# Skewness and Kurtosis - Lab

## Introduction

In this lab, you'll calculate skewness and kurtosis for a given dataset in SciPy using Python.

## Objectives

You will be able to:

- Calculate and interpret values of skewness and kurtosis

## Bring in SciPy

In the previous lesson, you have seen formulas to calculate skewness and kurtosis for your data. SciPy comes packaged with these functions and provides an easy way to calculate these two quantities, see scipy.stats.kurtosis and scipy.stats.skew. Check out the official SciPy documentation to dig deeper into this. Otherwise, simply pull up the documentation within the Jupyter notebook using `shift+tab` within the function call or pull up the full documentation with `kurtosis?` or `skew?`, once you have imported these methods from the SciPy package.

You'll generate two datasets and measure/visualize and compare their skew and kurtosis in this lab.

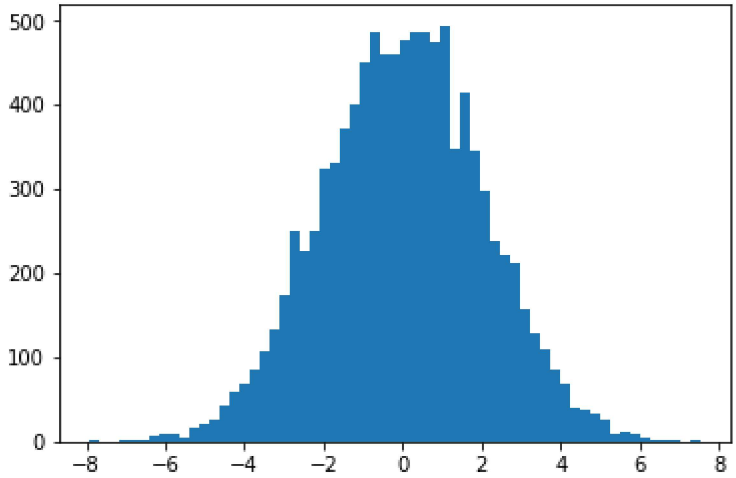
```
# Import required libraries
import numpy as np
import matplotlib.pyplot as plt

from scipy.stats import kurtosis, skew
```

## Take 1

- Generate a random normal variable `x_random` in NumPy with 10,000 values. Set the mean value to 0 and the standard deviation to 2.
- Plot a histogram of the data, set bins to `auto` (default).
- Calculate the skewness and kurtosis for this data distribution using the SciPy functions.
- Record your observations about the calculated values and the shape of the data.

```
x_random = np.random.normal(0, 2, 10000)
plt.hist(x_random, bins='auto')
print ('Skewness =', skew(x_random))
print ('kurtosis =', kurtosis(x_random))
```

```
Skewness = -0.021566018630739 61
kurtosis = -0.03888624629995174
```



```
# Your observations here

# Data is normally distributed (obviously) and thus symmetrical

# A very slight negative skewness is observed as there are slightly more values on t
# side of distribution mean than those on left side


# The kurtosis value shows that this distribution is Platykurtic: The Kurtosis < 0 .
# of kurtosis (Fisher's), 3 is subtracted from the Pearson kurtosis. Fisher's kurtos

# Data is light tailed, and has no outliers.
```

## Take 2

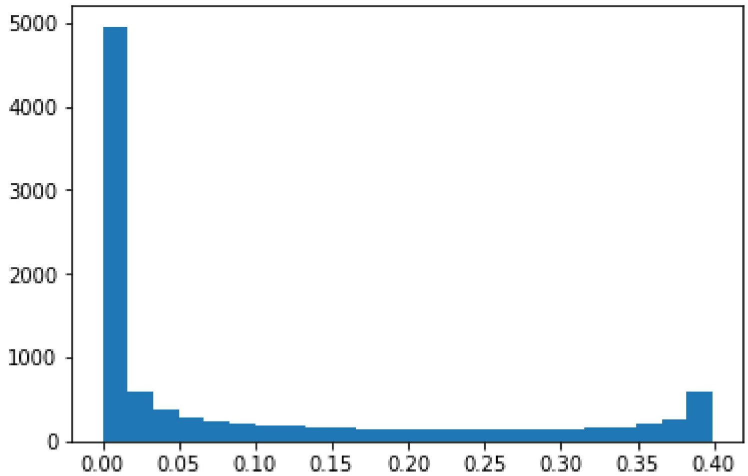
Let's generate another distribution

```
x = np.linspace( -5, 5, 10000 )
y = 1./(np.sqrt(2.*np.pi)) * np.exp( -.5*(x)**2)
```

- Plot a histogram for data $y$, and set bins to auto (default).
- Calculate the skewness and kurtosis for this data distribution using the SciPy functions.
- Record your observations about the calculated values and the shape of the data.

```
plt.hist(y, bins='auto')
print ('Skewness =', skew(y))
print ('kurtosis =', kurtosis(y))
```

```
Skewness = 1.109511549276228
kurtosis = -0.31039027765889804
```




```
# Your observations here

# A high positive skewness is observed as there are more values on the left
# side of the distribution mean than those on right side

# A negative kurtosis value indicates that the distribution has thinner tails
# and a flatter peak than the normal distribution. It is platykurtic. Note that the
# "comparing" to a normal distribution. Looking at the plot, the distribution is cle
# not normal. Kurtosis values are really mostly useful to look at when your observed
# is bell-shaped and you want to know if your tails are lighter or fatter than those
```

## Summary

In this lesson we learned how to calculate, visualize, and analyze the skewness and kurtosis for any given distribution. We worked with synthetic datasets at this stage to get the concepts cleared up. Later we will try these techniques on real datasets to see if they are fit for analysis (or not).

### Releases

No releases published

### Packages

No packages published

## Contributors 6

## Languages

- **Jupyter Notebook** 100.0%