Skewness and Kurtosis

We want to show the structure of simple functions in R and Python. To illustrate how to create those functions (without exceptions or testing for data type), we will implement the formulas for skewness and kurtosis of a random sample and compare with inbuilt functions. The skewness, which is a measure of asymmetry, is defined as:

$$Skewness = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^{(3/2)}}$$

The kurtosis is a measure of 'tailedness' (how fat a distribution is) and is defined as:

$$Kurtosis = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2}$$

Skewness and Kurtosis in R

```
library(moments)

# generate a vector x of Normal data
x = rnorm(100, 4, 1.5)

# skewness and kurtosis functions
myskewness=function(x) {
  m = mean(x)
  diff = (x-m)
  num = mean(diff^3)
  den = (mean(diff^2))^(3/2)
  num/den
}

mykurtosis = function(x) {
  m = mean(x)
  diff = (x-m)
  num = mean(diff^4)
  den = mean(diff^2)^2
  (num/den)
}

myskewness(x) # [1] 0.03989365
skewness(x) # [1] 0.04009463
mykurtosis(x) # [1] 2.693733
kurtosis(x) # [1] 2.693733
```

Skewness and Kurtosis in Python

```python
import numpy as np
from scipy.stats import skew, kurtosis

# Generate a vector x of Normal data
x = np.random.normal(4, 1.5, 100)

# Skewness and Kurtosis function
def myskewness(x):
    m = np.mean(x)
    diff = x - m
    num = np.mean(diff**3)
    den = np.std(x, ddof=1)**3
    return num / den

def mykurtosis(x):
    m = np.mean(x)
    diff = x - m
    num = np.mean(diff**4)
    den = np.mean(diff**2)**2
    return num / den

print("Custom Skewness:", myskewness(x))
print("Scipy Skewness:", skew(x))
print("Custom Kurtosis:", mykurtosis(x))
print("Scipy Kurtosis:", kurtosis(x))
```