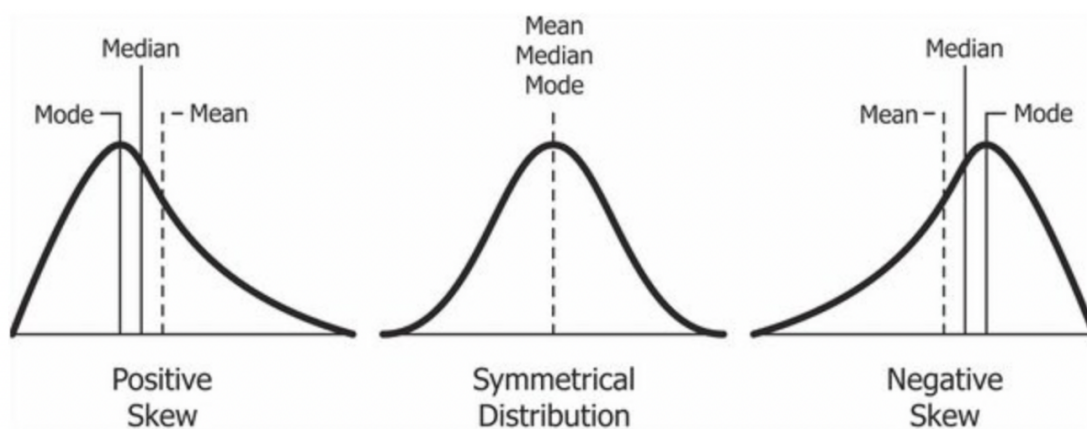


## Exercise : Skewness

Skewness, in statistics, is the degree of asymmetry observed in a probability distribution. Distributions can exhibit right (positive) skewness or left (negative) skewness to varying degrees. A normal distribution (bell curve) exhibits zero skewness. For negative value of skewness, it gives left skewed and for positive values it gives right skewed.



Credit : wikipedia

In python, using **skewnorm** from **scipy.stats**, we can use skewed distribution.

### Left Skewed Distribution(in Python)

```
from scipy.stats import skewnorm
import matplotlib.pyplot as plt

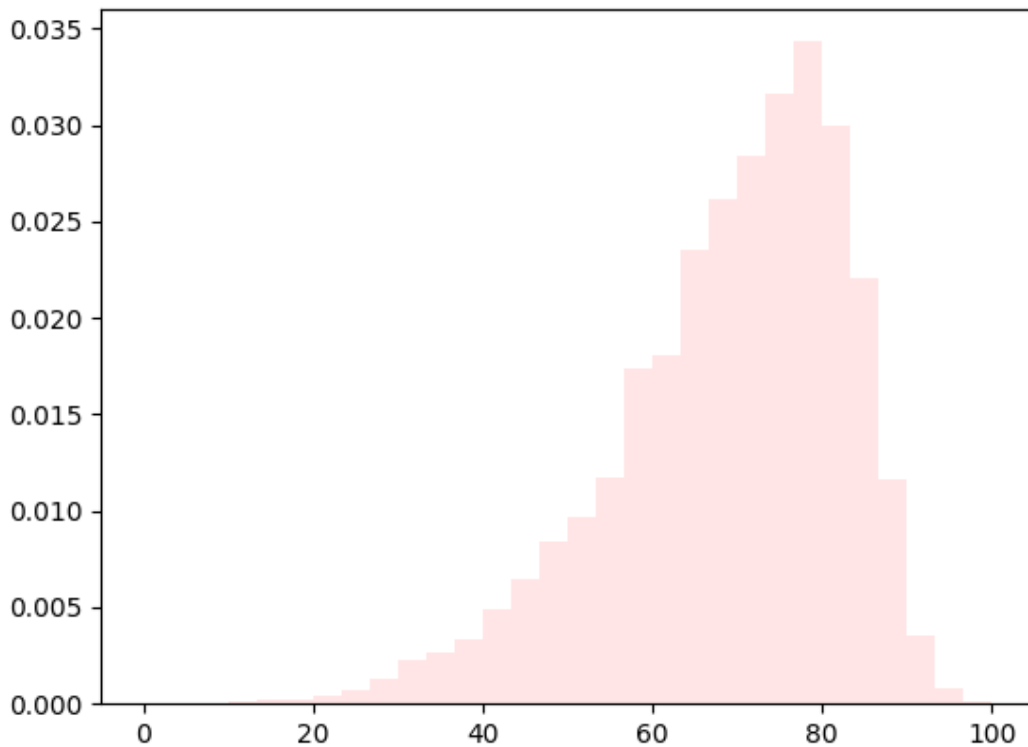
numValues = 10000
maxValue = 100
skewness = -5 # Negative values are left skewed, positive values
               are right skewed.

random = skewnorm.rvs(a=skewness, loc=maxValue, size=numValues) #
Skewnorm function

random = random - min(random) # Shift the set so the minimum value
                               is equal to zero.
random = random / max(random) # Standardize all the vlues between 0
                               and 1.
random = random * maxValue    # Multiply the standardized values by
```

*the maximum value.*

```
# Plot histogram to check skewness
plt.hist(random, 30, density=True, color='red', alpha=0.1)
plt.show()
```



## Right Skewed Distribution(in Python)

```
from scipy.stats import skewnorm
import matplotlib.pyplot as plt

numValues = 10000
maxValue = 100
skewness = 5 # positive values are right skewed.

random = skewnorm.rvs(a=skewness, loc=maxValue, size=numValues) #
Skewnorm function

random = random - min(random) # Shift the set so the minimum value
is equal to zero.
random = random / max(random) # Standadize all the vlues between 0
and 1.
random = random * maxValue # Multiply the standardized values by
the maximum value.
```

```
# Plot histogram to check skewness  
plt.hist(random, 30, density=True, color='red', alpha=0.1)  
plt.show()
```

