Universität Innsbruck

Department of Computer Science
Research Group Quality Engineering

MASTER THESIS

The Title

Martin urek

Supervisor: Ass.-Prof. Dr. Michael Felderer

Innsbruck, May 12, 2018

Contents

Introduction

When working with time series, we often want to determine whether one series causes changes in another. To find this relationship, measuring a cross-correlation and finding a lag is one way how to do it. Lag represents when change in one data series transfers to the other several periods later.

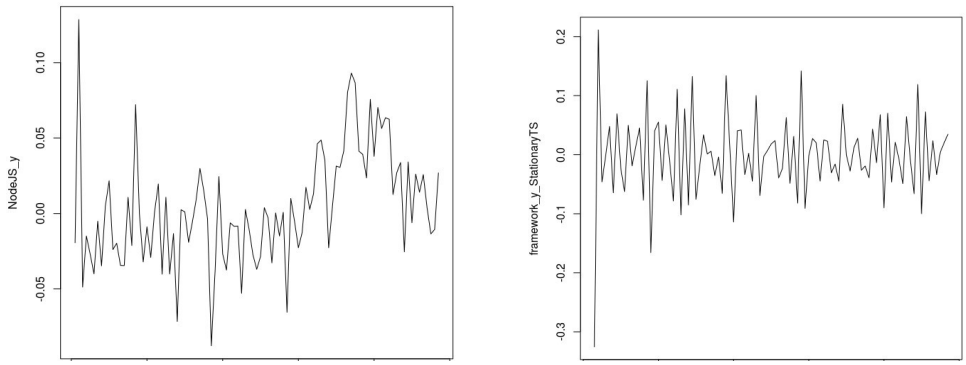
To ensure a cross-correlation calculation makes sense, first I have to determine, whether are the data stationary. A stationary time series is one whose properties do not depend on the time at which the series is observed[1]. More precisely, if y_t is a stationary time series, then for all s, the distribution of $(y_t,,y_{t+s})$ does not depend on t.

To determine whether my data are stationary, I've used the Dickey-Fuller test method of tseries package in R. Results can be seen in the table 0.1

As we can see, some data are not stationary, therefore I can't calculate the cross-correlation on them in this state. To transform non-stationary data into stationary, 2 approaches can be used. These are differencing and transforming. For example sentiment change for NodeJS has a p-value higher than 0.05 what indicates non-stationarity. I've taken these data and differenced the values. I've executed both, seasonal differencing and stationary differencing although seasonal probably was not needed because the data should not be dependant on the season. New differenced values do appear to be stationary in mean and variance, as the level and the variance of the series stays roughly constant over time. Sentiment for NodeJS before and after differencing can be seen in Figure 0.1

Stationarity test of web frameworks sentiment data		
Framework	Dickey-Fuller	p-value
NodeJS	-2.6775	0.2964
AngularJS	-3.883	0.0199
EmberJS	-4.0783	0.0199
VueJS	-3.438	0.0646
CakePHP	-3.480	0.04847
Laravel	-2.57	0.3431
Symfony	-4.3979	0.01

Table 0.1: Stationarity test of sentiment



(a) Before differencing (b) After differencing

Figure 0.1: NodeJS monthly sentiment values

The data are now cleared and eligible for cross correcalation calculation. For this task I've used ccf method in R. As we can see, NodeJS and AngularJS have pretty similar output graphs. In these the correlation values are relatively high and no matter what lag we choose to work with, number of releases positively affects a sentiment.

The content goes here...

Bibliography

[1] Rob J Hyndman, George Athanasopoulos, Slava Razbash, Drew Schmidt, Zhenyu Zhou, Yousaf Khan, Christoph Bergmeir, and Earo Wang. forecast: Forecasting functions for time series and linear models, 2013. *R package version*, 5.