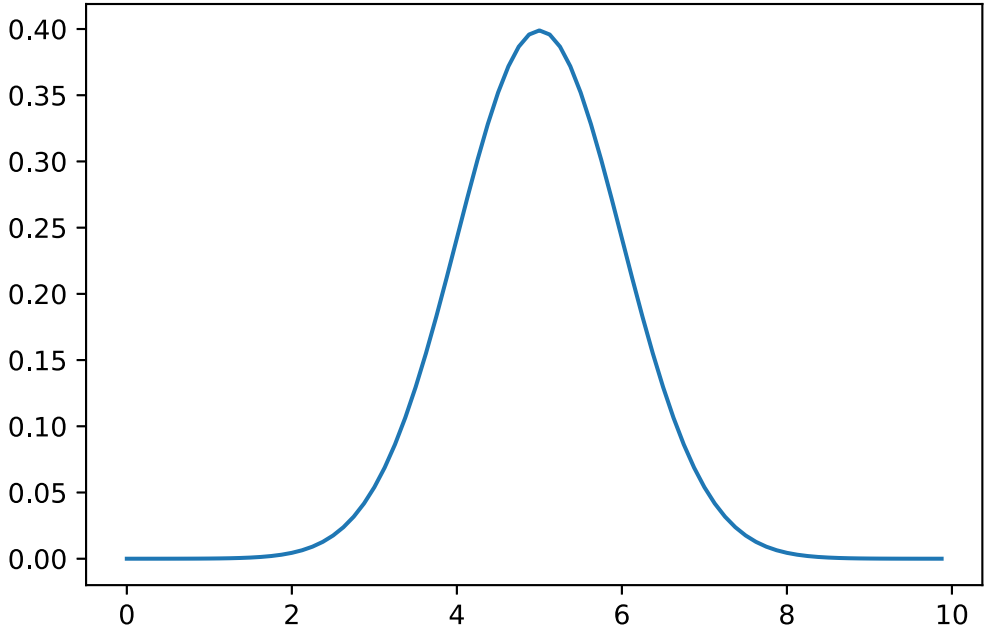
In [1]:
```
import random
import scipy
import numpy as np
import matplotlib.pyplot as plt
```

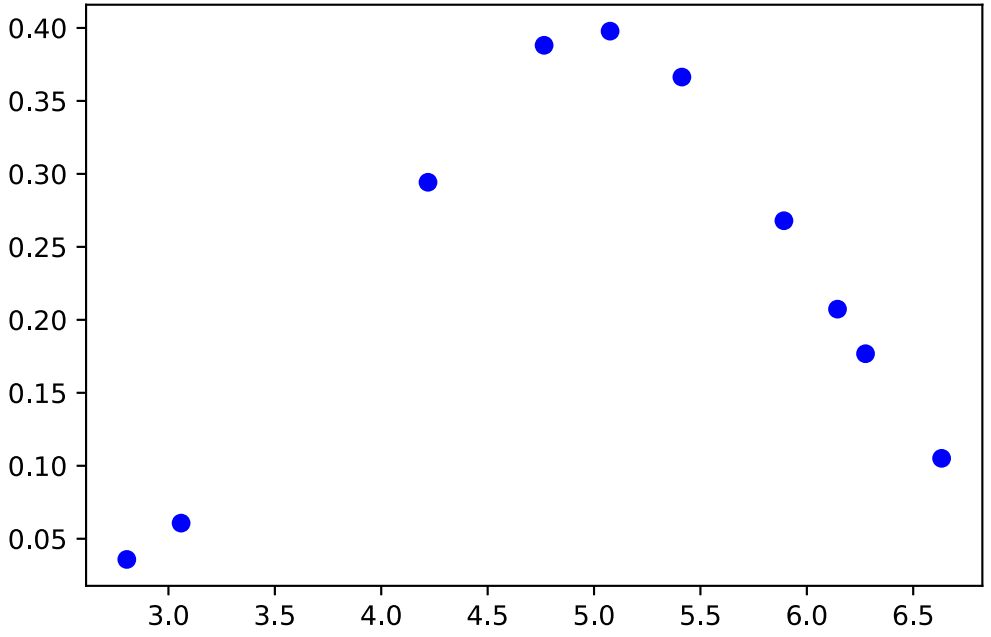
In [2]:
```
x = np.arange(0,10,0.125)

mean = 5
sigma = 1

gaussian_distribution = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x-m
ean)**2/2*(sigma**2)) )

plt.plot(x,gaussian_distribution);
```



In [7]:
```
# Selecting 1 samples of size ten from gaussian distribution(normal dis
tribution) with mean = 5 and variance = 1
g = np.random.normal(5, 1, size=(1, 10))

g
```

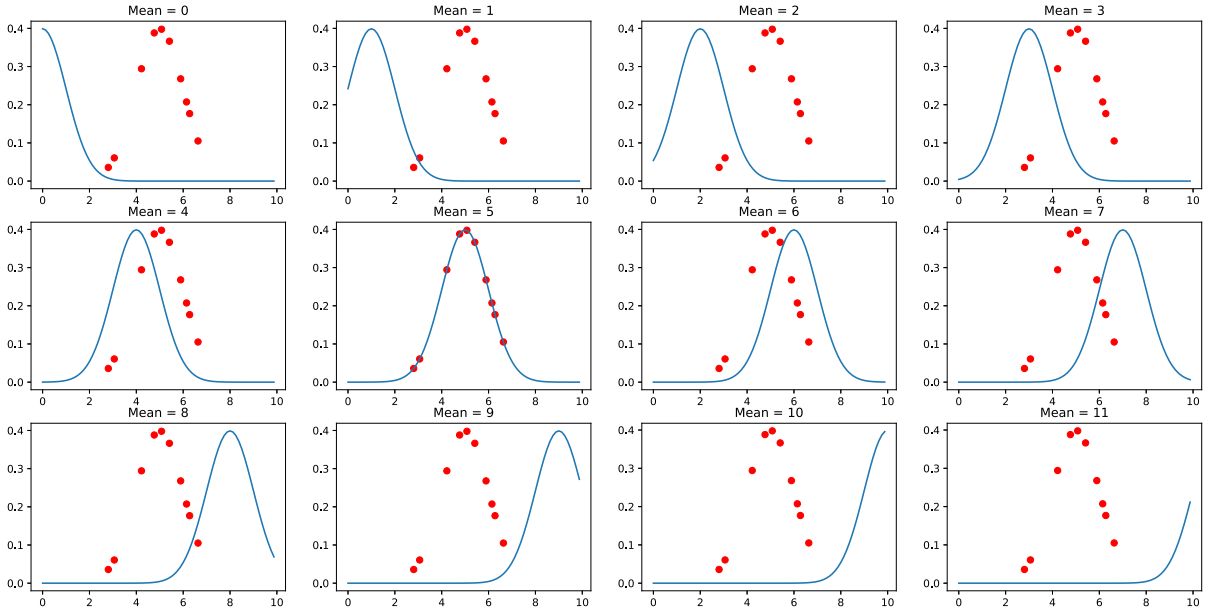
Out[7]:
```
array([[6.27586209, 5.89239374, 4.21976976, 2.80431055, 5.4128685 ,
        3.05942607, 6.63341939, 5.07548251, 6.14418367, 4.7653316 ]])
```

In [8]:
```
x = g[0]

gaussian_distribution = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x-m
ean)**2/2*(sigma**2)) )

plt.plot(x,gaussian_distribution,'bo');
```



In [9]:
```
x = g[0]
```

```
x = np.sort(x)

gaussian_distribution = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x-m
ean)**2/2*(sigma**2)) )
fig, ax = plt.subplots(3, 4, figsize=(20, 10))



mean = 0
sigma = 1

y = np.arange(0,10,0.125)


for i in range(0,3):
    for j in range(0,4):
        y_gaussian_distribution = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.e
xp( -((y-mean)**2/2*(sigma**2)) )

        ax[i , j].plot(x, gaussian_distribution,'ro')
        ax[i , j].plot(y,y_gaussian_distribution)
        ax[i , j].set_title("Mean = " + str(mean) )

        mean = mean + 1
```



In [10]:
```
p1 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[0]-mean)**2/2*(sigma
**2)) )
p2 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[1]-mean)**2/2*(sigma
**2)) )
p3 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[2]-mean)**2/2*(sigma
**2)) )
p4 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[3]-mean)**2/2*(sigma
**2)) )
p5 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[4]-mean)**2/2*(sigma
**2)) )
p6 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[5]-mean)**2/2*(sigma
**2)) )
p7 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[6]-mean)**2/2*(sigma
**2)) )
p8 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[7]-mean)**2/2*(sigma
**2)) )
p9 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[8]-mean)**2/2*(sigma
**2)) )
```

```
p10 = ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((x[9]-mean)**2/2*(sigm
a**2)) )
```

In [11]:
```
print(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10)
```

```
1.7329514182480338e-19 1.7518319008680274e-18 2.861258252382089e-14 1.7
192174017042947e-12 1.5450036091303935e-11 1.509474720907301e-10 3.1673
65548072565e-09 1.4282497521955018e-08 3.061376961201096e-08 2.22345740
142054e-07
```

In [12]:
```
p = []

for i in x:
    p.append( ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((i-mean)**2/2*
(sigma**2)) ) )
```

In [13]:
```
p
```

Out[13]:
```
[1.7329514182480338e-19,
 1.7518319008680274e-18,
 2.861258252382089e-14,
 1.7192174017042947e-12,
 1.5450036091303935e-11,
 1.509474720907301e-10,
 3.167365548072565e-09,
 1.4282497521955018e-08,
 3.061376961201096e-08,
 2.22345740142054e-07]
```

In [14]:
```
def multiplyList(myList) :

    # Multiply elements one by one
    result = 1
    for x in myList:
         result = result * x
    return result
```

In [15]:
```
likelihood = []

print(likelihood)
```

```
[]
```

In [ ]:
```

```

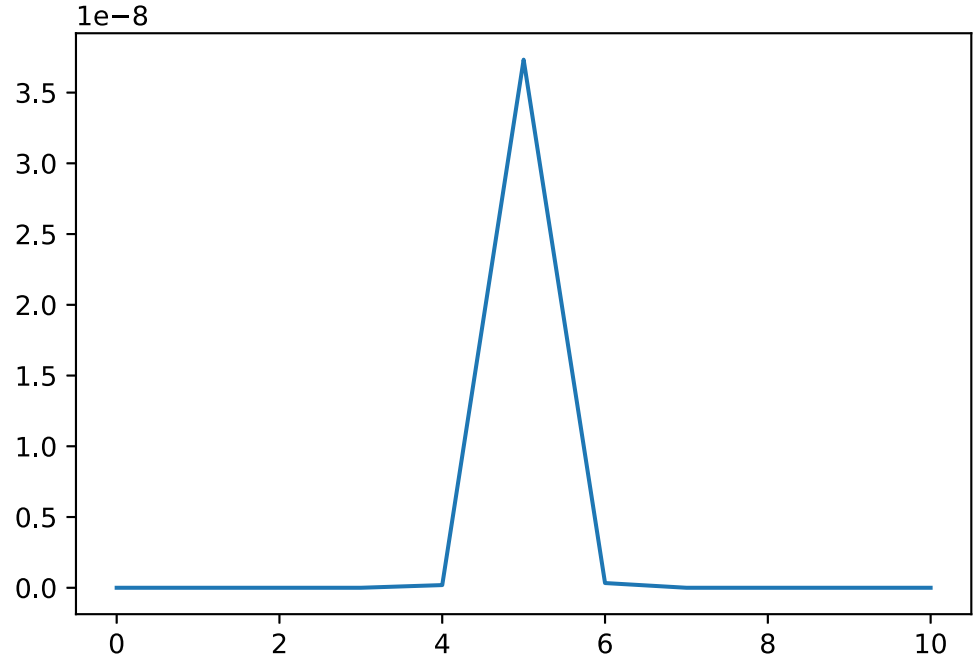
In [16]:
```
mean = 0
sigma = 1

likelihood = []

def find_likelihood(x,mu=0,sigma=1):
    p = []

    for i in x:
        p.append( ( 1/(np.sqrt(2*np.pi)*sigma) ) * np.exp( -((i-mu)**2/
2*(sigma**2)) ) )

    likelihood.append(multiplyList(p))
```

In [17]:
```
def likelihood_for_various_mean(x,means):
    for mean in means:
        find_likelihood(x,mu = mean)

    print(likelihood)
```

In [18]:
```
likelihood_for_various_mean(x,[0,1,2,3,4,5,6,7,8,9,10])
```

```
[4.683755856482879e-63, 2.171561159425227e-43, 4.570935176828576e-28,
4.368106720457617e-17, 1.8951194959442533e-10, 3.7328035923961435e-08,
3.3380194305095217e-10, 1.3551824297315384e-16, 2.4978240634519685e-27,
2.0901675886872444e-42, 7.940640854730818e-62]
```

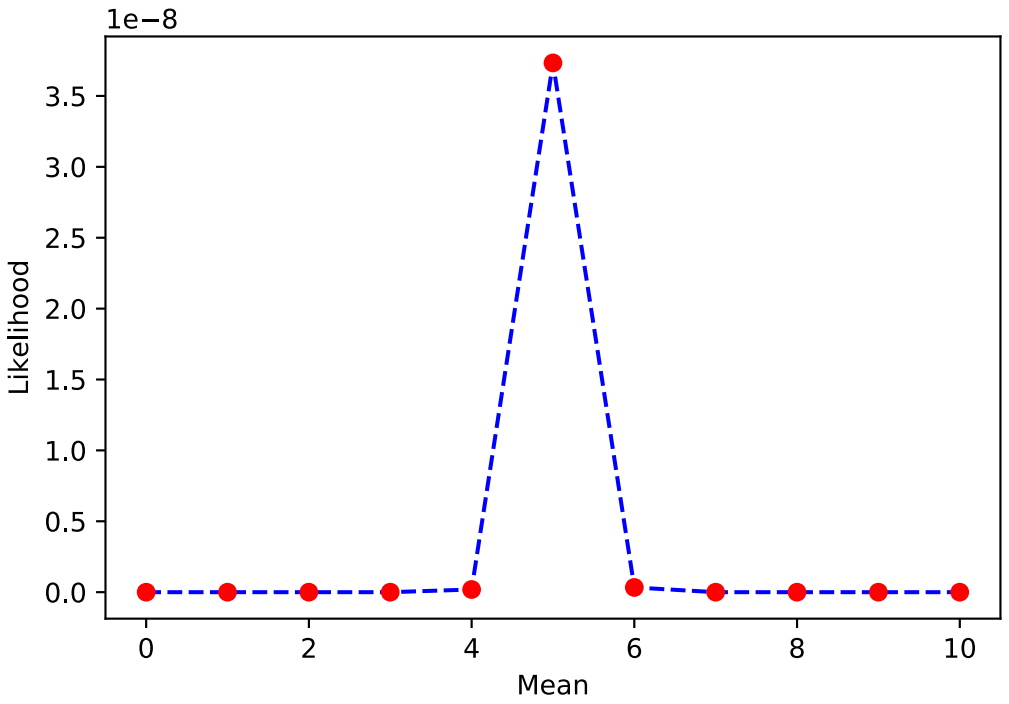
In [19]:
```
plt.plot([0,1,2,3,4,5,6,7,8,9,10] , likelihood);
```



In [20]:
```
likelihood = []

likelihood_for_various_mean(g[0],[0,1,2,3,4,5,6,7,8,9,10])

plt.plot([0,1,2,3,4,5,6,7,8,9,10] , likelihood ,'b--');
plt.plot([0,1,2,3,4,5,6,7,8,9,10] , likelihood ,'ro');
plt.xlabel('Mean')
plt.ylabel('Likelihood')
```

```
[4.683755856482878e-63, 2.171561159425227e-43, 4.570935176828577e-28,
4.368106720457617e-17, 1.8951194959442533e-10, 3.732803592396143e-08,
3.3380194305095233e-10, 1.3551824297315384e-16, 2.4978240634519677e-27,
2.0901675886872448e-42, 7.940640854730816e-62]
```

Out[20]:
```
Text(0, 0.5, 'Likelihood')
```



In [ ]:

In [ ]: