**MIT xPRO Data Engineering Certificate**

**Time Series**

**What Is a Time Series?**

A time series is a sequence of data points that occur in successive order over some period of time. In particular, a time series allows you to see what factors influence certain variables from period to period. Time series analysis can be useful to see how an economic variable changes over time.

**How Do You Analyze a Time Series?**

There are different techniques to analyze time series. Below, the most important techniques are listed and described, and links to online resources are provided to help you learn more about these techniques and their implementation.

**Trends**

A trend is a pattern in data that shows how a time series changes in comparison to relatively higher or lower values over a period of time. In other words, a trend can be observed when the time series increases or decreases. In the real world, trends do not repeat but usually happen for a certain period of time and then disappear. Consider the following example: A stock price may increase for a while, but it’s unlikely that the price will increase forever. Rather, at some point, the stock price will decrease and the trend will change.

A trend could be:

* Uptrend: If the time series analysis shows an upward slope, then the trend is an uptrend.
* Downtrend: If the time series analysis shows a downward slope, then the trend is a downtrend.
* Stationary trend: If the time series analysis doesn’t show any slope, then the trend is stationary.

More information about how trends occur and how they can be visualized in Python can be found here: [Time Series Analysis in Python.](https://www.machinelearningplus.com/time-series/time-series-analysis-python/)

**Percentage Change**

The percentage change is used to calculate by what percentage a variable has changed over a time period. One example is the percentage change in end of day closing prices of stocks. The percentage change is calculated by the formula:

Percentage change =(Current value - Previous value) ÷ Previous value × 100

There are numerous ways to compute the percentage change of a time series. For instance, one could implement the formula above from scratch into a Python *function* and pass the time series to the *function* to compute the percentage change.

Another approach would be to use the pandas Python library and call the [pct\_change()](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pct_change.html)*function* with your time series as an argument to compute the percentage change.

More information about how percentage changes work and how they can be visualized in Python can be found here: [Pandas Percentage Change *Function*.](https://www.geeksforgeeks.org/python-pandas-dataframe-pct_change/)

**Rolling Time *Windows***

In time series analysis, *window* *functions* are useful because they let you perform operations on subsets of your data. Rolling *window* *functions* specifically let you perform operations over each row in a *dataframe.* Consider the following example:

Suppose you have 20 days’ worth of stock market data and you want to know the average price for that stock over the last six days. In this case, the computation is easy. You just need to sum the prices of the last six days and divide it by the number of days.

The situation becomes a little bit more complicated if you want to compute the average of the previous six days for each data point in your dataset. In this case, your rolling *window* will have a size of six; this means that for each point in time, it contains the average of the past six days.

The pandas Python library offers a powerful *function* to compute the rolling time *window* of a time series: the *function* [rolling()](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.rolling.html)

In SQL, rolling time windows can be computed using *window functions.* More information about these *functions* can be found here*:* [*Window Functions* in SQL](https://mode.com/sql-tutorial/sql-window-functions/)

More information about rolling time *windows* and how they can be implemented in Python can be found here: [Rolling Time *Windows* in Python](https://pandas.pydata.org/pandas-docs/stable/user_guide/window.html)