In [1]:

```
%matplotlib inline
```

In [3]:

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

In [5]:

```
np.random.lognormal()
```

Out[5]:

```
4.074617245102864
```

In [129]:

```
x = np.linspace(1., 5., 10000)
```

In [130]:

```
mu, sigma = 1., 1.
```

In [138]:

```
pdf_log_norm = np.exp(-(np.log(x) - mu)**2 / (2 * sigma**2))/ (x  * np.sqrt(2 * np.pi*
sigma**2))
```

In [132]:

```
x_m, alpha = 1.0, 1.0
```

In [136]:

```
x_p = np.linspace(1., 5., 10000)
pdf_pareto = (alpha*(x_m**alpha)) / (x_p**(alpha+1))
```

In [134]:

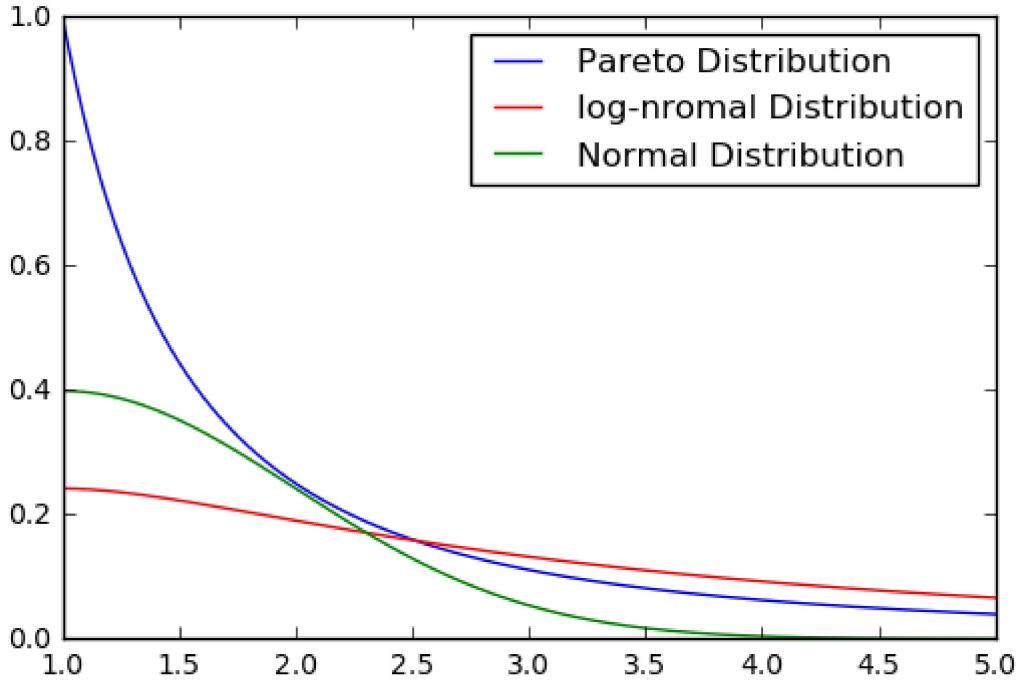
```
pdf_normal = np.exp((-(x-mu)**2)/(2*sigma**2))/np.sqrt(2 * np.pi * sigma**2)
```

In [139]:

```
plt.plot(x_p, pdf_pareto, color='b', label='Pareto Distribution')
plt.plot(x, pdf_log_norm, color='r', label='log-nromal Distribution')
plt.plot(x, pdf_normal, color='g', label='Normal Distribution')
plt.legend()
```

Out[139]:

```
<matplotlib.legend.Legend at 0x7f59aaf9d290>
```

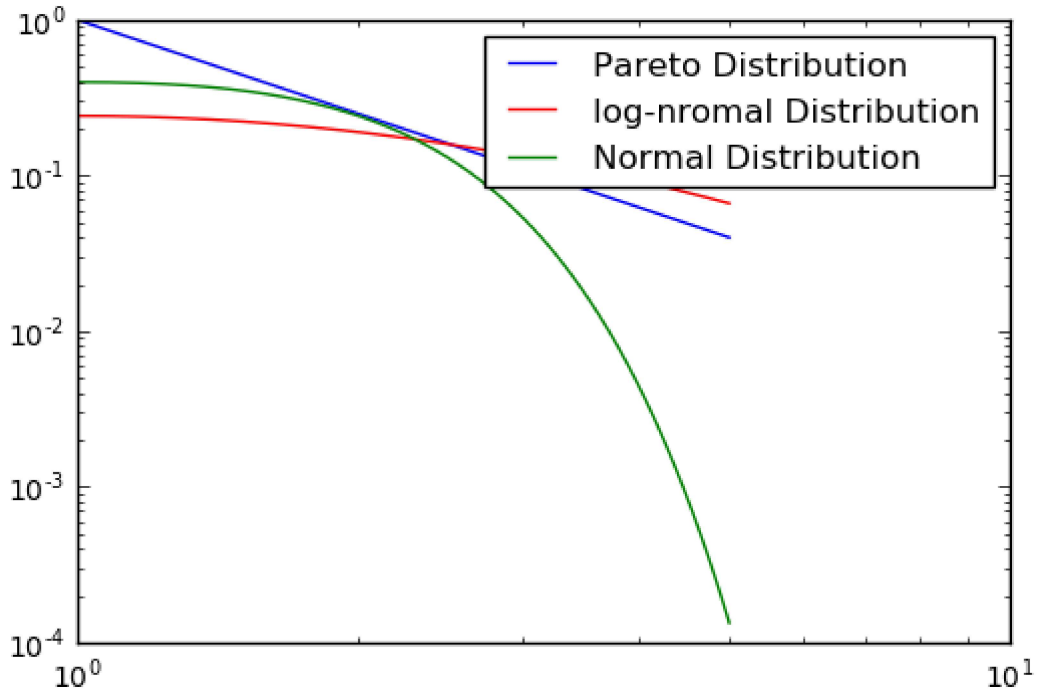


In [94]:

```
plt.plot(x_p, pdf_pareto, color='b', label='Pareto Distribution')
plt.plot(x, pdf_log_norm, color='r', label='log-nromal Distribution')
plt.plot(x, pdf_normal, color='g', label='Normal Distribution')
plt.xscale('log')
plt.yscale('log')
plt.legend()
```

Out[94]:

```
<matplotlib.legend.Legend at 0x7f59abb9b9d0>
```



In [122]:

```
x = np.linspace(1., 10., 10000)
```

In [123]:

```
alpha_2 = [0, 0.2, 0.5, 2, 5, 100,1]
```

In [124]:

```
plot = []
for i in alpha_2:
    plot.append((x**(1-i))/(1-i))
```

In [125]:

```
plot.append(np.log(x))
```