Percentiles

In this lesson, we will learn about percentiles.

WE'LL COVER THE FOLLOWING

Representing data through percentiles

Another useful description of a dataset is by using percentiles.

For this we consider ordered data, meaning data that is sorted in ascending order. The 25^{th} percentile marks a data point in the ordered data such that 25% of the data is below this data point and thus 75% is above this data point. If we say that the 25^{th} percentile score on an exam was 85%, then 25% of the candidates scored less than 85% on the exam.

The percentiles of a dataset are commonly referred to as the 'empirical percentiles' as they are the percentiles of the dataset, not of the underlying distribution. The 50^{th} empirical percentile is equivalent to the median of the data. Common intervals to look at are the 50% region around the median, also called the **interquartile range** or IQR.

IQR runs from the 25^{th} empirical percentile to the 75^{th} empirical percentile. The 95% region, which runs from the 2.5^{th} empirical percentile to the 97.5^{th} empirical percentile. Percentiles of a dataset may be computed with the `percentile()` function in the `numpy` package. The first argument is the data, the second argument is a list of percentiles:

```
import numpy as np
import matplotlib.pyplot as plt
```

```
import numpy.random as rnd
from scipy.stats import norm


data = rnd.normal(loc=10, scale=2, size=100)
lower, median, upper = np.percentile(data, [2.5, 50, 97.5])
lower_Q, upper_Q = np.percentile(data, [25, 75])
print('2.5 percentile:', lower)
print('50 percentile:', median)
print('97.5 percentile:', upper)
print('95 percentile:', upper - lower)
print('IQR percentile:', upper_Q - lower_Q)
```

Theoretical percentiles of normal distribution

Theoretical percentiles of a given distribution may be computed with the `ppf` function. The acronym `ppf` stands for percent point function. The percentiles need to be specified as decimals instead of percentiles, for example, 0.5 for the 50 percentile. The theoretical values for the Normal distribution used above are given in the code below:

```
from scipy.stats import norm

norm_theory = norm.ppf([0.025, 0.5, 0.975], loc=10, scale=2)
print (norm_theory)
```

Box and whisker plot

Box-whisker plots or boxplots are also a way to visualize the level and spread of the data. From a boxplot, you can see whether the data is symmetric or not, and how widely the data is spread. You can go back to this lesson for more detail.

Let's test your understanding of random variables with a quiz.