Seasonality and Anomaly Detection in Event Data Using the Discrete Fourier Transformation

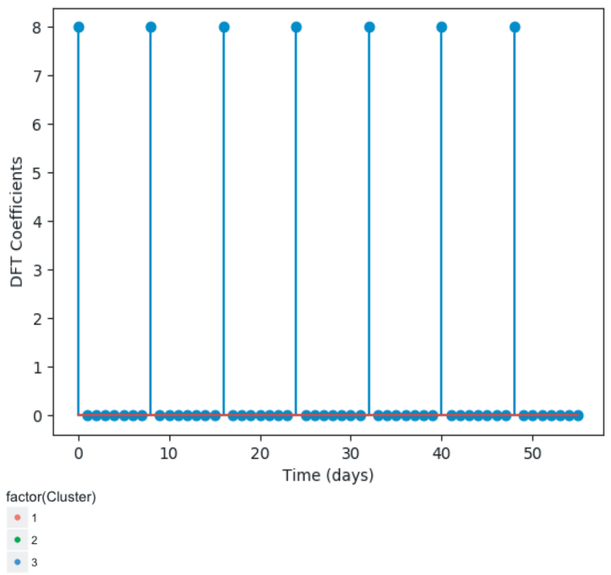
Aryana Collins Jackson | 10-1-19 | Supervised by Dr Seán Lacey

INTRODUCTION - The Discrete Fourier Transform (DFT) algorithm is used for the detection of seasonality in discrete data. In most cases, event instances are common, which is to say that a discrete time series contains non-zero values at every time point. This project explores how the DFT may be used with ***binary event data***, which is defined as rare data in which instances occur at a low frequency and many time points contain a zero. The DFT has never been used in this context before. Thorough understanding of the intricacies of the DFT algorithm and the mathematical concepts that provide the foundation to such work has led to the following results: Two methods (using DFT coefficients and clustering) successfully detect the number of cycles, two methods (using spectral analysis and sinusoidal decomposition) successfully detect periods, one method (using the Convolution Theorem) successfully detects signal shifts, and one method is developed and proposed for identifying anomalies in rare binary data. As the input data is unlabeled without clear independent and dependent variables, the work involves unsupervised learning with mathematics, statistics, time series analysis, and machine learning.

Cycles

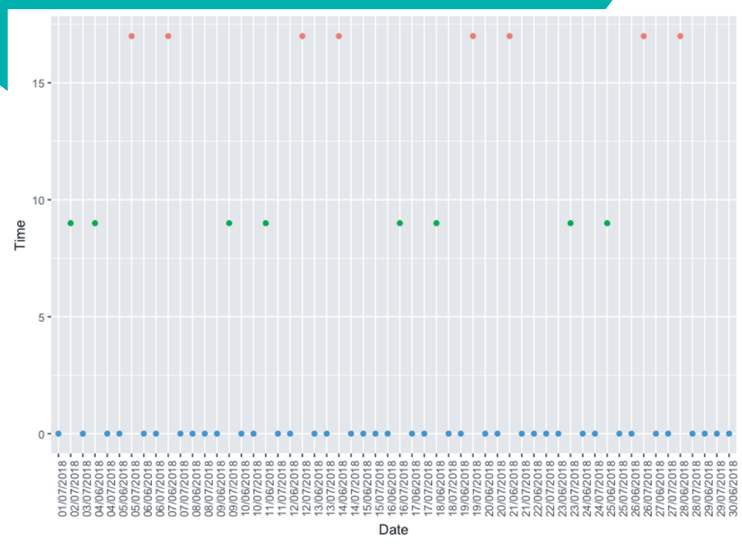
DFT COEFFICIENTS

Running a binary event array with 56 samples through the DFT function produces this graph. The peak occurs at index 8. 56/8=7, meaning ***one cycle*** occurs every 7 days.



CLUSTER ANALYSIS

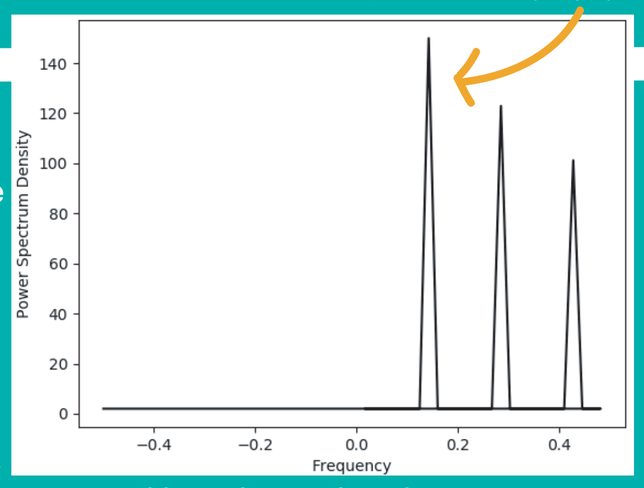
The dataset can be organised into clusters based on day of the week, time of day, date per month, etc. In this graph, a dataset with two cycles is found to have three cycles (one corresponds to days in which an event does not occur).



Period

SPECTRAL ANALYSIS

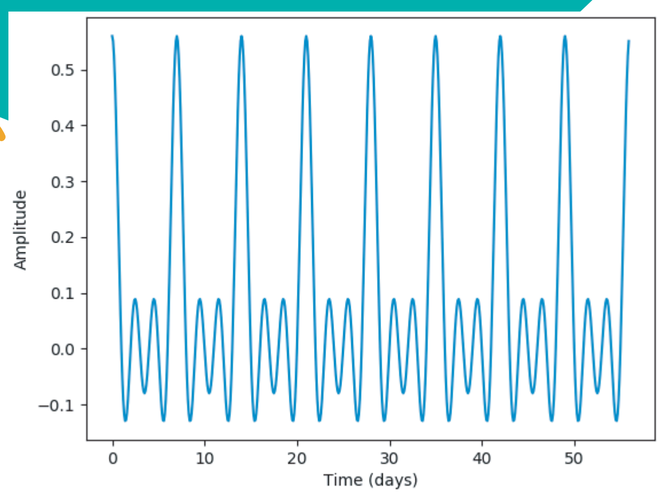
The power spectral density (PSD) plot can be calculated for every possible frequency in the dataset; the possible frequencies are dependent on the sampling rate. This graph shows an initial peak at approximately 0.14: 1/0.14 = ~7, which means that the period is 7 (days).



SINUSOIDAL DECOMPOSITION

The dataset can also be decomposed into sine and cosine waves. The difference between the peaks of the waves corresponds to the period of the signal. This graph displays a weekly event decomposed into sine and cosine waves. As shown, the peaks occur at index 7, 14, etc. corresponding to a period of 7.

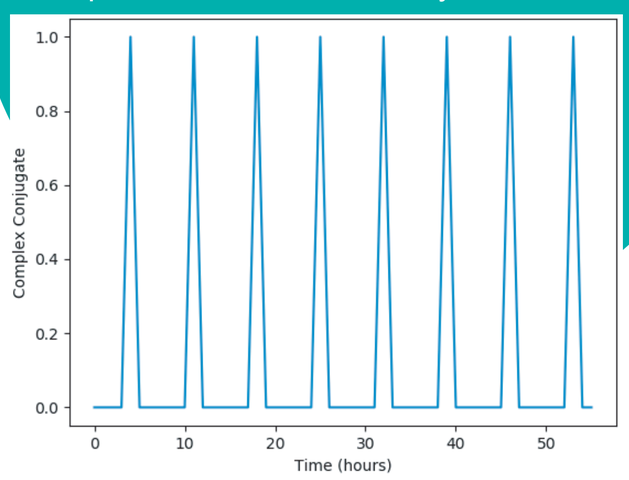
once the period is found, a "perfect" dataset with no anomalies or signal shift can be created. This dataset can be used to detect anomalies and a signal shift in the original dataset



Signal Shift

CONVOLUTION THEOREM

The shifted binary array is multiplied by the unshifted array. That result is divided by the product of the vector norm of the shifted array and the vector norm of the unshifted array. The *argmax* of the result is the signal shift in the time domain. This graph shows how the index 3 (day 4) contains the highest peak and thus the signal shift of this particular dataset is 4 days.



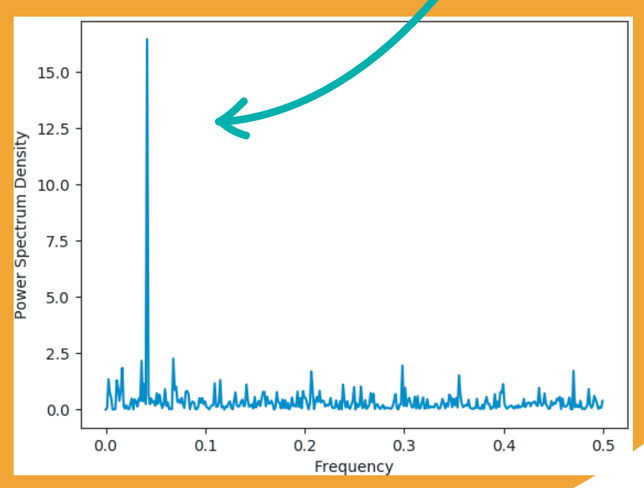
Results

The four main steps here (cycle number, period, signal shift, and anomaly detection) were performed on a dataset from an old social media site called Brightkite. One month of data from one user was analysed. Some of the results are included here.

Two clusters in the data were detected.

The strongest frequency corresponds to a 24-hour period.

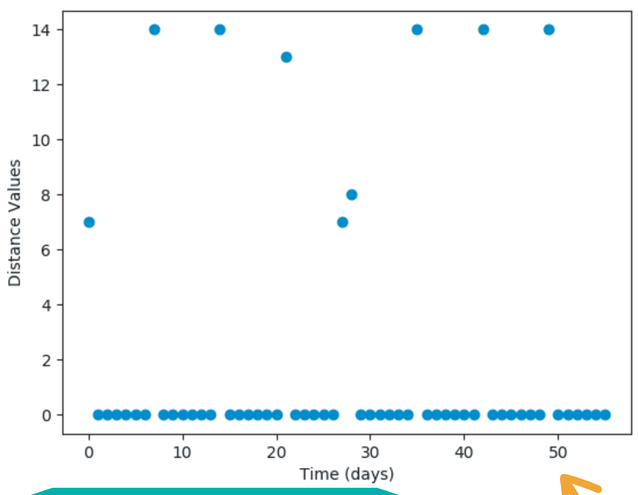
These results were tested with five different tests, including Pearson correlation, in which teh *r*-value returned was ***1 with a p-value<0.05***.



Anomalies

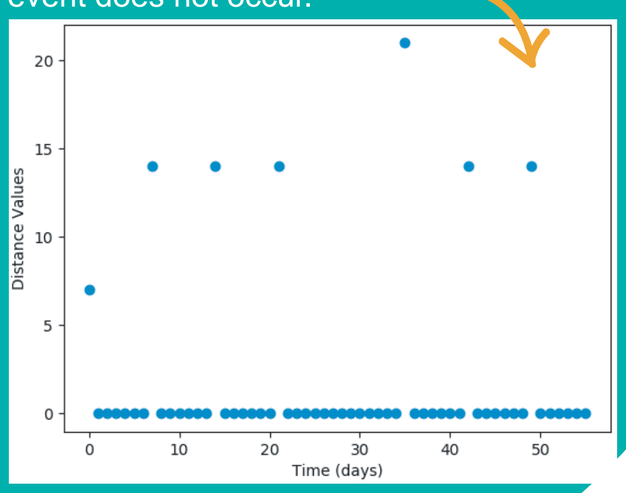
SUM OF DISTANCES

This method was created by the author in order to address the problem of detecting anomalies in binary data. Each instance (previously denoted with a "1") is changed to a value describing the distance to its nearest neighbours. This graph shows what happens when an unexpected event occurs.



there are two types of anomalies: unexpected events that occur and expected events that haven't occurred

This graph shows what happens when an expected event does not occur.



CONCLUSION - This project has shown how seasonality can be successfully detected in discrete rare data and how anomaly detection can be performed with binary data. The number of cycles can be successfully determined from the DFT coefficients and cluster analysis. Periods can be accurately detected using spectral analysis and sinusoidal decomposition. Signal shift detection can be accurately completed using the Convolution Theorem. Perhaps most important, anomaly detection can be performed with high accuracy in rare and binary data. Fourier can now be used for pinpointing exact seasonality along with signal shifts. When examining discrete rare data in time series, finding the exact day, minute, second, or even millisecond in which an instance occurs is important, and now that is possible for data with various fixed-length periods. Expected events that have not occurred and unexpected events that have occurred can be identified and investigated.