Data Analysis Timber Strength n=164

In this notebook we are evaluating the optimal distribution fitting the data. At first, we import a set of libraries

In [1]:
```
import OpenAIUQ as auq
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import pearsonr
#from openseespy.postprocessing.Get_Rendering import *

matplotlib.rcParams.update({'font.size':16})
```

Problem Definition

Timber strength [MPa]									
48.78	32.02	45.54	32.40	48.37	50.98	35.58	40.53	29.11	65.35
41.64	39.34	34.12	33.06	29.93	40.71	28.97	47.25	65.61	45.19
39.77	46.33	45.92	33.47	36.38	34.63	34.56	32.68	37.78	70.22
35.89	46.99	36.47	35.67	46.86	24.84	26.89	43.26	43.33	41.75
54.04	22.67	28.98	28.46	36.00	28.83	38.64	47.61	53.63	37.51
35.43	39.62	40.85	23.16	23.19	42.31	24.25	28.13	41.85	31.60
22.75	44.78	56.60	44.51	36.88	39.33	44.54	32.48	33.19	37.65
44.78	26.63	28.76	42.47	44.30	39.93	40.85	36.81	39.15	28.00
43.99	43.48	47.42	48.39	44.59	39.60	39.97	35.88	54.71	46.01
47.74	30.05	33.61	38.05	44.00	38.16	37.69	33.92	43.64	43.48
25.39	30.33	44.36	35.03	40.39	43.33	41.78	57.99	56.80	40.27
38.00	39.21	35.30	31.33	41.72	69.07	33.14	49.57	43.07	39.05
25.98	51.39	33.18	27.31	29.90	51.90	55.23	40.20	43.12	32.76
36.84	50.91	36.85	53.99	35.17	33.71	36.53	49.59	30.02	45.97
34.49	49.65	17.98	43.41	34.44	46.50	22.74	32.03	38.81	23.14
38.71	47.83	27.90	28.71	27.93	36.92	34.40	39.20	24.09	53.00
30.53	44.07	44.36	58.34						

We have a dataset of 164 data of timber strength. We want to evaluate the statistics of the data.

Import Data

First, we import the dataset represent 164 data of Timber strength collected in the file 'timber164.dat'. After we import the dataset, we evaluate:

- min value
- max value
- range

In [ ]:
```
auq?
```

In [ ]:
```
auq.data1?
```

In [2]:

```
#========================================
#TIMBER 164
#========================================
dataset=np.loadtxt('timber164.dat')

#d is the object collecting the dataset
d=auq.data1(dataset)
data=d.data

print('Data analysis')
print('min value: {:.4}'.format(d.datamin))
print('max value: {:.4}'.format(d.datamax))
print('range: {:.4}'.format(d.r))
```

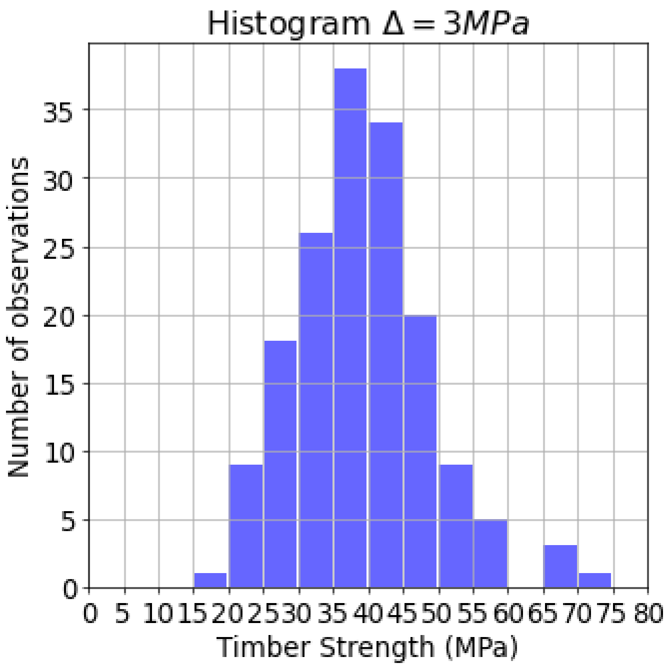
```
Data analysis
min value: 17.98
max value: 70.22
range: 52.24
```

Histogram

Let's build an histogram to analyze the data:

In [3]:

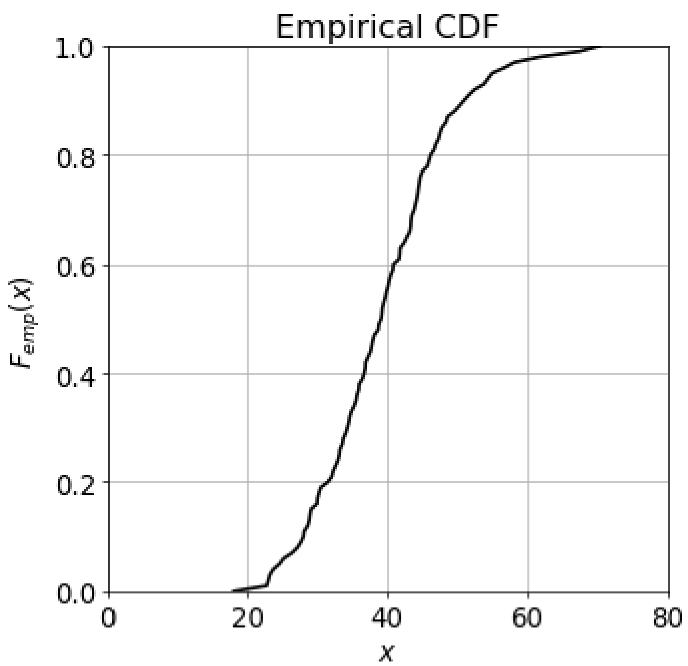
```
#Choose binning
#bin1=[23,26,29,32,35,38,41]
#bin1='auto'
bin1=np.arange(0,85,5)

fig1=plt.figure(num=1,figsize=(6,6),dpi=60)
#num: figure number
#figsize: fugure dimension
#dpi

ax=fig1.add_subplot(1,1,1)
#plot=1 row, 1 column

n,bin,patches=ax.hist(d.data, bins=bin1, histtype='bar',facecolor='blue', density=Fa
#bins: limits of the bins
#facecolor: color of the bar
#density: False, True
#alpha: transparency
#rwidth: distance between histograms

ax.set_title('Histogram $\Delta=3 MPa$')
ax.set_xlabel('Timber Strength (MPa)')
ax.set_ylabel('Number of observations')
ax.grid()
ax.set_xlim(0,80)
ax.set_xticks(bin1);
```



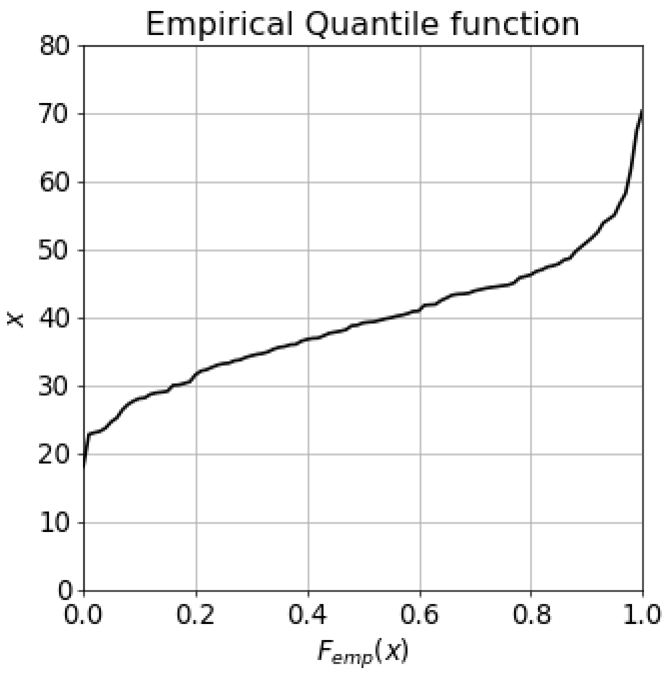
Cumulative Relative frequency

In [ ]:
```
auq.data1.get_quantile?
```

In [4]:
```
d.set_range(xiniz=0,xfin=80)
#set the range of interest of the variable

d.get_cdf_emp()
#evaluate the empirical cdf

d.plot_cdf_emp(fignum=2,figsize=(6,6),figdpi=60)
#plot the empirical cdf
```



In [5]:
```
d.plot_quantile_emp(fignum=3,figsize=(6,6),figdpi=60)
#plot the empirical quantile function
```



Metrics

Now we evaluate some metrics:

- sample mean
- sample standard deviation
- sample coefficient of variation
- sample skewness
- sample kurtosis

In [6]:
```
print('Data analysis')
print('Mean: {:.4} MPa'.format(d.mean))
print('Standard deviation: {:.3} MPa'.format(d.std))
print('Coefficient of variation: {:.3}'.format(d.cov))
print('Skewness: {:.3}'.format(d.g1))
print('Kurtosis: {:.3}'.format(d.g2))
```

```
Data analysis
Mean: 39.32 MPa
Standard deviation: 9.44 MPa
Coefficient of variation: 0.24
Skewness: 0.536
Kurtosis: 3.61
```

The coefficient of variation ν shows a relatively high value of spreading around the mean value. This is expected for a natural material like the timber

The low skewness γ_1 shows a degree of symmetry of the data around the mean value. But the number is data is too small to have a reliable estimate of skewness.

The kurtosis value γ_2 shows that the tails are different from the Gaussian. But the number is data is too small to have a reliable estimate of kurtosis.

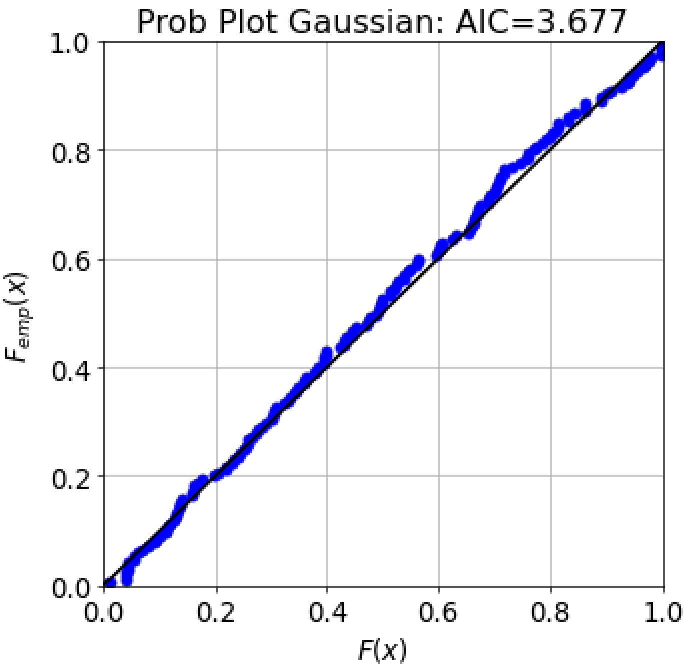
In [ ]:
```
auq.dist?
```

In [ ]:
```
auq.dist.MLEfit?
```

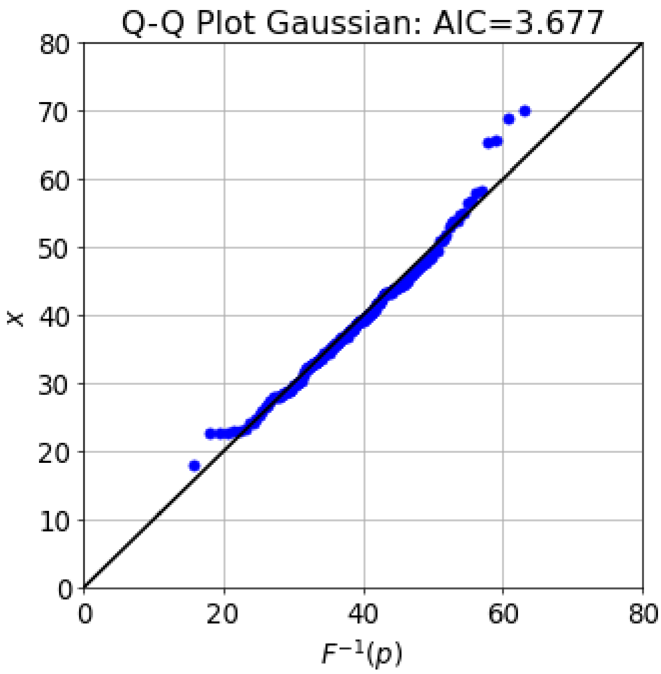
Gaussian fit

In [7]:
```
x1=auq.dist('Gaussian')
#Gaussian, Lognormal, Weibull, Uniform

x1.set_range(xiniz=0,xfin=80,size=100)
#xiniz: lower bound
#xfin: Upper bound
#size: number of points

x1.MLEfit(d.data)
#Evaluate the parameters of x through MLE
```

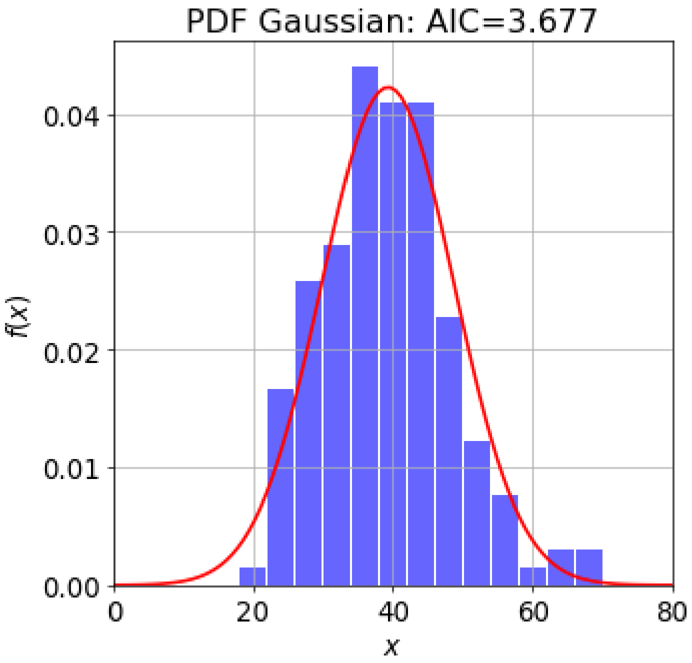
In [8]:
```
fig=auq.distplot(num=1,figsize=(6,6),dpi=60)
#fig is the object associated to the distribution plot

fig.prob_plot(d.data,x1)
```



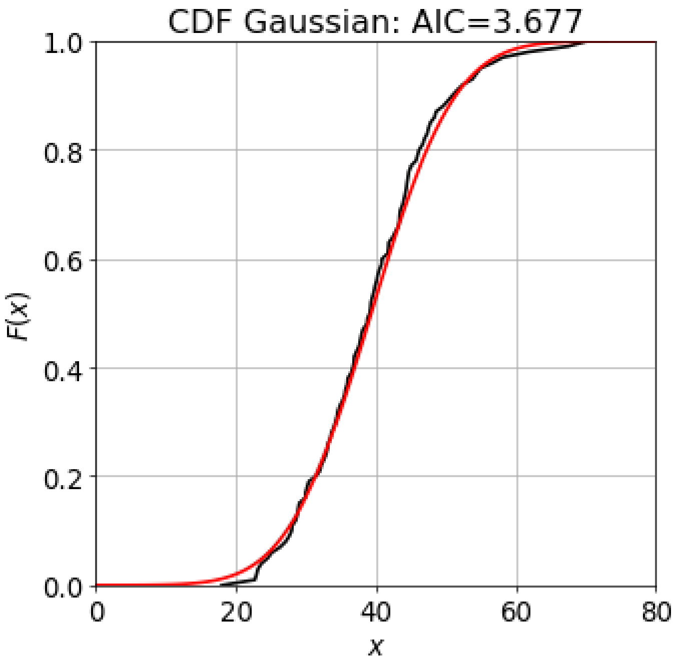
In [9]:
```
fig.qq_plot(d.data,x1)
```



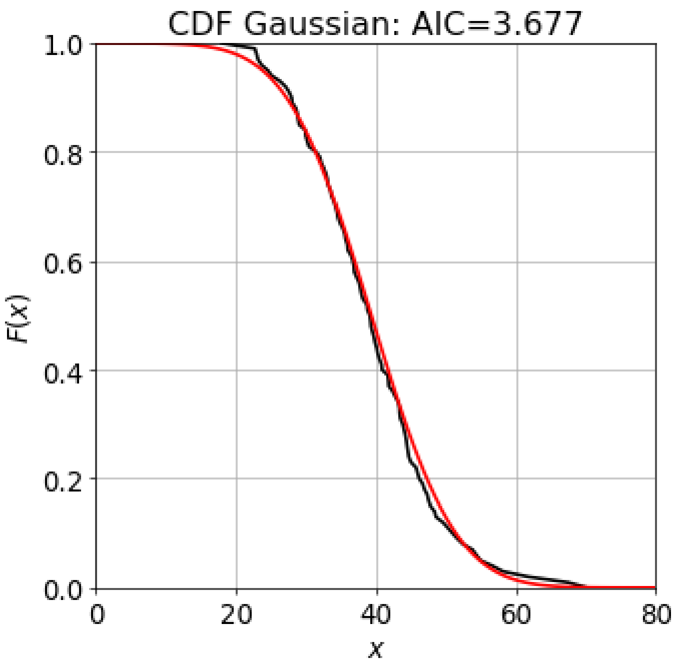
In [10]:
```
fig.pdf_plot(d.data,x1)
```



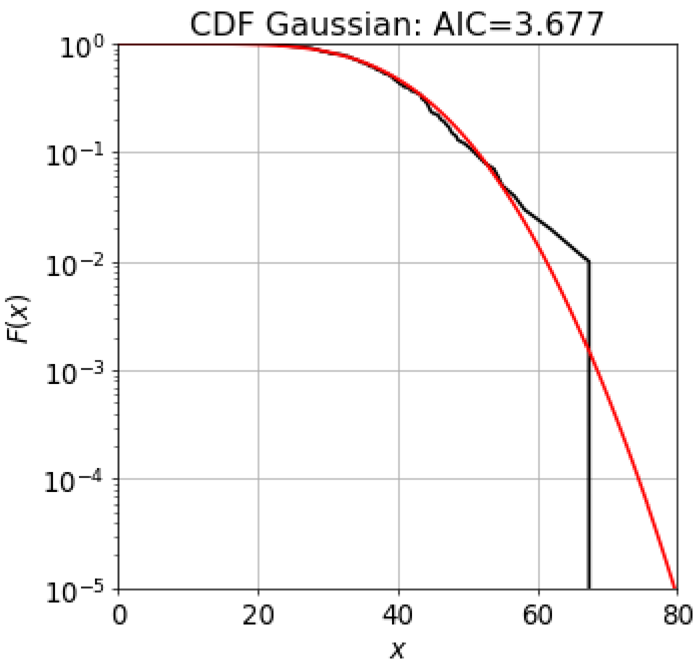
In [11]:
```
fig.cdf_plot(d.data,x1)
```



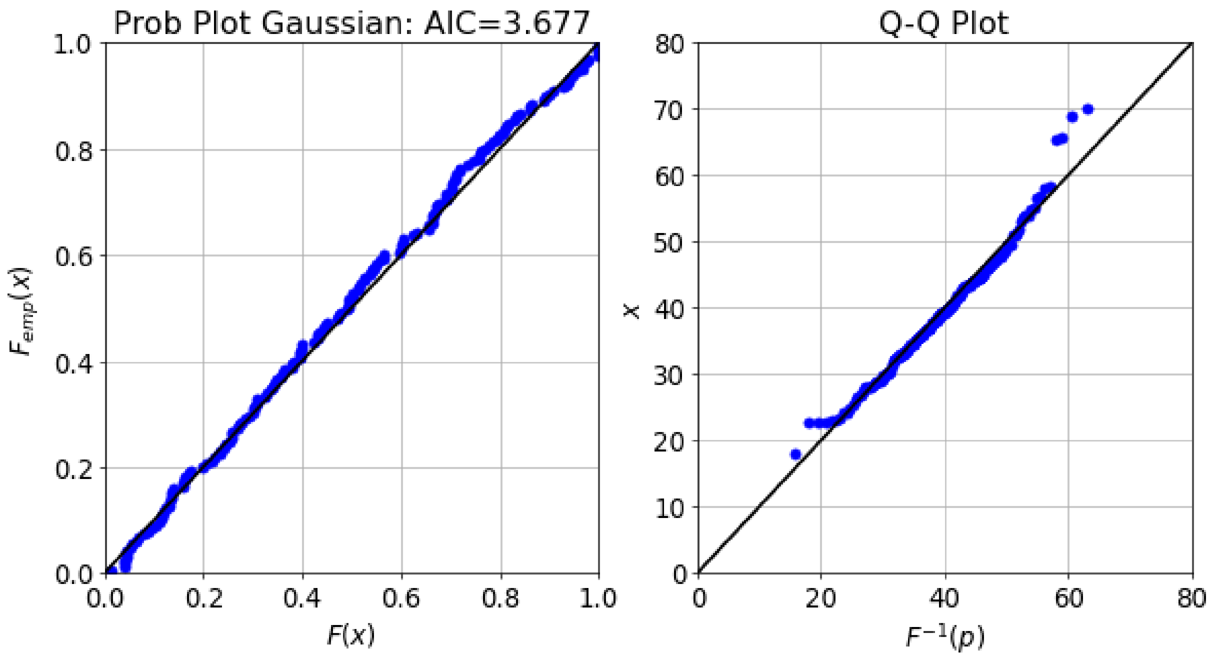
In [12]:
```
fig.poe_plot(d.data,x1)
```



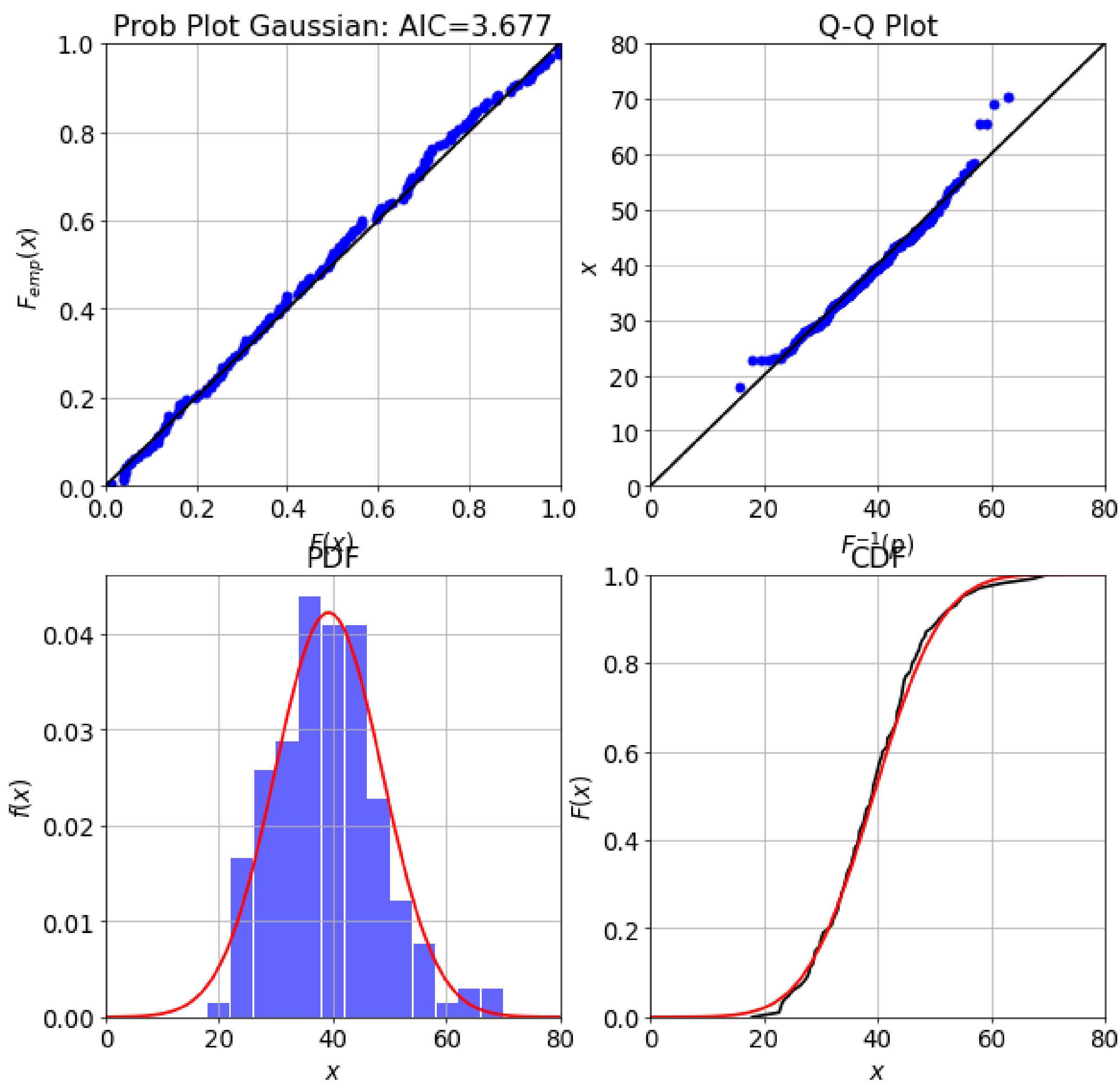
In [13]:
```
fig.poe_plot(d.data,x1,logscale='yes')
```



In [14]:
```
fig=auq.distplot(num=1,figsize=(12,6),dpi=60)
fig.prob_plot_w2(d.data,x1)
```

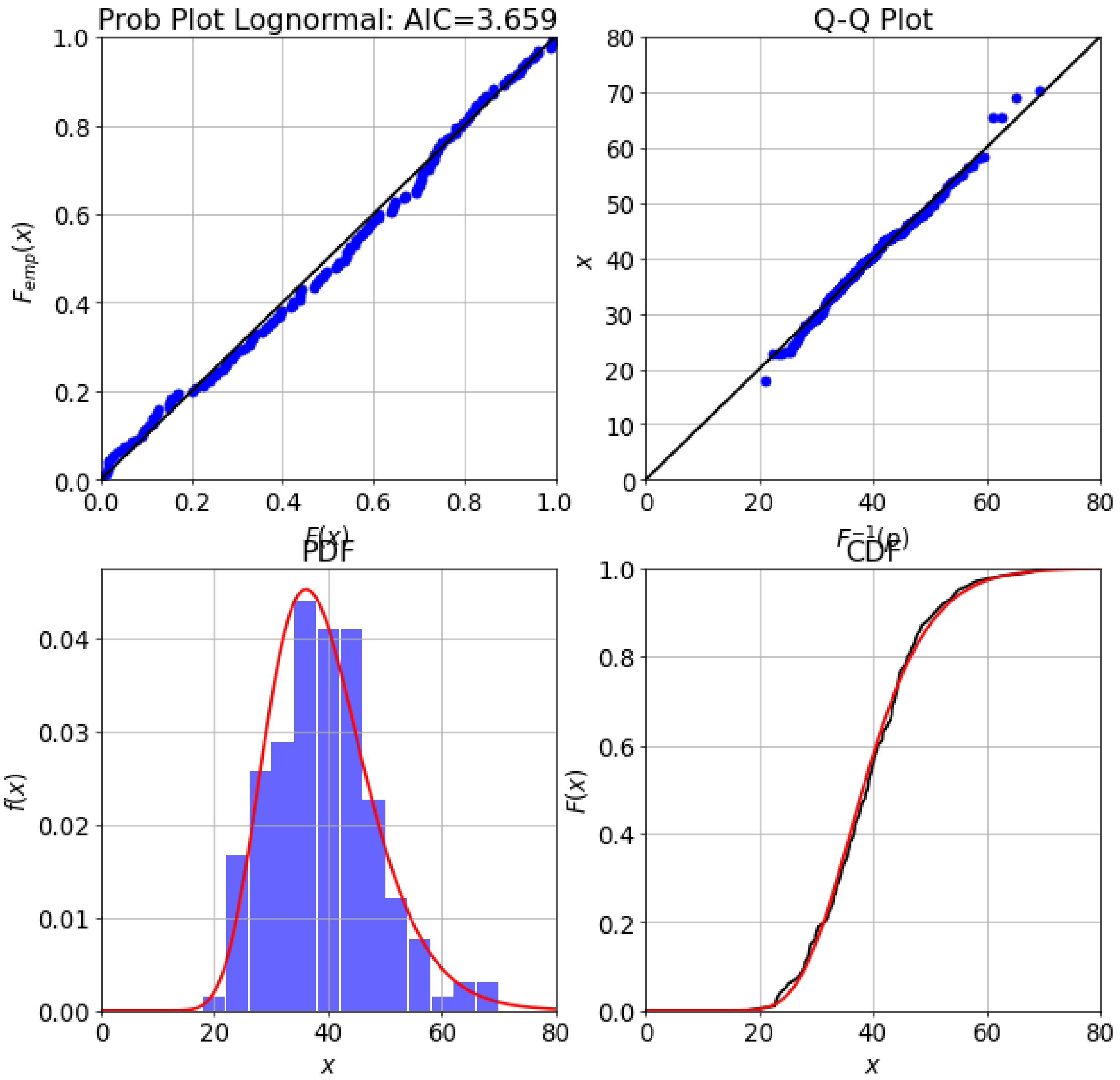


In [15]:
```
fig=auq.distplot(num=1,figsize=(12,12),dpi=60)
fig.prob_plot_w4(d.data,x1)
```



Lognormal fit

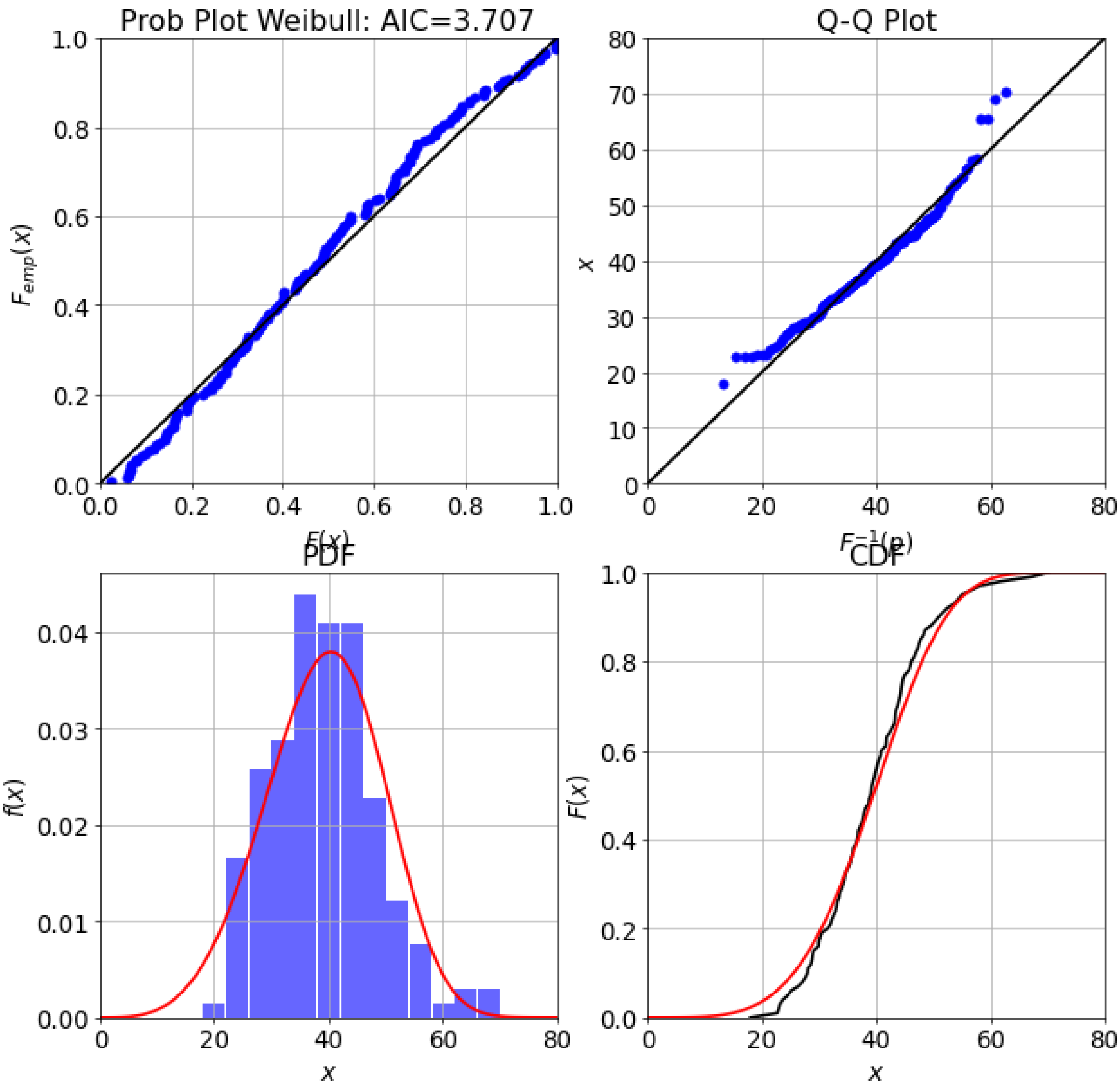
In [16]:
```
x2=auq.dist('Lognormal')
#Gaussian, Lognormal, Weibull, Uniform

x2.set_range(xiniz=0,xfin=80,size=100)
#xiniz: Lower bound
#xfin: Upper bound
#size: number of points

x2.MLEfit(d.data)
#Evaluate the parameters of x through MLE

fig=auq.distplot(num=2,figsize=(12,12),dpi=60)
#fig is the object associated to the distribution plot
fig.prob_plot_w4(d.data,x2)
```



Weibull fit

In [17]:
```
x3=auq.dist('Weibull')
#Gaussian, Lognormal, Weibull, Uniform

x3.set_range(xiniz=0,xfin=80,size=100)
#xiniz: Lower bound
#xfin: Upper bound
#size: number of points

x3.MLEfit(d.data)
#Evaluate the parameters of x through MLE
```

```python
fig=auq.distplot(num=3,figsize=(12,12),dpi=60)
#fig is the object associated to the distribution plot
fig.prob_plot_w4(d.data,x3)
```




```
In [ ]:
```