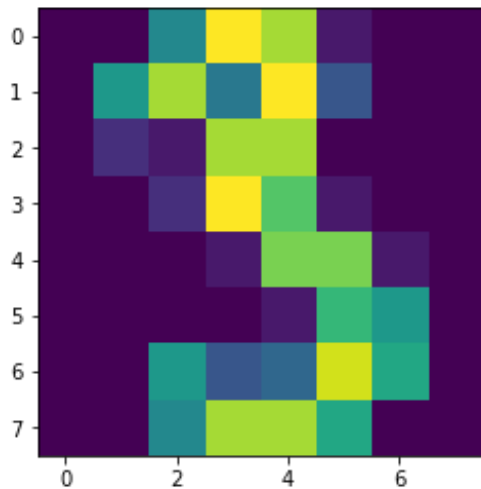
```
In [33]: digits['data'].shape
```

```
Out[33]: (1797, 64)
```

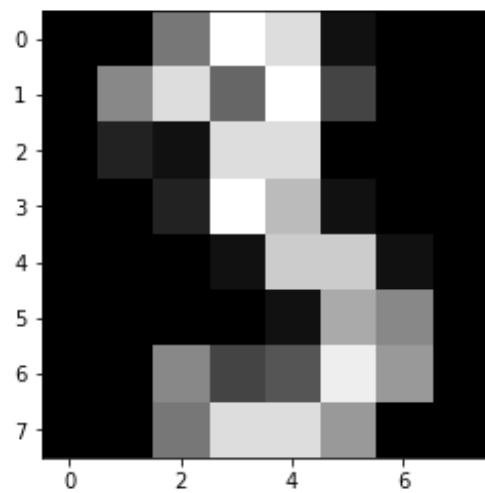
```
In [34]: digits.images.shape
```

```
Out[34]: (1797, 8, 8)
```

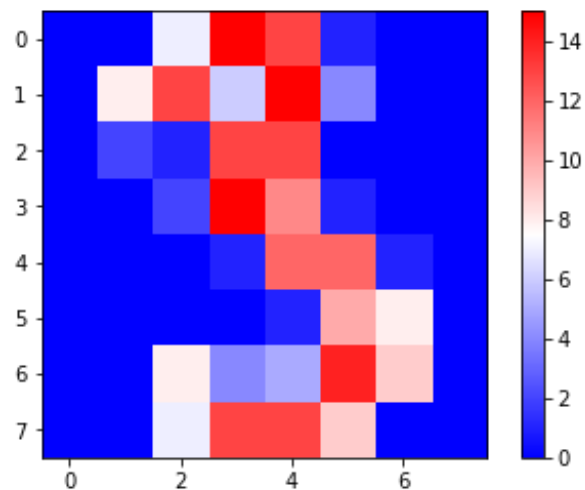
```
In [35]: plt.imshow(digits.images[3])  
plt.show()
```



```
In [36]: plt.imshow(digits.images[3], cmap='gray')  
plt.show()
```



```
In [37]: plt.imshow(digits.images[3], cmap='bwr')  
plt.colorbar()  
plt.show()
```



see <http://www.cs.columbia.edu/~amueller/comsw4995s19/schedule/> for more detail

```
In [ ]:
```