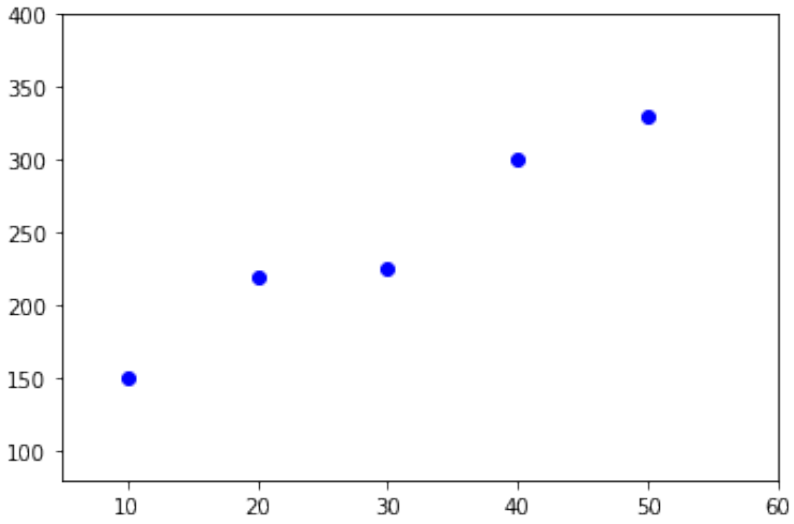
线性回归算法

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

In [2]:
```
x = np.array([10, 20, 30, 40, 50])
y = np.array([150, 220, 225, 300, 330])
```

In [3]:
```
plt.scatter(x, y, color='blue')
plt.xlim(5, 60)
plt.ylim(80, 400)
plt.show()
```



最小二乘法

In [4]:
```
import matplotlib.image as mpimg
```

In [5]:
```
img = mpimg.imread('formula1.png')
img.shape
plt.imshow(img)
plt.axis('off')
plt.show()
```

$$Q = \sum_{i=1}^{m} (y^{(i)} - ax^{(i)} - b)^2$$

当a, b取什么值时Q最小?

$$a = \frac{\sum_{i=1}^{m}(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum_{i=1}^{m}(x^{(i)} - \bar{x})^2} \qquad b = \bar{y} - a\bar{x}$$

In [6]:
```
x_mean = np.mean(x)
y_mean = np.mean(y)
```

In [7]:
```
u = 0    # 分子
v = 0    # 分母
```

In [8]:
```
for x_i, y_i in zip(x, y):
    u += (x_i - x_mean) * (y_i - y_mean)
    v +=  (x_i - x_mean) ** 2
```

In [9]:
```
a = u / v
b = y_mean - a * x_mean
```

In [10]:
```
a
```

Out[10]: 4.4

In [11]:
```
b
```

Out[11]: 113.0

In [12]:
```
x_predict = 55
y_predict = a * x_predict + b
```

In [13]:
```
y_predict
```

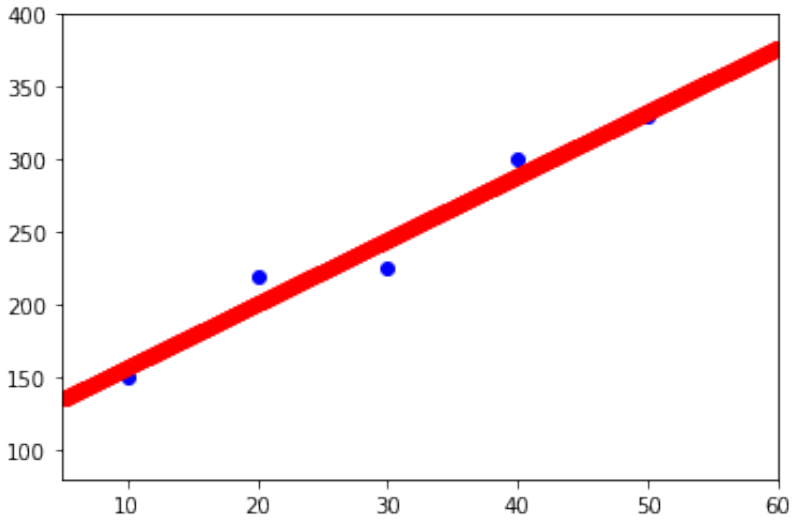
Out[13]: 355.0

In [14]:
```
x2 = np.linspace(5, 60, num = 1000)
```

In [15]:
```
plt.scatter(x, y, color='blue')
plt.xlim(5, 60)
plt.ylim(80, 400)

plt.scatter(x2, a * x2 + b, color='red')

plt.show()
```



In [16]:
```
from ML.linear import LinearRegression
```

In [17]:
```
lr = LinearRegression()
```

In [18]:
```
lr.fit(x, y)
```

Out[18]:
```
<ML.linear.LinearRegression at 0x118a40198>
```

In [21]:
```
lr.predict(np.array([55]))
```

Out[21]:
```
array([355.])
```