0120_generate_log_normal_samples

February 7, 2019

1 Generating log normal samples from provided arithmetic mean and standard deviation of original population

The log normal distribution is frequently a useful distribution for mimicking process times in healthcare pathways (or many other non-automated processes). The distribution has a right skew which may frequently occur when some clinical process step has some additional complexity to it compared to the 'usual' case.

To sample from a log normal distribution we need to convert the mean and standard deviation that was calculated from the original non-logged population into the mu and sigma of the underlying log normal population.

(For maximum computation effuiciency, when calling the function repeatedly using the same mean and standard deviation, you may wish to split this into two functions - one to calculate mu and sigma which needs only calling once, and the other to sample from the log normal distribution given mu and sigma).

In [1]:
```
import numpy as np

def generate_lognormal_samples(mean, stdev, n=1):
    """
    Returns n samples taken from a lognormal distribution, based on mean and
    standard deviation calaculated from the original non-logged population.

    Converts mean and standard deviation to underlying lognormal distribution
    mu and sigma based on calculations desribed at:
        https://blogs.sas.com/content/iml/2014/06/04/simulate-lognormal-data-
        with-specified-mean-and-variance.html

    Returns a numpy array of floats if n > 1, otherwise return a float
    """

    # Calculate mu and sigma of underlying lognormal distribution
    phi = (stdev ** 2 + mean ** 2) ** 0.5
    mu = np.log(mean ** 2 / phi)
    sigma = (np.log(phi ** 2 / mean ** 2)) ** 0.5

    # Generate lognormal population
    generated_pop = np.random.lognormal(mu, sigma , n)
```

```
    # Convert single sample (if n=1) to a float, otherwise leave as array
    generated_pop = \
        generated_pop[0] if len(generated_pop) == 1 else generated_pop

    return generated_pop
```

1.1 Test the function

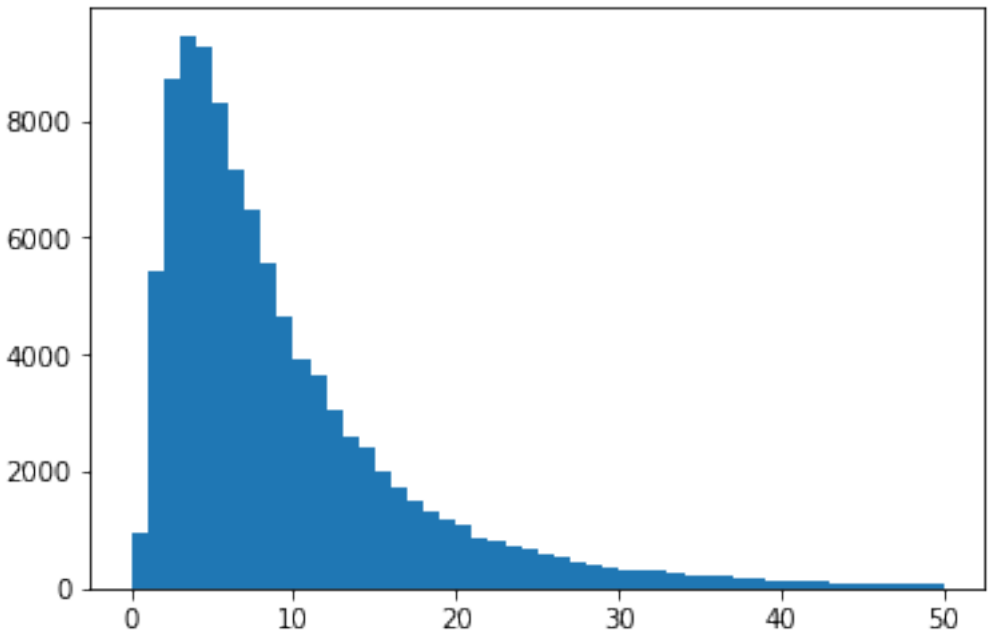
We will generate a population of 100,000 samples with a given mean and standard deviation (these would be calculated on the non-logged population), and test the resulting generated population has the same mean and standard deviation.

```
In [2]: mean = 10
        stdev = 10
        generated_pop = generate_lognormal_samples(mean, stdev, 100000)
        print ('Mean:', generated_pop.mean())
        print ('Standard deviation:', generated_pop.std())
```

```
Mean: 10.043105926813356
Standard deviation: 9.99527575740651
```

Plot a histogram of the generated population:

```
In [3]: import matplotlib.pyplot as plt
        %matplotlib inline
        bins = np.arange(0,51,1)
        plt.hist(generated_pop, bins=bins)
        plt.show()
```



1.2 Generating a single sample

The function will return a single number if no `n` is given in the function call:

```
In [4]: print (generate_lognormal_samples(mean, stdev))
```

```
6.999376449335125
```