

Skewness and Kurtosis - Lab

Introduction

In this lab, we shall look at calculating skewness and kurtosis for a given dataset in python and SciPy.

Objectives

You will be able to:

- Measure, visualize, and interpret the kurtosis of a dataset
- Measure, visualize, and interpret the skewness of a dataset

Bring in SciPy

In the previous lesson, we have seen formulas for calculating skewness and kurtosis for a given dataset. 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 for digging deeper into this. Right now we can use format:

```
scipy.stats.kurtosis(x)
```

```
scipy.stats.skew(x)
```

Where x is a random continuous variable.

We shall generate two synthetic datasets and measure/visualize and compare their skew and kurtosis in this lab. So lets get to it.

```
In [5]: # Import required Libraries
import numpy as np
import matplotlib.pyplot as plt

from scipy.stats import kurtosis, skew
```

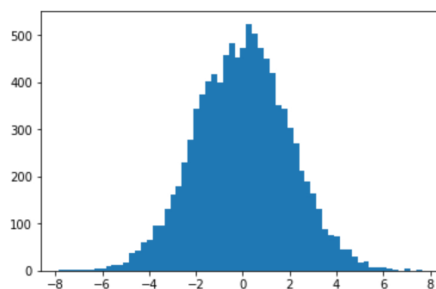
Take 1

Perform following steps

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

```
In [8]: 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 = -0.0025781248189666343
kurtosis = 0.03976806960642154
```



```
In [ ]: # 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 the right
# side of distribution mean than those on left side

# Kurtosis value shows that this distribution is Platykurtic: (Kurtosis < 3)

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

Take 2

Let's generate another distribution

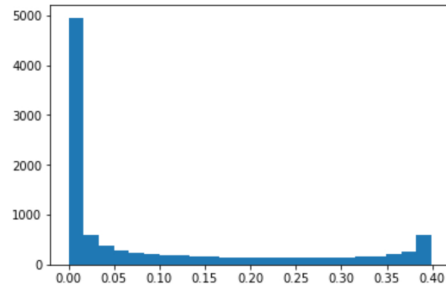
```
In [9]: x = np.linspace( -5, 5, 10000 )
        y = 1./(np.sqrt(2.*np.pi)) * np.exp( -.5*(x)**2 ) # normal distribution
```

Just like above, perform following steps

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

```
In [11]: plt.hist(y, bins='auto')
         print ('Skewness =', skew(y))
         print ('kurtosis =', kurtosis(y))
```

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



```
In [ ]: # Your observations here

# Data is normally distributed but not highly symmetrical. It has more values on Left than right

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

# A negative kurtosis value indicates that the distribution has Lighter tails
# and a flatter peak than the normal distribution.

# Data is Light tailed, and no outliers.
```

Level up (Optional)

Use the formulas given in the previous lesson to calculate skewness and kurtosis with numpy only. Compare your results to the ones shown using Scipy. Do you see any differences in the results? Try at least one other method to calculate skewness and compare the results to previous ones.

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 shall try these techniques on real datasets to see if they are fit for analysis (or not).