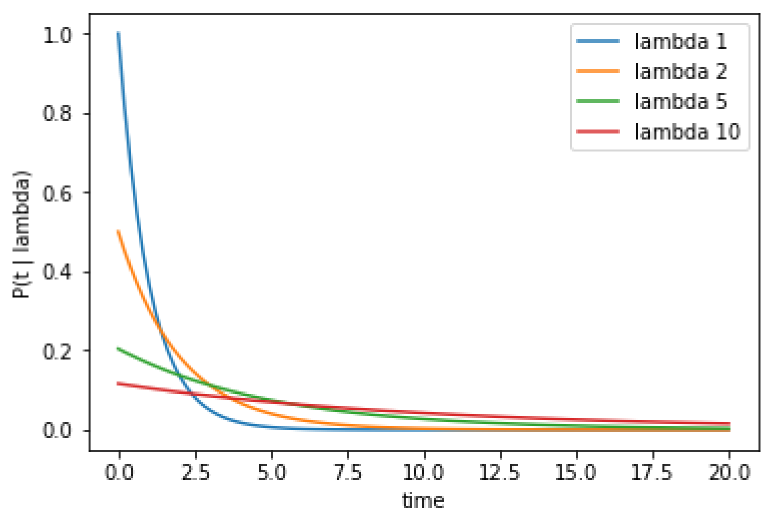
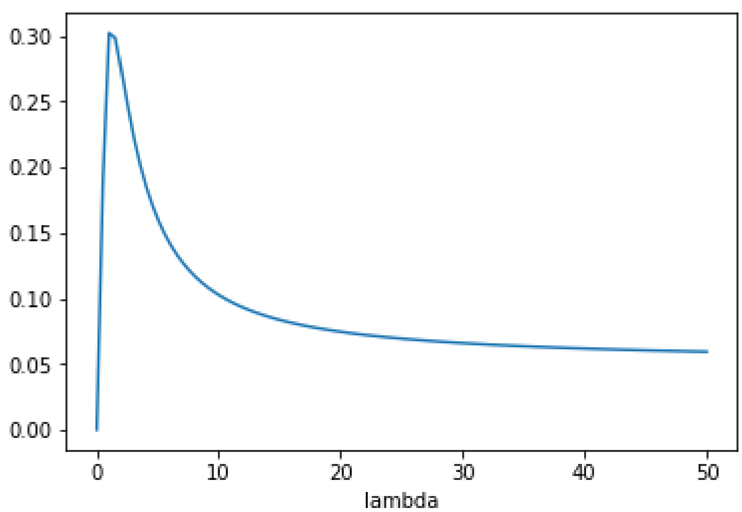
In [3]:

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

# the exponential function
#
def exponential(t, l, a, b):
    z = l * (np.exp(-a/l) - np.exp(-b/l))
    p = np.exp(-t/l)/z
    return p

a = 0
b = 20
t = np.linspace(a, b, 100)

p1  = exponential(t,  1, a, b) # lambda = 1
p2  = exponential(t,  2, a, b) # lambda = 2
p5  = exponential(t,  5, a, b) # lambda = 5
p10 = exponential(t, 10, a, b) # lambda = 10

plt.plot(t, p1,  label='lambda 1')
plt.plot(t, p2,  label='lambda 2')
plt.plot(t, p5,  label='lambda 5')
plt.plot(t, p10, label='lambda 10')
plt.legend(loc="upper right")
plt.xlabel("time")
plt.ylabel("P(t | lambda)")
plt.show()
```



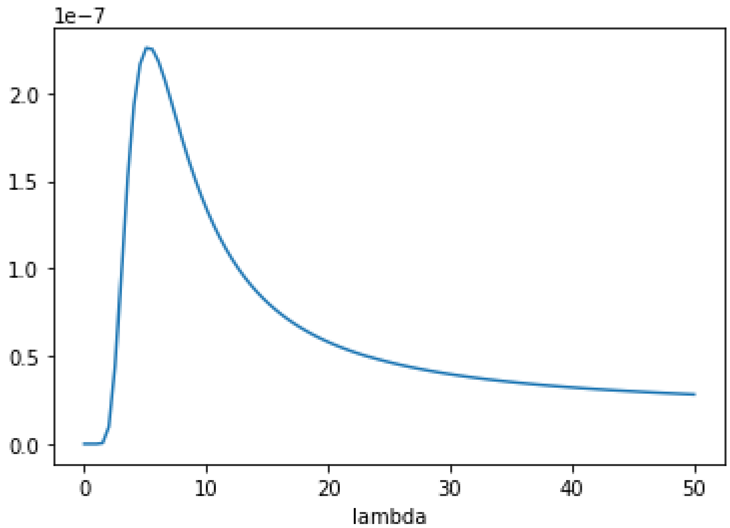
In [4]:
```
l = np.linspace(0.01, 50, 100)

# the likelihood function
#
def likelihood(l, data, a, b):
    like = []

    for x in range(len(l)):  # for each lambda, calculate the likelihood
(thislike)

        thisl = l[x] # the value of lambda

        thislike = 1 # initialzie the likelihood to 1
        for d in range(len(data)):
            thislike *= exponential(data[d], thisl, a, b)   # multiply t
he probability of all data points

        like.append(thislike) # store this likelihood
    return like

# only one data point!
# and you already get a bump!
#
data = [1.2]
print("N = ", len(data), "data mean", np.mean(data))
like = likelihood(l, data, a, b)
plt.plot(l, like)
plt.xlabel("lambda")
plt.show()
```

```
N =  1 data mean 1.2
```



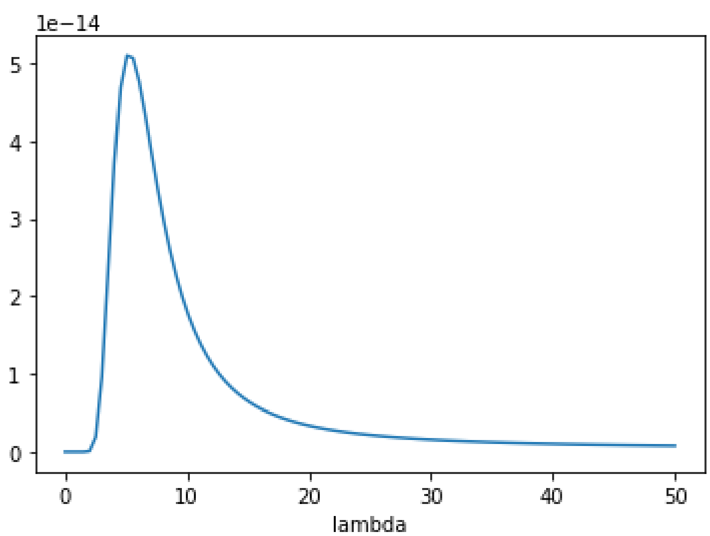
In [5]:
```
# the data in the lecture notes
#
data = [1.2, 2.1, 3.4, 4.1, 7, 11]
print("N = ", len(data), "data mean", np.mean(data))
like = likelihood(l, data, a, b)
plt.plot(l, like)
plt.xlabel("lambda")
plt.show()
```

```
N =  6 data mean 4.8
```



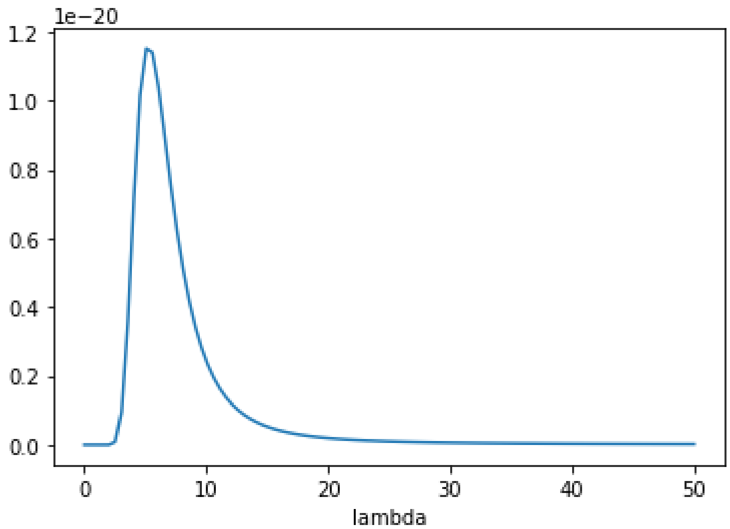
In [6]:
```
# add more data with same mean
#
data = [1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11]
print("N = ", len(data), "data mean", np.mean(data))
like = likelihood(l, data, a, b)
plt.plot(l, like)
plt.xlabel("lambda")
plt.show()
```

```
N =  12 data mean 4.8
```



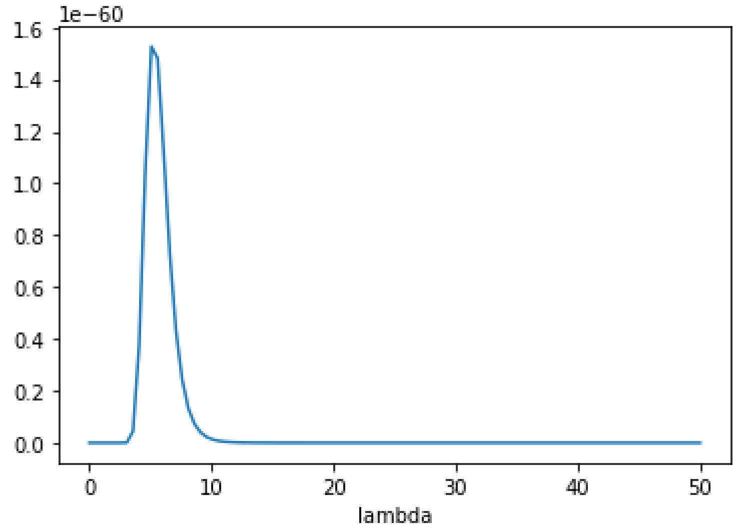
In [7]:
```
# even more data with same mean
#
data = [1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11]
print("N = ", len(data), "data mean", np.mean(data))
like = likelihood(l, data, a, b)
plt.plot(l, like)
plt.xlabel("lambda")
plt.show()
```

```
N =  18 data mean 4.8
```



In [8]:
```
# alot more data with the same mean
#
data = [1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11,
        1.2, 2.1, 3.4, 4.1, 7, 11]
print("N = ", len(data), "data mean", np.mean(data))
like = likelihood(l, data, a, b)
plt.plot(l, like)
plt.xlabel("lambda")
plt.show()
```

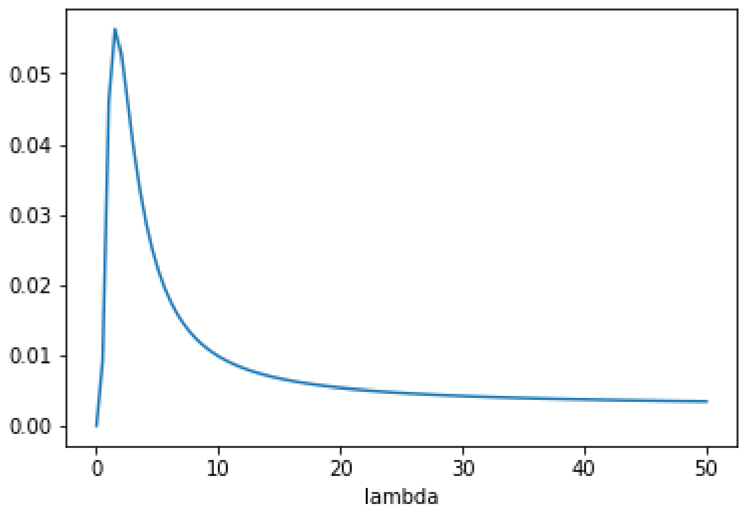
```
N =  54 data mean 4.799999999999999
```



In [17]:

```
# some weird data.
# Is the exponential distribution a good model for this data?
#
data = [0.1,3]
print("N = ", len(data), "data mean", np.mean(data))
like = likelihood(l, data, a, b)
plt.plot(l, like)
plt.xlabel("lambda")
plt.show()
```

```
N =  2 data mean 1.55
```



In [20]:
```
# what happens when you add a more informative prior for lambda
#
# Someone tells you in previous experiments the mean of the wait times f
rom previous experiments
#
prev_mean = 300
prior = exponential(l, prev_mean, a, b)
#print(prior)

data = [1.2, 2.1, 3.4, 4.1, 7, 11, 1.2, 2.1, 3.4, 4.1, 7, 11]
print("N = ", len(data), "data mean", np.mean(data))
like = likelihood(l, data, a, b)

both = like*prior

plt.plot(l, prior)
plt.title("Prior distribution")
plt.xlabel("lambda")
plt.show()

plt.plot(l, like)
plt.title("Likelihood = probability of data given lambda")
plt.xlabel("lambda")
plt.show()

plt.plot(l, both)
plt.title("Lkelihood * Prior")
plt.xlabel("lambda")
plt.show()
```

```
N =  12 data mean 4.8
```