

In []:

6- Area Plot- Alan Grafiği:

Tanım:

Alan grafiği tıpkı çizgi grafiği gibi bir değişkenin değişimini gösterir. X eksenini genellikle zamanı veya düzenlenmiş bir veriyi temsil eder. Y eksenini ise o noktadaki nümerik değeri temsil eder. Çizgi grafiğinden farklı olarak çizginin alt kısmındaki alan belirtilir.

In []:

In [25]:

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

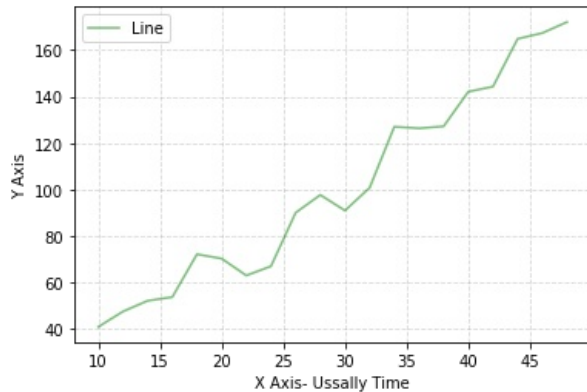
# Fixing random state for reproducibility
np.random.seed(19680801)

x = np.arange(10, 50.0, 2.0)
y = x ** 1.3 + np.random.rand(*x.shape) * 30.0
s = np.random.rand(*x.shape) * 800 + 500

plt.plot(x, y, c="g", alpha=0.5,
         label="Line")

#plt.plot(x, y, s, c="g", alpha=0.5, marker=r'o')

plt.xlabel("X Axis- Ussally Time")
plt.ylabel("Y Axis")
plt.legend(loc='upper left')
plt.grid(color='gray', linestyle='dashed', alpha=0.3)
plt.show()
```



In [38]:

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

# Fixing random state for reproducibility
np.random.seed(19680801)

x = np.arange(10, 50.0, 2.0)
y = x ** 1.3 + np.random.rand(*x.shape) * 30.0
y2 = x ** 1.3 + np.random.rand(*x.shape) * 30.0
s = np.random.rand(*x.shape) * 800 + 500

plt.fill_between(x, y, alpha=0.5, label='Area')

"""
overall_mean =statistics.mean(y)
print(overall_mean)
#-----
plt.fill_between(x, y,overall_mean,

                 alpha=0.5, label='Area'
                )
#-----
"""
"""
#-----
plt.fill_between(x, y,overall_mean,
                 where=(y>overall_mean),
                 interpolate =True,
                 alpha=0.5, label='Area'
                )
plt.plot(x,y)

#-----
plt.fill_between(x, y,overall_mean,
                 where=(y<overall_mean),
                 interpolate =True,
                 alpha=0.5, label='Area',color='red'
                )

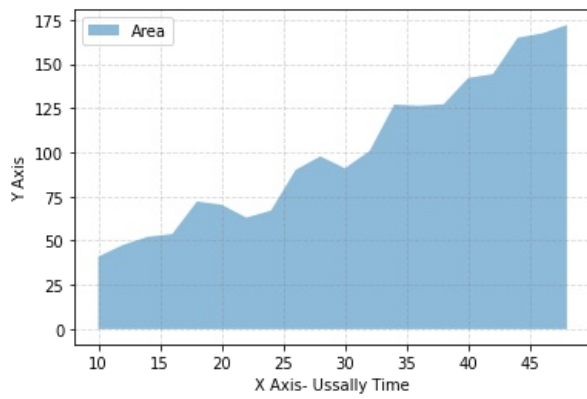
#-----
"""
"""
#-----
plt.fill_between(x, y,y2,
                 where=(y>y2),
                 interpolate =True,
                 alpha=0.5, label='Above'
                )
plt.plot(x,y,color='blue', label='Line1')
plt.plot(x,y2,color='black',linestyle='--',label='Line2')

#-----
plt.fill_between(x, y,y2,
                 where=(y<y2),
                 interpolate =True,
                 alpha=0.5, label='Below',color='red'
                )

#-----

"""
#plt.plot(x, y, s, c="g", alpha=0.5, marker=r'o')

plt.xlabel("X Axis- Usually Time")
plt.ylabel("Y Axis")
plt.legend(loc='upper left')
plt.grid(color='gray', linestyle='dashed', alpha=0.3)
plt.show()
```



In []:

In []:

In [64]:

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

# Fixing random state for reproducibility
np.random.seed(19680801)

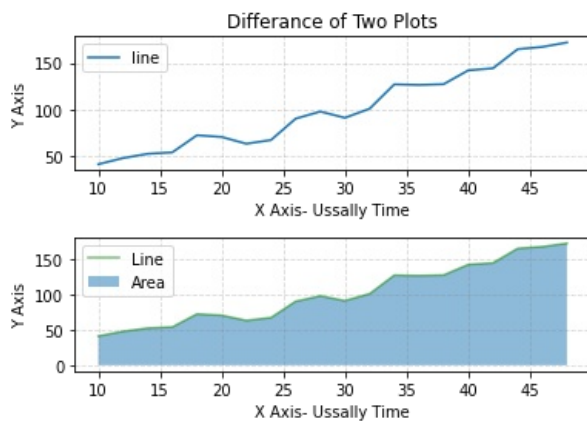
fig, (ax1, ax2) = plt.subplots(2, 1)
# make a little extra space between the subplots
fig.subplots_adjust(hspace=0.5)

x = np.arange(10, 50.0, 2.0)
y = x ** 1.3 + np.random.rand(*x.shape) * 30.0
s = np.random.rand(*x.shape) * 800 + 500

ax1.set_title('Differance of Two Plots')

ax1.plot(x,y,label='line')
ax1.set_xlabel("X Axis- Ussally Time")
ax1.set_ylabel("Y Axis")
ax1.legend(loc='upper left')
ax1.grid(color='gray', linestyle='dashed', alpha=0.3)
#####
plt.plot(x, y, c="g", alpha=0.5, label="Line")
ax2.fill_between(x,y, label='Area',alpha=0.5)
ax2.set_xlabel("X Axis- Ussally Time")
ax2.set_ylabel("Y Axis")
ax2.legend(loc='upper left')
ax2.grid(color='gray', linestyle='dashed', alpha=0.3)

plt.show()
```



Yukarıda her iki grafik arasındaki fark veya benzerlik görülmektedir. Sadece bazı durumlarda Area kullanmak avantaj sağlamaktadır.

Ne için Kullanılır:

Alan grafiği bir değişkenin zamana veya düzenlenmiş bir diğer değişkene bağlı değişimini gösterir. Çizgi grafik ile arasında gösterim farkı olmamakla birlikte bazı durumlarda daha okunaklı bir grafik olmaktadır.

In [71]:

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

# Fixing random state for reproducibility
np.random.seed(19680801)

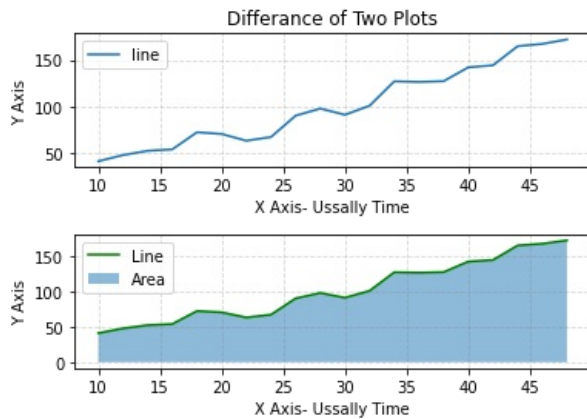
fig, (ax1, ax2) = plt.subplots(2, 1)
# make a little extra space between the subplots
fig.subplots_adjust(hspace=0.5)

x = np.arange(10, 50.0, 2.0)
y = x ** 1.3 + np.random.rand(*x.shape) * 30.0
s = np.random.rand(*x.shape) * 800 + 500

ax1.set_title('Differance of Two Plots')

ax1.plot(x,y,label='line')
ax1.set_xlabel("X Axis- Ussally Time")
ax1.set_ylabel("Y Axis")
ax1.legend(loc='upper left')
ax1.grid(color='gray', linestyle='dashed', alpha=0.3)
#####
plt.plot(x, y, c="g", alpha=1, label="Line")
ax2.fill_between(x,y, label='Area',alpha=0.5)
ax2.set_xlabel("X Axis- Ussally Time")
ax2.set_ylabel("Y Axis")
ax2.legend(loc='upper left')
ax2.grid(color='gray', linestyle='dashed', alpha=0.3)

plt.show()
```



Çeşitleri:

Alan grafiklerinin kümülatif -Stacked Area Chart- versiyonu bulunmaktadır ve bu verilerin toplanması gerektiği yani kümülatif değer teşkil ettiği durumlarda kullanılır. Bunun için ayrıca bir başlık oluşturulacaktır.

In []:

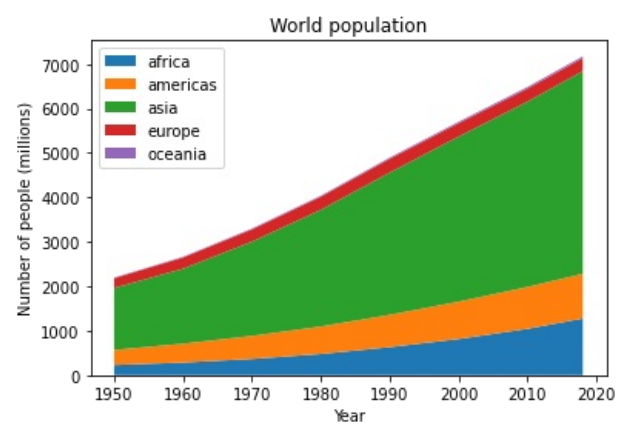
In [68]:

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

# data from United Nations World Population Prospects (Revision 2019)
# https://population.un.org/wpp/, license: CC BY 3.0 IGO
year = [1950, 1960, 1970, 1980, 1990, 2000, 2010, 2018]
population_by_continent = {
    'africa': [228, 284, 365, 477, 631, 814, 1044, 1275],
    'americas': [340, 425, 519, 619, 727, 840, 943, 1006],
    'asia': [1394, 1686, 2120, 2625, 3202, 3714, 4169, 4560],
    'europe': [220, 253, 276, 295, 310, 303, 294, 293],
    'oceania': [12, 15, 19, 22, 26, 31, 36, 39],
}

fig, ax = plt.subplots()
ax.stackplot(year, population_by_continent.values(),
            labels=population_by_continent.keys())
ax.legend(loc='upper left')
ax.set_title('World population')
ax.set_xlabel('Year')
ax.set_ylabel('Number of people (millions)')

plt.show()
```



In []:

Kullanım Hataları:

In []:

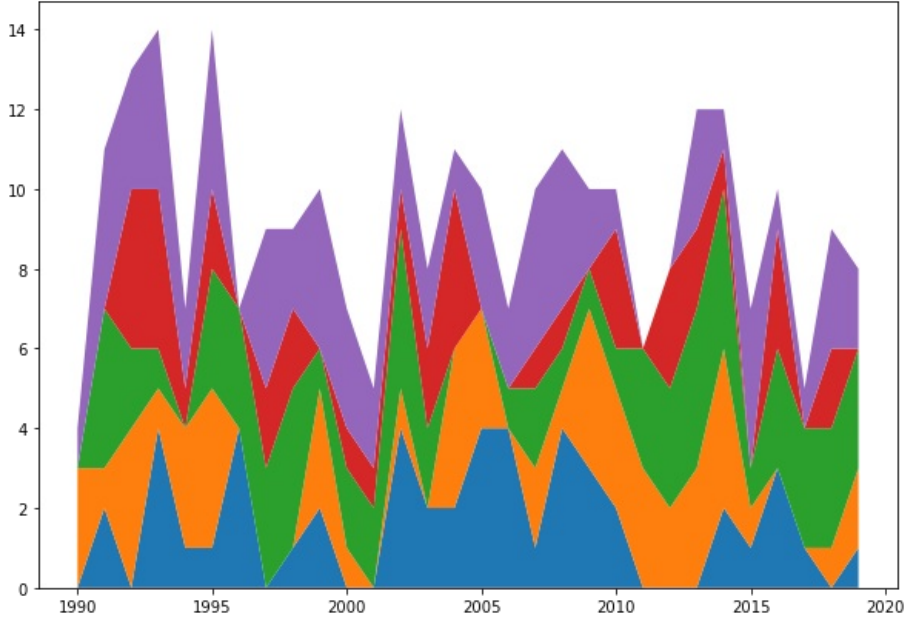
In [72]:

```
# Libraries
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats

x = np.arange(1990, 2020) # (N,) array-like

y = [np.random.randint(0, 5, size=30) for _ in range(5)] # (M, N) array-like

fig, ax = plt.subplots(figsize=(10, 7))
ax.stackplot(x, y);
```



Yukarıdaki grafik okunaklılık açısından kötü bir grafiktir, daha çok düzenli azalan veya yükselen veriler için uygundur.

In []:

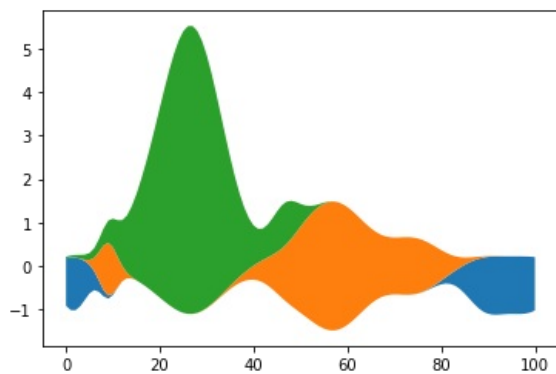
In [74]:

```
# Fixing random state for reproducibility
np.random.seed(19680801)

def gaussian_mixture(x, n=5):
    """Return a random mixture of *n* Gaussians, evaluated at positions *x*."""
    def add_random_gaussian(a):
        amplitude = 1 / (.1 + np.random.random())
        dx = x[-1] - x[0]
        x0 = (2 * np.random.random() - .5) * dx
        z = 10 / (.1 + np.random.random()) / dx
        a += amplitude * np.exp(-(z * (x - x0))**2)
    a = np.zeros_like(x)
    for j in range(n):
        add_random_gaussian(a)
    return a

x = np.linspace(0, 100, 101)
ys = [gaussian_mixture(x) for _ in range(3)]

fig, ax = plt.subplots()
ax.stackplot(x, ys, baseline='wiggle')
plt.show()
```



In []: