

201600779 김영민

In [8]:

```
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
from collections import Counter
import numpy as np
```

1-1번

In [2]:

```
from matplotlib import font_manager, rc
font_name = font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
rc('font', family=font_name)

def make_simple_line_chart():
    years = [1950, 1960, 1970, 1980, 1990, 2000, 2010]
    gdp = [300.2, 513.3, 1075.9, 2862.5, 8979.6, 10289.7, 14958.3]

    plt.plot(years, gdp, color='green', marker='v', linestyle='solid')

    plt.title('명목 GDP')

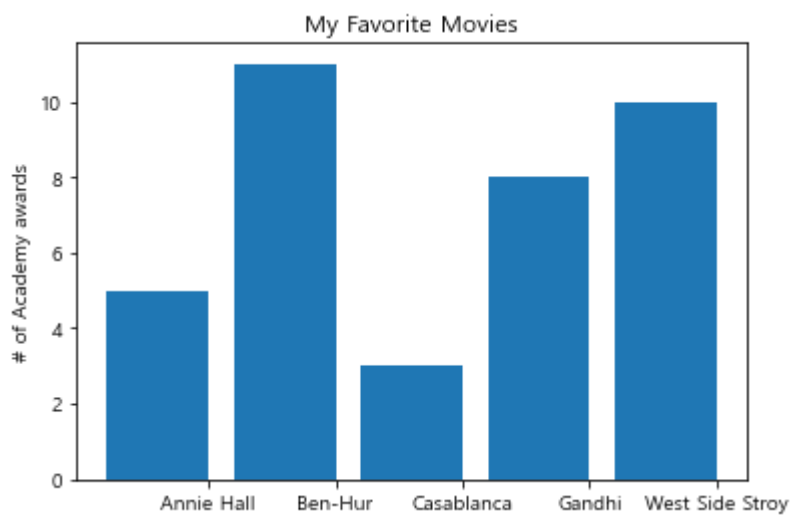
    plt.ylabel('Billions of $')
    plt.show()
make_simple_line_chart()
```



1-2번

In [3]:

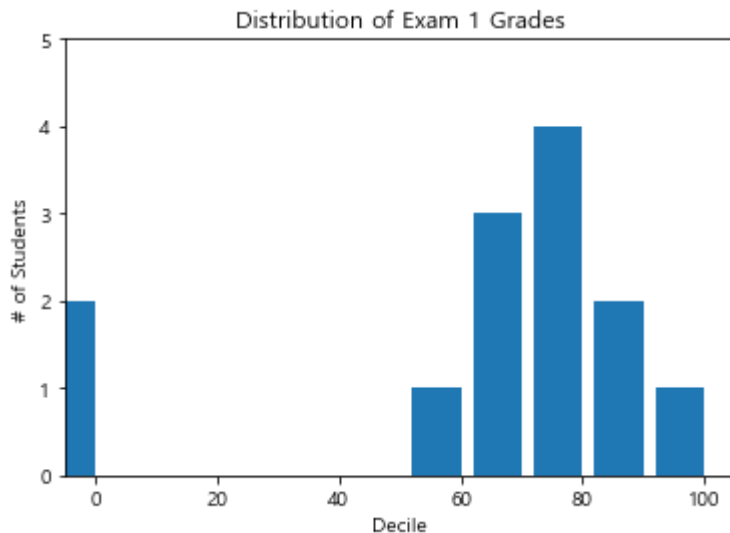
```
def make_simple_bar_chart():  
    movies= ['Annie Hall', 'Ben-Hur', 'Casablanca', 'Gandhi', 'West Side Stroy']  
    num_oscars = [5,11,3,8,10]  
  
    xs = [i+0.1 for i in range(len(movies))]  
    plt.bar(xs,num_oscars)  
    plt.ylabel("# of Academy awards")  
    plt.title("My Favorite Movies")  
  
    plt.xticks([i+0.5 for i in range(len(movies))],movies)  
    plt.show()  
make_simple_bar_chart()
```



1-3번

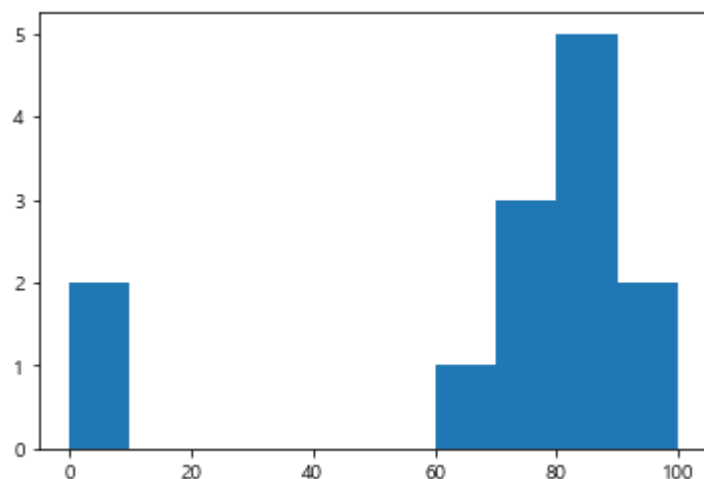
In [7]:

```
def make_simple_hist():
    grades = [83,95,91,87,70,0,85,82,100,67,73,77,0]
    decile = lambda grade: grade // 10*10
    histogram = Counter(decile(grade) for grade in grades)
    plt.bar([x-4 for x in histogram.keys()],histogram.values(),8)
    plt.axis([-5,105,0,5])
    plt.xlabel("Decile")
    plt.ylabel("# of Students")
    plt.title("Distribution of Exam 1 Grades")
    plt.show()
make_simple_hist()
```



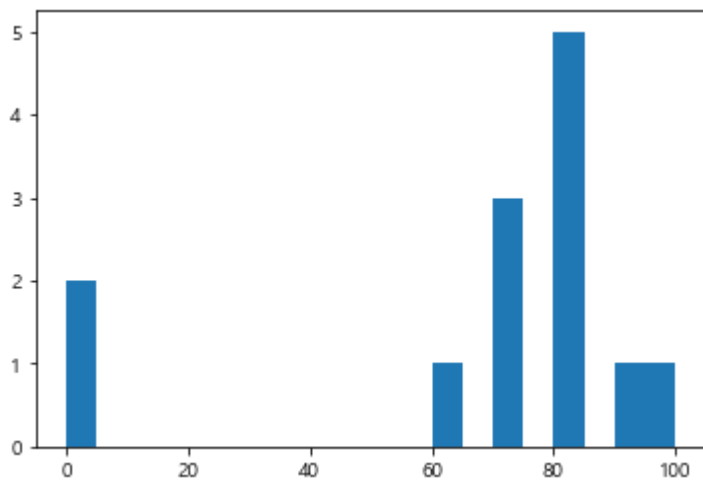
In [9]:

```
grades = np.array([83,85,91,87,70,0,85,82,100,67,73,77,0])
grade = grades // 10*10
plt.hist(grade)
plt.show()
```



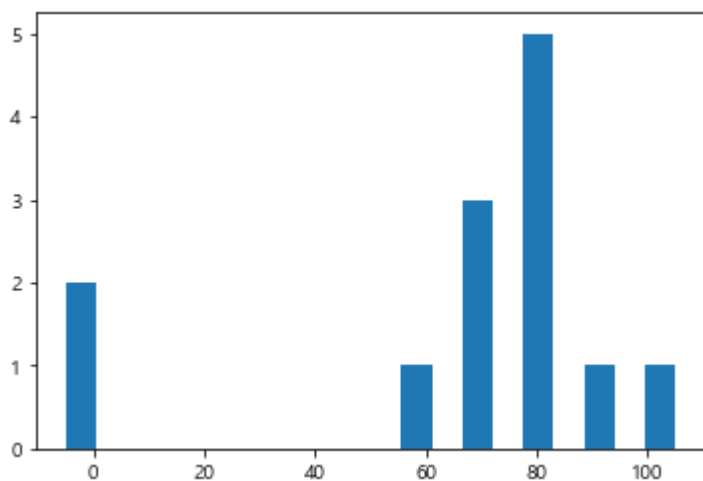
In [10]:

```
plt.hist(grade,bins = 20)  
plt.show()
```



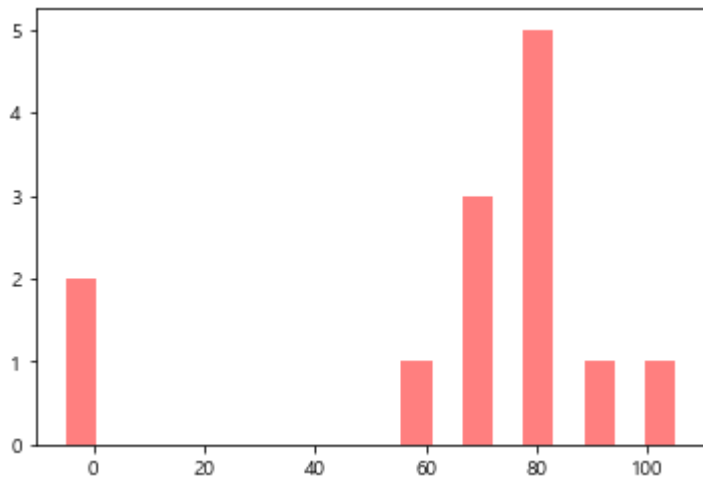
In [13]:

```
plt.hist(grade,bins=20 ,range = (grade.min()-5,grade.max()+5))  
plt.show()
```



In [14]:

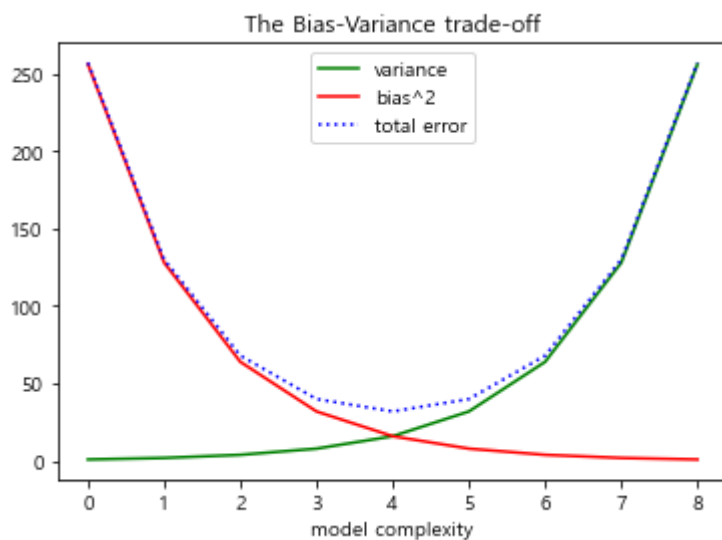
```
plt.hist(grade,bins=20 ,range = (grade.min()-5,grade.max()+5), alpha=.5,  
         histtype = 'stepfilled',color = 'red',edgecolor = 'none')  
plt.show()
```



1-4번

In [15]:

```
def make_several_line_charts():  
    variance = [1,2,4,8,16,32,64,128,256]  
    bias_squared = [256,128,64,32,16,8,4,2,1]  
    total_error = [x+y for x,y in zip(variance,bias_squared)]  
  
    xs = range(len(variance))  
  
    plt.plot(xs,variance,'g-',label='variance')  
    plt.plot(xs,bias_squared,'r-',label='bias^2')  
    plt.plot(xs,total_error,'b:',label='total error')  
  
    plt.legend(loc='upper center')  
    plt.xlabel("model complexity")  
    plt.title('The Bias-Variance trade-off')  
    plt.show()  
make_several_line_charts()
```



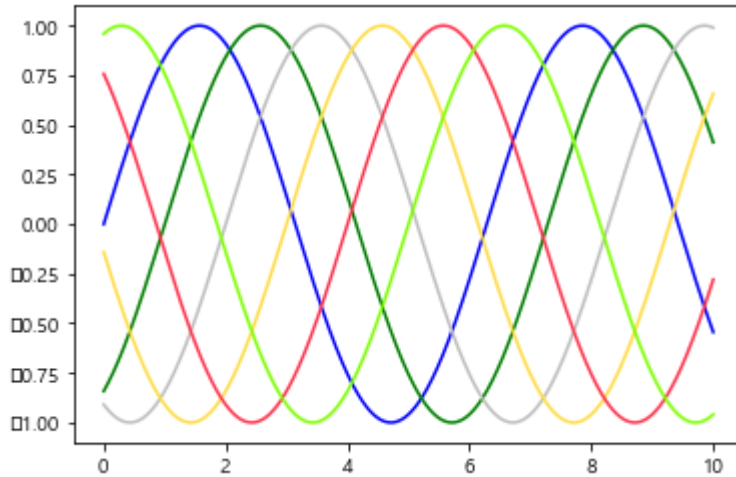
In [16]:

```

x = np.linspace(0, 10, 1000)

plt.plot(x, np.sin(x - 0), color='blue')      # specify color by name
plt.plot(x, np.sin(x - 1), color='g')        # short color code (rgbcmk)
plt.plot(x, np.sin(x - 2), color='0.75')     # Grayscale between 0 and 1
plt.plot(x, np.sin(x - 3), color='#FFDD44')  # Hex code (RRGGBB from 00 to FF)
plt.plot(x, np.sin(x - 4), color=(1.0,0.2,0.3)) # RGB tuple, values 0 to 1
plt.plot(x, np.sin(x - 5), color='chartreuse'); # all HTML color names supported

```



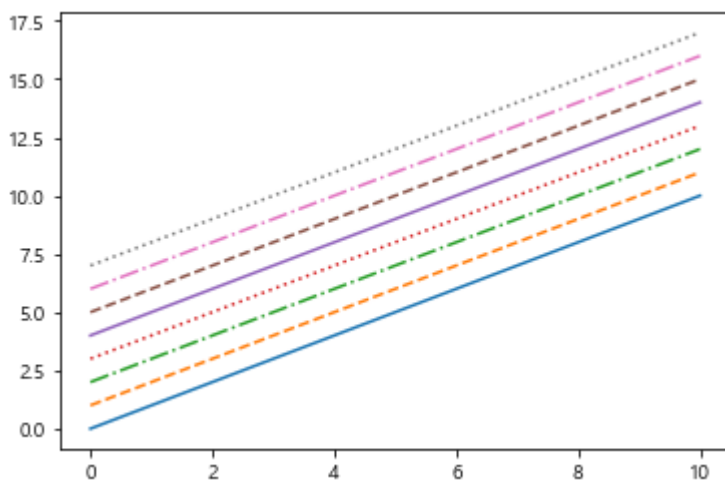
In [17]:

```

plt.plot(x, x + 0, linestyle='solid')
plt.plot(x, x + 1, linestyle='dashed')
plt.plot(x, x + 2, linestyle='dashdot')
plt.plot(x, x + 3, linestyle='dotted');

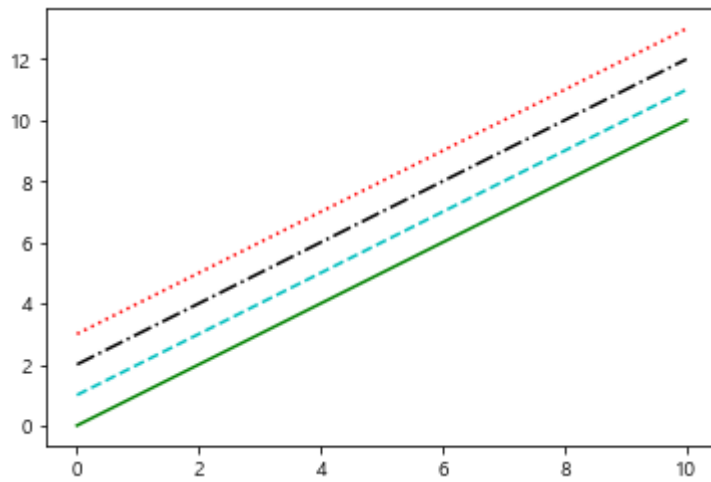
# For short, you can use the following codes:
plt.plot(x, x + 4, linestyle='-') # solid
plt.plot(x, x + 5, linestyle='--') # dashed
plt.plot(x, x + 6, linestyle='-.') # dashdot
plt.plot(x, x + 7, linestyle=':'); # dotted

```



In [18]:

```
plt.plot(x, x + 0, '-g') # solid green  
plt.plot(x, x + 1, '--c') # dashed cyan  
plt.plot(x, x + 2, '-.k') # dashdot black  
plt.plot(x, x + 3, ':r'); # dotted red
```

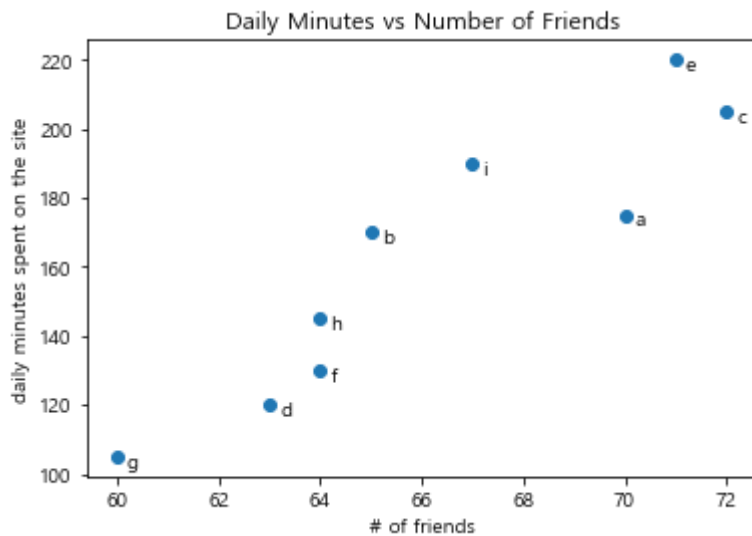


1-5번

In [19]:

```
def make_scatter_plot():
    friends =[70,65,72,63,71,64,60,64,67]
    minutes = [175,170,205,120,220,130,105,145,190]
    labels = ['a','b','c','d','e','f','g','h','i']
    plt.scatter(friends,minutes)

    for label,friend_count,minute_count in zip(labels,friends,minutes):
        plt.annotate(label,xy=(friend_count,minute_count),xytext=(5,-5),
                    textcoords = 'offset points')
    plt.title("Daily Minutes vs Number of Friends")
    plt.xlabel('# of friends')
    plt.ylabel("daily minutes spent on the site")
    plt.show()
make_scatter_plot()
```



In [24]:

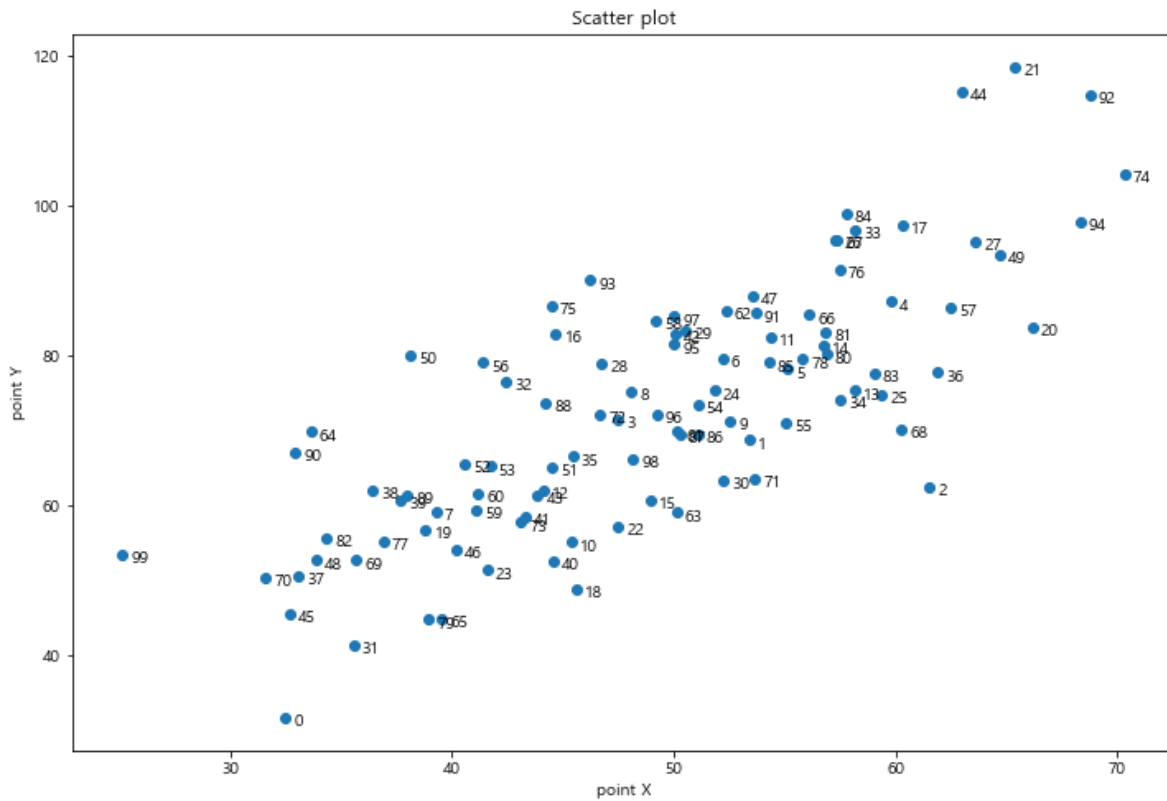
```
data = pd.read_csv('data.csv',header=None)
data.columns = ['x','y']
data.head()
```

Out[24]:

	x	y
0	32.502345	31.707006
1	53.426804	68.777596
2	61.530358	62.562382
3	47.475640	71.546632
4	59.813208	87.230925

In [22]:

```
plt.figure(figsize=(12,8))
plt.scatter(data['x'],data['y'])
plt.xlabel('point X')
plt.ylabel('point Y')
plt.title('Scatter plot')
for idx,i,j in zip(data.index,data['x'],data['y']):
    plt.annotate(idx,xy=(i,j),xytext=(5,-5),
                  textcoords = 'offset points')
```

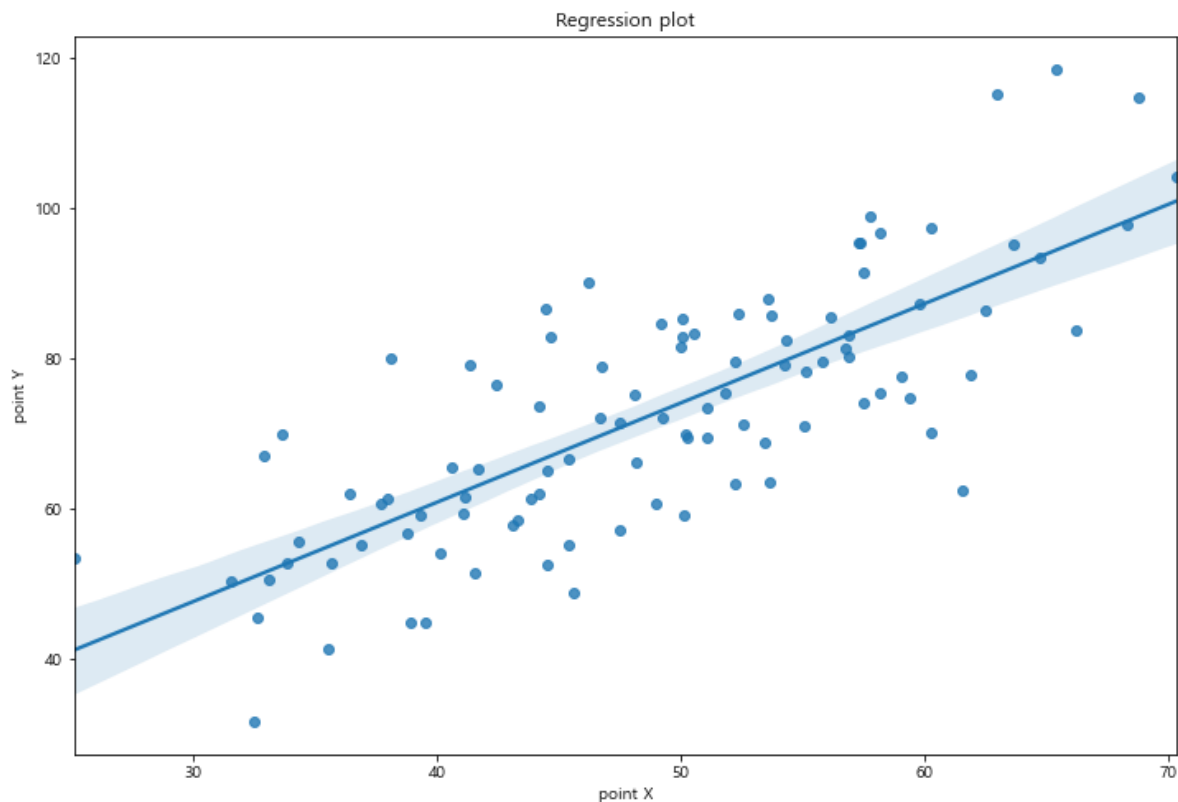


In [25]:

```
import seaborn as sns
plt.figure(figsize=(12,8))
sns.regplot(data=data,x='x',y='y')
plt.title('Regression plot')
plt.xlabel('point X')
plt.ylabel('point Y')
```

Out[25]:

Text(0, 0.5, 'point Y')



201600779 김영민

