

Veri Görselleme

Çizimler

Veri görsellemek için python da birçok kütüphane mevcuttur. Bu kütüphanelerin neredeyse hepsi matplotlib kütüphanesi üzerine kurulmuştur.

Matlab benzeri bir ortam oluşturulma hedefi güdülmüştür.

Bir görselin (figür) yapısını incelediğimizde aşağıdaki unsurlardan oluştuğunu görebiliriz. Plot = çizim

title

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

# Verimizi hazırlayalım

# 0 dan 2 ye kadar 0.01 araliklarla bir dizi olustur.
t = np.arange(0.0, 2.0, 0.01)
# t icin, sin (2pi * t) sini hesaplayarak yeni bir dizi olustur.
s = 1 + np.sin(2 * np.pi * t)

print(t)
print("\nLength of array t:>>>",len(t),"<<<\n")
print("\nShape of array t:>>>",np.shape(t),"<<<\n")
print(s)
print("\nLength of array s:>>>",len(s),"<<<\n")
print("\nShape of array s:>>>",np.shape(s),"<<<\n")
```

[0.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.11	0.12	0.13
0.14	0.15	0.16	0.17	0.18	0.19	0.2	0.21	0.22	0.23	0.24	0.25	0.26	0.27
0.28	0.29	0.3	0.31	0.32	0.33	0.34	0.35	0.36	0.37	0.38	0.39	0.4	0.41
0.42	0.43	0.44	0.45	0.46	0.47	0.48	0.49	0.5	0.51	0.52	0.53	0.54	0.55
0.56	0.57	0.58	0.59	0.6	0.61	0.62	0.63	0.64	0.65	0.66	0.67	0.68	0.69
0.7	0.71	0.72	0.73	0.74	0.75	0.76	0.77	0.78	0.79	0.8	0.81	0.82	0.83
0.84	0.85	0.86	0.87	0.88	0.89	0.9	0.91	0.92	0.93	0.94	0.95	0.96	0.97
0.98	0.99	1.	1.01	1.02	1.03	1.04	1.05	1.06	1.07	1.08	1.09	1.1	1.11
1.12	1.13	1.14	1.15	1.16	1.17	1.18	1.19	1.2	1.21	1.22	1.23	1.24	1.25
1.26	1.27	1.28	1.29	1.3	1.31	1.32	1.33	1.34	1.35	1.36	1.37	1.38	1.39
1.4	1.41	1.42	1.43	1.44	1.45	1.46	1.47	1.48	1.49	1.5	1.51	1.52	1.53
1.54	1.55	1.56	1.57	1.58	1.59	1.6	1.61	1.62	1.63	1.64	1.65	1.66	1.67
1.68	1.69	1.7	1.71	1.72	1.73	1.74	1.75	1.76	1.77	1.78	1.79	1.8	1.81
1.82	1.83	1.84	1.85	1.86	1.87	1.88	1.89	1.9	1.91	1.92	1.93	1.94	1.95

```
1.96 1.97 1.98 1.99]
```

```
Length of array t:>>> 200 <<<
```

```
Shape of array t:>>> (200,) <<<
```

```
[1.00000000e+00 1.06279052e+00 1.12533323e+00 1.18738131e+00
1.24868989e+00 1.30901699e+00 1.36812455e+00 1.42577929e+00
1.48175367e+00 1.53582679e+00 1.58778525e+00 1.63742399e+00
1.68454711e+00 1.72896863e+00 1.77051324e+00 1.80901699e+00
1.84432793e+00 1.87630668e+00 1.90482705e+00 1.92977649e+00
1.95105652e+00 1.96858316e+00 1.98228725e+00 1.99211470e+00
1.99802673e+00 2.00000000e+00 1.99802673e+00 1.99211470e+00
1.98228725e+00 1.96858316e+00 1.95105652e+00 1.92977649e+00
1.90482705e+00 1.87630668e+00 1.84432793e+00 1.80901699e+00
1.77051324e+00 1.72896863e+00 1.68454711e+00 1.63742399e+00
1.58778525e+00 1.53582679e+00 1.48175367e+00 1.42577929e+00
1.36812455e+00 1.30901699e+00 1.24868989e+00 1.18738131e+00
1.12533323e+00 1.06279052e+00 1.00000000e+00 9.37209480e-01
8.74666766e-01 8.12618685e-01 7.51310113e-01 6.90983006e-01
6.31875447e-01 5.74220708e-01 5.18246326e-01 4.64173205e-01
4.12214748e-01 3.62576010e-01 3.15452894e-01 2.71031373e-01
2.29486757e-01 1.90983006e-01 1.55672074e-01 1.23693320e-01
9.51729475e-02 7.02235141e-02 4.89434837e-02 3.14168389e-02
1.77127493e-02 7.88529869e-03 1.97327157e-03 0.00000000e+00
1.97327157e-03 7.88529869e-03 1.77127493e-02 3.14168389e-02
4.89434837e-02 7.02235141e-02 9.51729475e-02 1.23693320e-01
1.55672074e-01 1.90983006e-01 2.29486757e-01 2.71031373e-01
3.15452894e-01 3.62576010e-01 4.12214748e-01 4.64173205e-01
5.18246326e-01 5.74220708e-01 6.31875447e-01 6.90983006e-01
7.51310113e-01 8.12618685e-01 8.74666766e-01 9.37209480e-01
1.00000000e+00 1.06279052e+00 1.12533323e+00 1.18738131e+00
1.24868989e+00 1.30901699e+00 1.36812455e+00 1.42577929e+00
1.48175367e+00 1.53582679e+00 1.58778525e+00 1.63742399e+00
1.68454711e+00 1.72896863e+00 1.77051324e+00 1.80901699e+00
1.84432793e+00 1.87630668e+00 1.90482705e+00 1.92977649e+00
1.95105652e+00 1.96858316e+00 1.98228725e+00 1.99211470e+00
1.99802673e+00 2.00000000e+00 1.99802673e+00 1.99211470e+00
1.98228725e+00 1.96858316e+00 1.95105652e+00 1.92977649e+00
1.90482705e+00 1.87630668e+00 1.84432793e+00 1.80901699e+00
1.77051324e+00 1.72896863e+00 1.68454711e+00 1.63742399e+00
1.58778525e+00 1.53582679e+00 1.48175367e+00 1.42577929e+00
1.36812455e+00 1.30901699e+00 1.24868989e+00 1.18738131e+00
1.12533323e+00 1.06279052e+00 1.00000000e+00 9.37209480e-01
8.74666766e-01 8.12618685e-01 7.51310113e-01 6.90983006e-01
6.31875447e-01 5.74220708e-01 5.18246326e-01 4.64173205e-01
4.12214748e-01 3.62576010e-01 3.15452894e-01 2.71031373e-01
2.29486757e-01 1.90983006e-01 1.55672074e-01 1.23693320e-01
9.51729475e-02 7.02235141e-02 4.89434837e-02 3.14168389e-02]
```

```
1.77127493e-02 7.88529869e-03 1.97327157e-03 0.00000000e+00
1.97327157e-03 7.88529869e-03 1.77127493e-02 3.14168389e-02
4.89434837e-02 7.02235141e-02 9.51729475e-02 1.23693320e-01
1.55672074e-01 1.90983006e-01 2.29486757e-01 2.71031373e-01
3.15452894e-01 3.62576010e-01 4.12214748e-01 4.64173205e-01
5.18246326e-01 5.74220708e-01 6.31875447e-01 6.90983006e-01
7.51310113e-01 8.12618685e-01 8.74666766e-01 9.37209480e-01]
```

```
Length of array s:>>> 200 <<<
```

```
Shape of array s:>>> (200,) <<<
```

Not: Verilerin miktarini 'size' özelliği ile elde edebilirsiniz

Not 2: Verilerin şeklini shape özelliği ile elde edebilirsiniz

Not 3: Verilerinizi print() fonksiyonu ile ekrana yazdırmak size faydalı olacaktır.

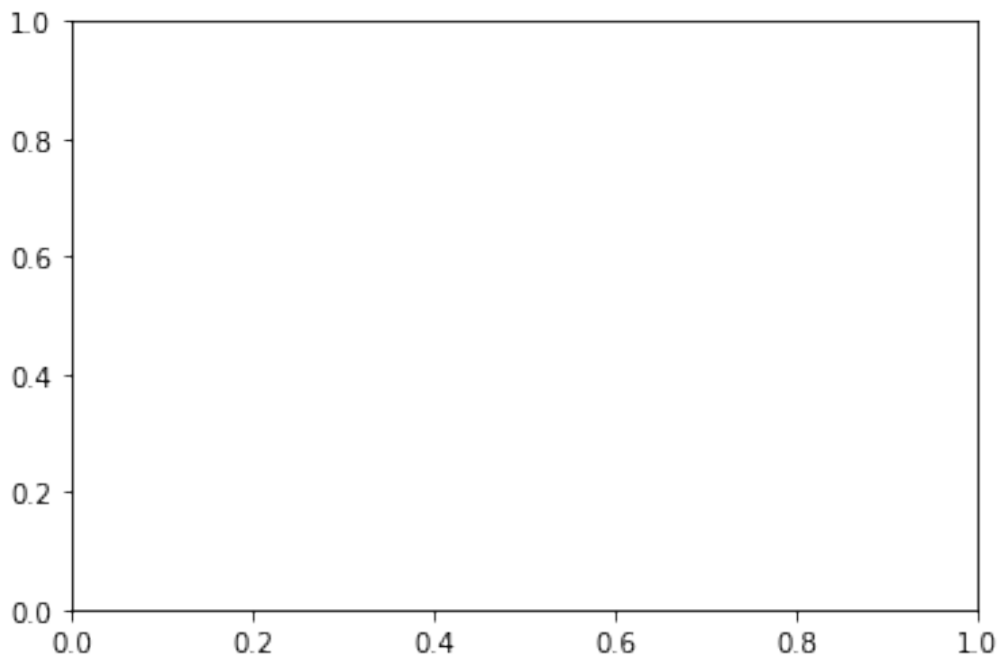
pyplot kütüphanesinin subplots komutunu kullanarak bir figür yaratabiliyoruz.

subplots komutuna daha fazla parametre de vermemiz mümkün.

nrows satır sayısı ncols sütun sayısı sharex ve sharey aynı x ve y eksenleri kullanmasını belirtir

subplots komutu bize geri donuste bir figure nesnesi bir de eksenler listesi verir

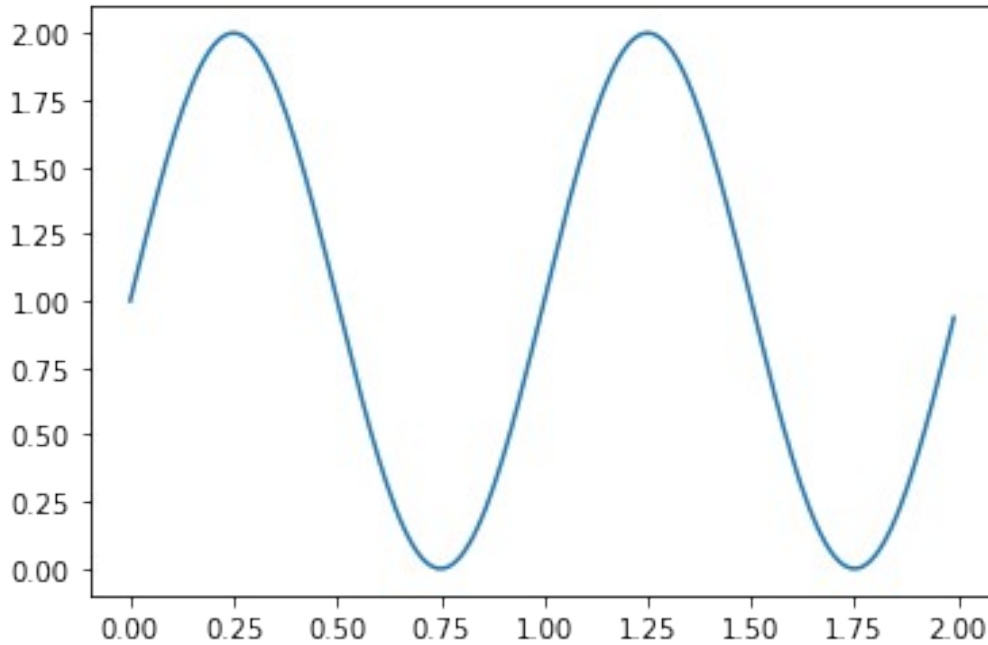
```
fig, ax = plt.subplots()
```



Ürettiğimiz verileri görselimizin eksenlerine yükleyelim.

```
fig, ax = plt.subplots()  
ax.plot(t, s)
```

```
[<matplotlib.lines.Line2D at 0x115f48828>]
```

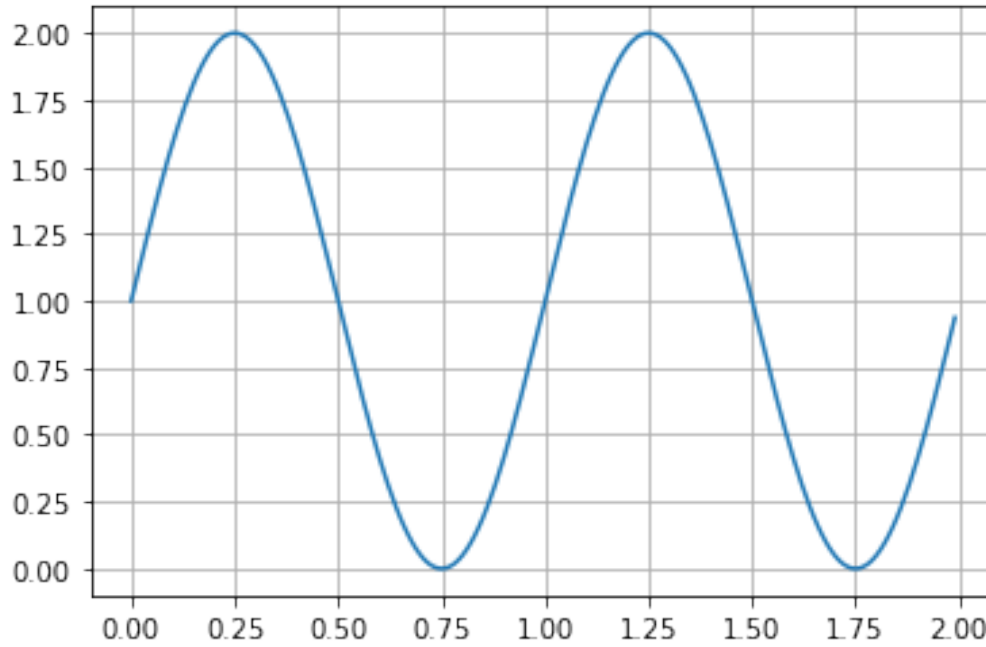


plt.show komutu ile, görselimizi ekranda gösterebiliyoruz. jupyter notebook dışındaki ortamlarda, bu ayrı bir pencere olarak çıkacaktır.

```
plt.show()
```

Çizimizi daha güzel görebilmek için kılavuz çizgilerini (grid) ekleyelim.

```
fig, ax = plt.subplots()  
ax.plot(t, s)  
ax.grid()
```



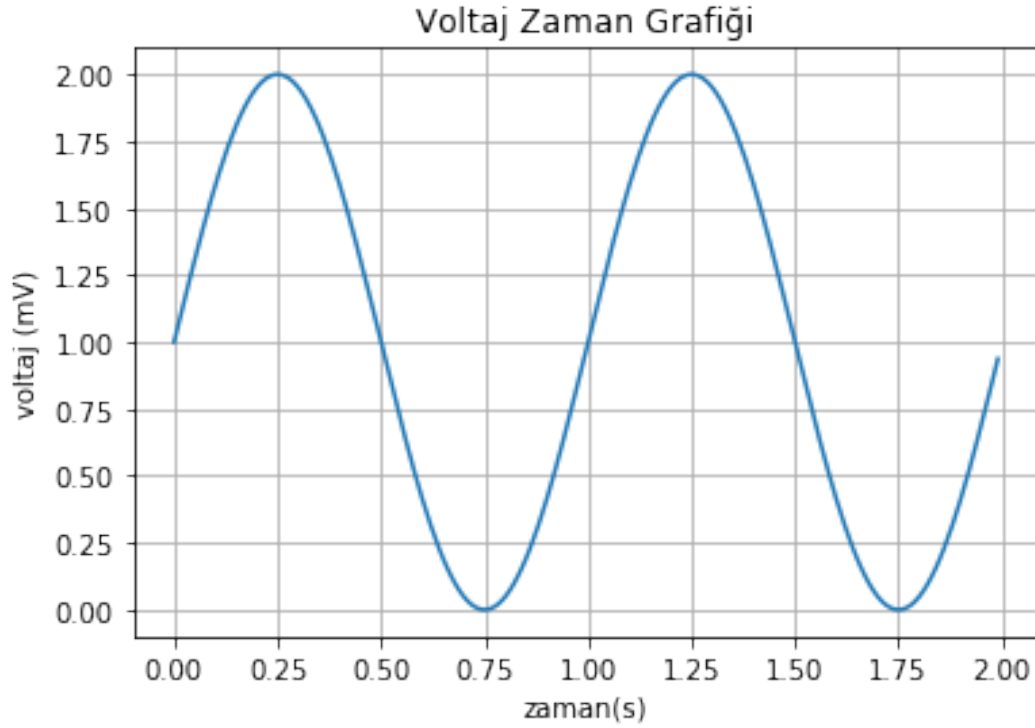
```
plt.show()
```

Çizime başlık ve eksenlere yazılarını ekleyelim.

```
fig, ax = plt.subplots()
ax.plot(t, s)
ax.grid()

ax.set_xlabel='zaman(s)', ylabel = 'voltaj (mV)', title='Voltaj Zaman
Grafiği')

[Text(0, 0.5, 'voltaj (mV)'),
 Text(0.5, 0, 'zaman(s)'),
 Text(0.5, 1.0, 'Voltaj Zaman Grafiği')]
```



```
plt.show()
```

Dosyamızı kaydetmek için fig.save() komutunu kullanabiliriz.

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

# Veriler:
t = np.arange(0.0, 2.0, 0.01)
s = np.sin(2 * np.pi * t)

# Figürü oluşturun
fig, ax = plt.subplots()

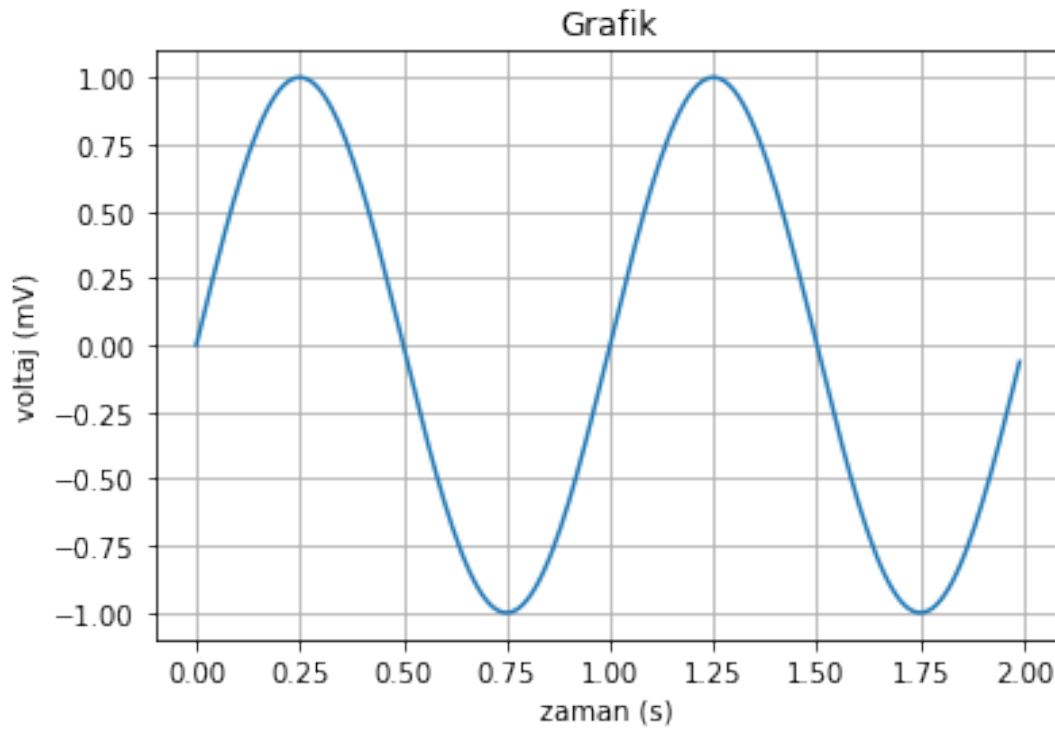
# Veriyi figüre koy
ax.plot(t, s)

# Gerekli metinleri ekleyelim
ax.set(xlabel='zaman (s)', ylabel='voltaj (mV)',
       title='Grafik')

# Kılavuz çizgileri
ax.grid()

# e = 3
# Görseli kaydet
```

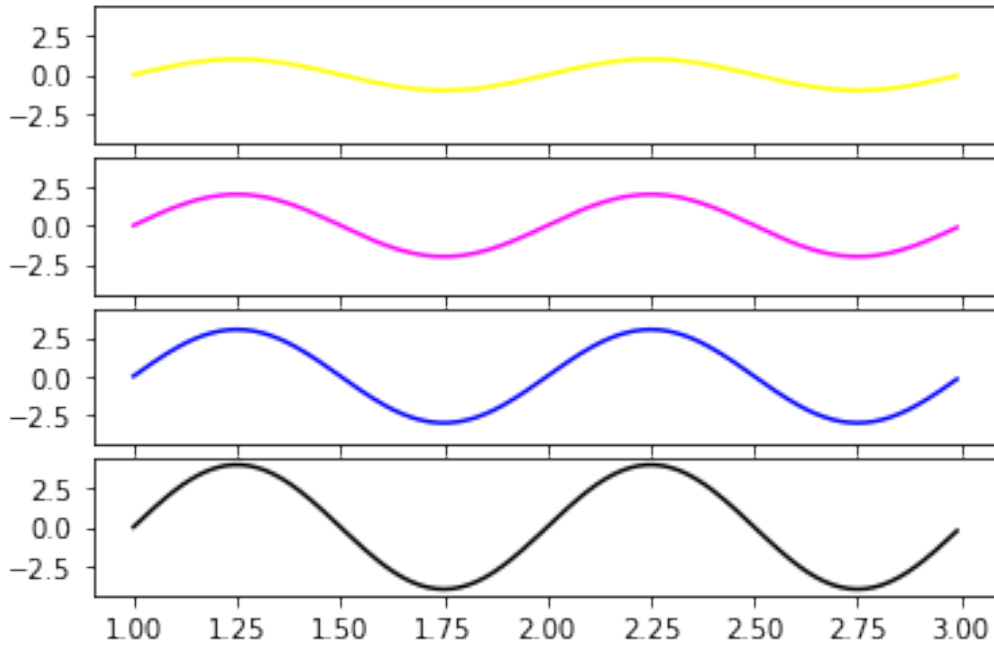
```
fig.savefig( "XXX.pdf")  
plt.show()
```



Aynı figürün içine birden fazla çizim eklemek isteyebiliriz. Bu durumda subplots'a uygun parametreleri vermemiz gerekir.

Birden çok grafiği aynı ekranda görüntülemek istersek

```
fig2, (ax1,ax2,ax3,ax4) = plt.subplots(4,1, sharey = True, sharex =  
False)  
  
ax1.plot(t,s, color = 'yellow')  
ax2.plot(t,s*2,color = 'magenta')  
ax3.plot(t,s*3, color = 'blue')  
ax4.plot(t+1,s*4, color = 'black')  
  
#fig2.tight_layout()  
#plt.subplots_adjust(hspace = .001)  
plt.show()
```



Grafiğimizdeki diğer çizimleri de istediğiniz şekilde renklendirin.

Grafiğimizin marjları ile oynama yapmamız da mümkün.

```
plt.subplots_adjust(left=None, bottom=None, right=None, top=None,  
wspace=None, hspace=None)
```

<Figure size 432x288 with 0 Axes>

Ürettiğimiz grafiklerdeki değişimleri görmek zor. Grafiğimizi daha anlaşılır olabilmesi için değiştirelim.

İlk önce y eksenlerinin ölçeklendirmesini aynı yaparsak anlamamız daha rahat olacaktır.

Bunun için grafiğimizin y eksenleri aynı olsun özelliğini etkinleştirelim.

```
sharey = True
```

Sonra grafiğimizdeki çizimleri bir sütunda göstermek yerine bir satırda gösterirsek yine karşılaştırma yapmamız daha rahat olacaktır.

Çizimlerimizi farklı şekillerde de yerleştirebiliriz.

```
fig2, ax = plt.subplots(2,2, sharey = True, sharex = True)
```

```
# ax in şekli ne???  
print(ax.shape)
```

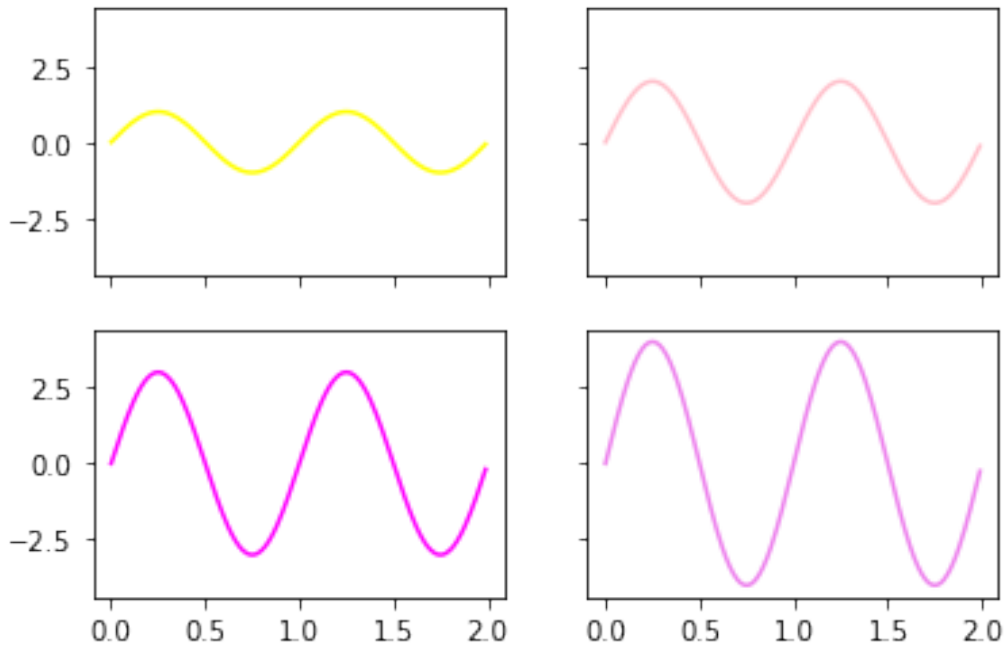
```
ax[0,0].plot(t,s, color = 'yellow')  
ax[0,1].plot(t,s*2, color = 'pink')
```



```
ax[1,0].plot(t,s*3, color = 'magenta')
ax[1,1].plot(t,s*4, color = 'violet')

plt.show()

(2, 2)
```



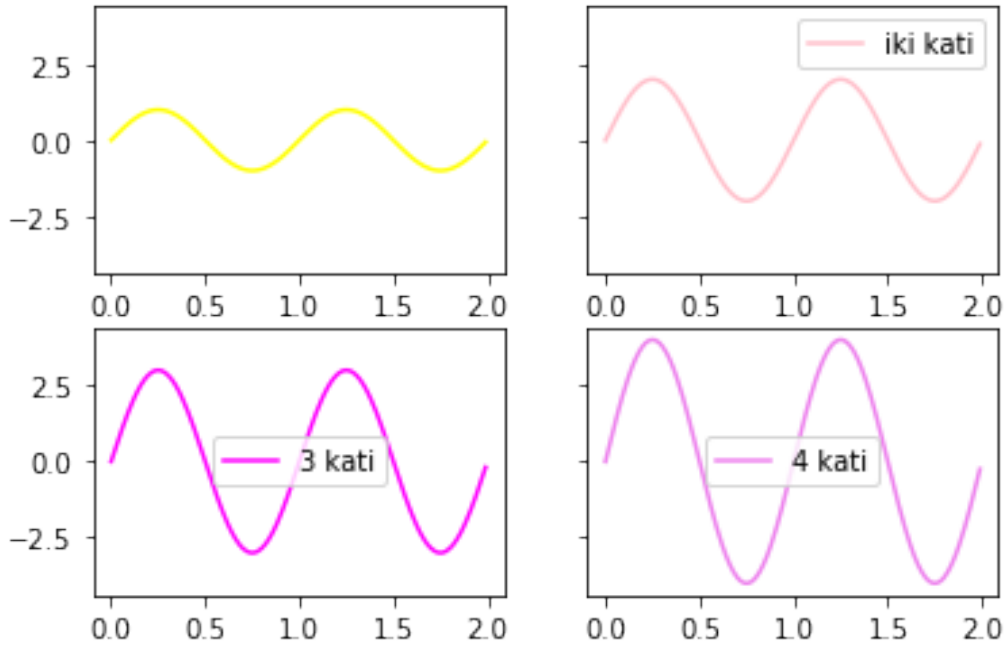
Çizimlerimizin ne olduklarını görsellerin üzerine not almak isteyebiliriz. Bu durumda etiket (label) özelliğini kullanabiliriz.

```
fig2, ax = plt.subplots(2,2, sharey = True)

ax[0,0].plot(t,s, color = 'yellow', label = 'bizimki')
ax[0,1].plot(t,s*2, color = 'pink', label = 'iki kati')
ax[1,0].plot(t,s*3, color = 'magenta', label = '3 kati')
ax[1,1].plot(t,s*4, color = 'violet', label = '4 kati')

#ax[0,0].legend()
ax[0,1].legend()
ax[1,0].legend()
ax[1,1].legend()

plt.show()
```



```
fig2, ax = plt.subplots(2,2, sharey = True)

ax[0,0].plot(t,s, color = 'yellow', label = 'bizimki')
ax[0,1].plot(t,s*2, color = 'pink', label = 'iki kati')
ax[1,0].plot(t,s*3, color = 'magenta', label = '3 kati')
ax[1,1].plot(t,s*4, color = 'violet', label = '4 kati')

ax[0,0].set_title('Sarı Çizgi')
ax[0,0].set_xlabel('s')
ax[0,0].set_ylabel('mV')

ax[1,0].set_title('Pembe Çizgi')
ax[1,0].set_xlabel('s')
ax[1,0].set_ylabel('mV')

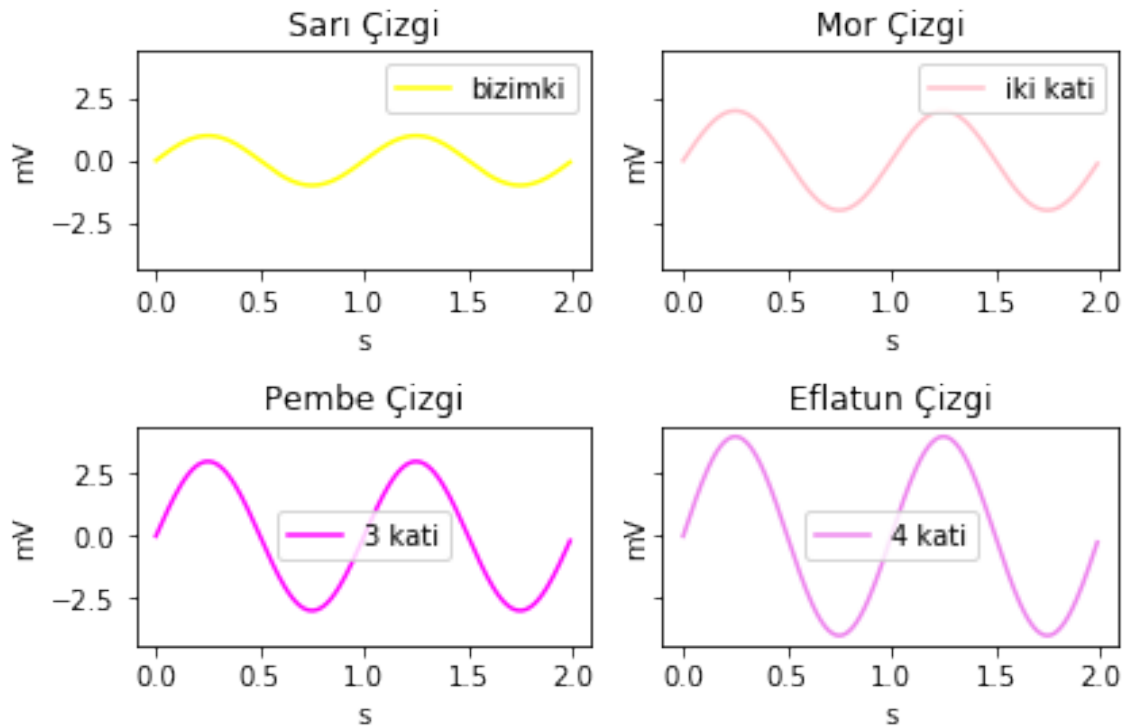
ax[0,1].set_title('Mor Çizgi')
ax[0,1].set_xlabel('s')
ax[0,1].set_ylabel('mV')

ax[1,1].set_title('Eflatun Çizgi')
ax[1,1].set_xlabel('s')
ax[1,1].set_ylabel('mV')

ax[0,0].legend()
ax[0,1].legend()
ax[1,0].legend()
ax[1,1].legend()
```

```
fig2.suptitle('Güzel Çizgilerimiz', fontsize = 26,y= 1.1)
fig2.tight_layout()
plt.show()
```

Güzel Çizgilerimiz



Görselimizin etiketinin yerini değiştirmek için, dikey eksendeki koordinatlarını değiştirebiliriz.

```
fig2.suptitle('Güzel Çizgilerimiz', fontsize = 16, y= 1.1)
```

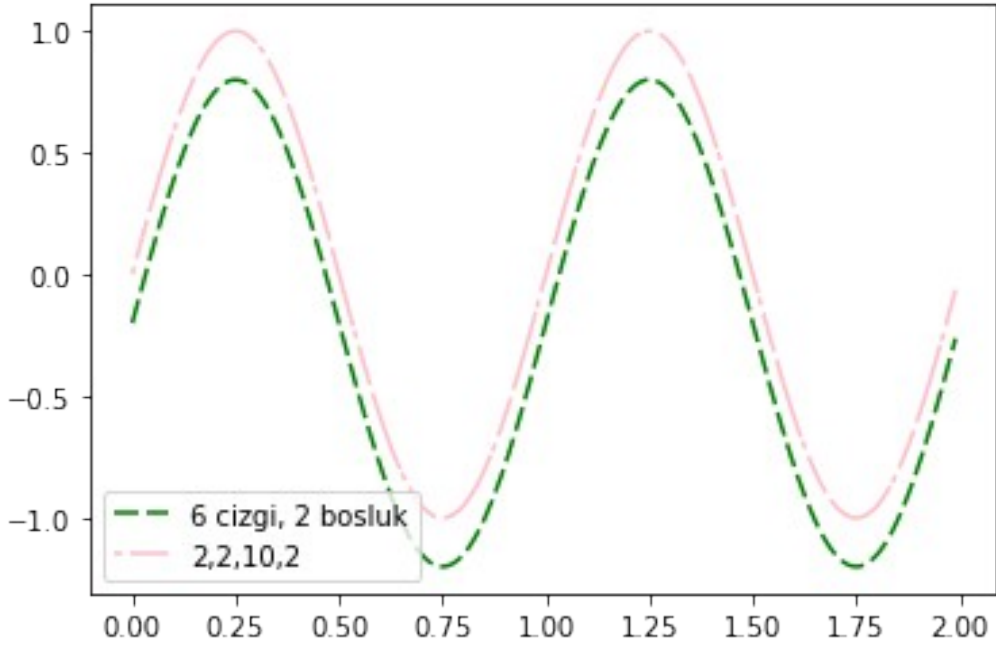
plot komutu birden fazla kez çağırıldığında, yeni bir çizim ekler.

```
fig, ax = plt.subplots()

yesilCizgi = ax.plot(t, s - 0.2, dashes=[6, 2], label='6 çizgi, 2 boşluk', color = 'green')

cizgil, = ax.plot(t, s, label='2,2,10,2')
cizgil.set_dashes([2,2,10,2,20,1])
cizgil.set_color('pink')

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



plot komutunun diğer özelliklerini görmek için aşağıdaki linke tıklayın

https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html?highlight=plot#matplotlib.pyplot.plot

İki farklı türde veriyi de aynı görsel üzerine çizdirmemiz mümkün.

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

# Verilerimizi olusturalim
t = np.arange(0.01, 10, 0.01)
data1 = np.exp(t)
data2 = np.sin(2 * np.pi * t)

fig, ax1 = plt.subplots()
color = 'red'
ax1.set_xlabel('zaman (s)')
ax1.set_ylabel('exp', color=color)
ax1.plot(t, data1, color=color)
ax1.tick_params(axis='y', labelcolor=color)

ax2 = ax1.twinx() # x eksenini ortak baska bir eksen olusturalim

color = 'blue'
ax2.set_ylabel('sin', color=color) # x etiketi ax1 ile ortak
ax2.plot(t, data2, color=color)
ax2.tick_params(axis='y', labelcolor=color)

fig.tight_layout() # y etiketleri duzgun ciksin diye
plt.show()
```

