In [4]:
```
# !pip install numpy
# !pip install matplotlib
# !pip install sklearn
# !pip install scipy
```

In [5]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
print ("complete")
```

```
complete
```

Analysis methodology ¶

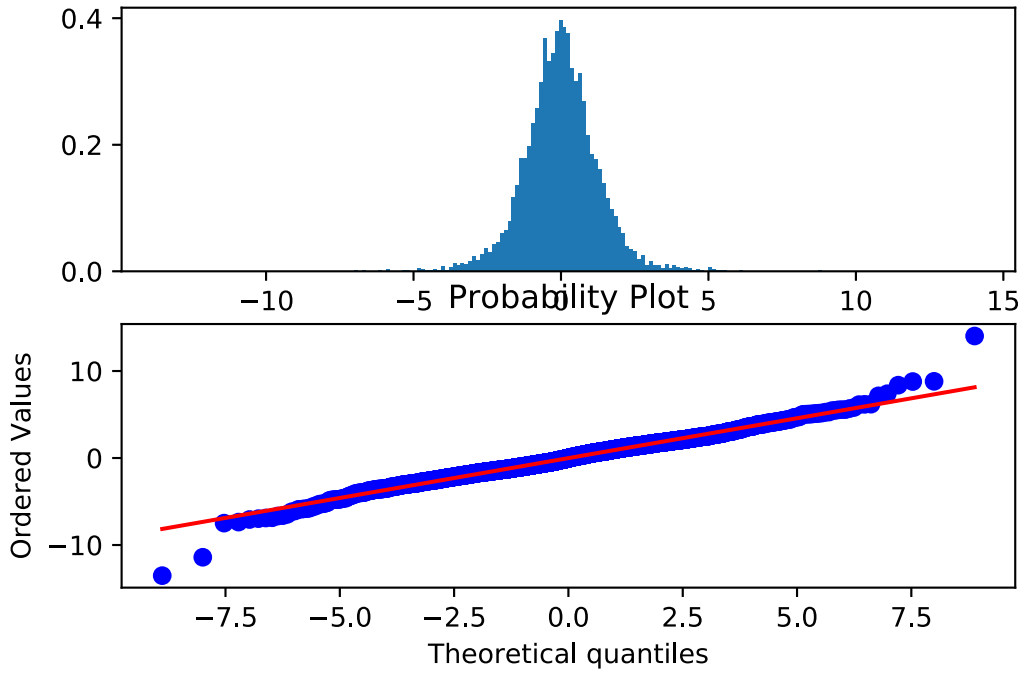
The main process for finding the distribution is to use the scipy probplot function to plot each data set against a probability ditribution to find out which distibution the data set most likly fits. Using this method allows you to use the line of best fit to model the data and see which distribution is the most linear for the given data set.

In [6]:
```
file = "distA.csv"
data = np.loadtxt(file)
print (data)

fig = plt.figure()
ax1 = plt.subplot(211)
hist = ax1.hist(data, bins="auto", density= True)

ax2 = plt.subplot(212)
#datavals = stats.probplot(data, plot=ax2, dist = "expon")
#datavals = stats.probplot(data, plot=ax2, dist = "uniform")
datavals = stats.probplot(data, plot=ax2, dist = "laplace")
plt.show()
```

```
[-1.3173  -0.58993 -0.18312 ... -0.59624  0.81575  2.0798 ]
```



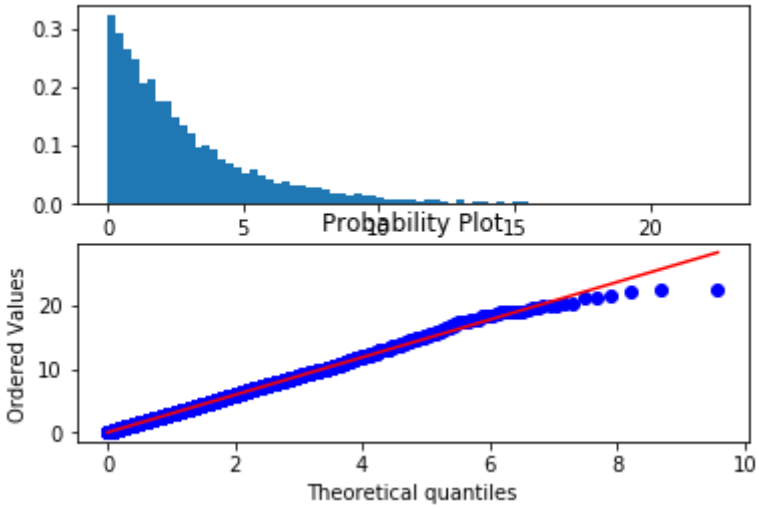
DistA analysis

- in this case, the data set most closely fits the laplace distibution
- laplace is similar to the gaussian in shape, but the curve is far more pointy
- i tried the exponential, gausian, and the uniform before settling on laplace

In [6]:
```
file2 = "distB.csv"
data2 = np.loadtxt(file2)

fig2 = plt.figure()
ax1 = plt.subplot(211)
hist2 = ax1.hist(data2, bins="auto", density= True)

ax2 = plt.subplot(212)
datavals = stats.probplot(data2, plot=ax2, dist ='expon')
plt.show()
```



DistB analysis

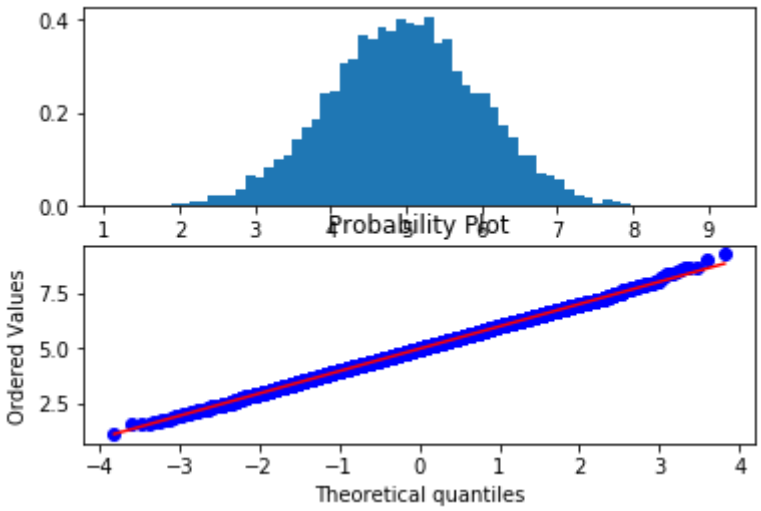
- the exponential distribution best fits the data
- to find this i notices that the data curve unlike other distibutions is squished to the left.
- tracing the tops of the bars shoes a curve that most closely represents an exp(-x) leading to the assumption that that data is of the exponential distribution.
- i did try some of the others like the coshy, and the cosine ditribution before setting on exponential

In [7]:
```
file3 = "distC.csv"
data3 = np.loadtxt(file3)
print (data3)

fig3 = plt.figure()
ax1 = plt.subplot(211)
hist3 = ax1.hist(data3, bins="auto", density= True)

ax2 = plt.subplot(212)
datavals = stats.probplot(data3, plot=ax2)
plt.show()
```

```
[4.9372 4.2752 4.3512 ... 5.1363 4.1114 6.6454]
```



DistC analysis

- after Dist A, I was aware that this data set was either laplace of Gaussian
- probplots automatically defaults to the normal or gaussian distribution if no alternative is provided.
- the most linear of the 2, laplace and Gaussian was the Gaussian distribution.