Pomona College

Senior Thesis in Mathematics

Time Series Analysis in Microsoft Error Data

Author:
Grant Belsterling

Advisor:
Dr. Talithia Williams

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

April 11, 2018

Abstract

In this paper I summarize and expand upon my findings from the Harvey Mudd Clinic. In that project, I worked with Microsoft's Bing team to obtain performance metrics Bing collects about the user experience and Microsoft's hardware infrastructure. These metrics were recorded over time, making them natural fits for applying time series analysis.

Contents

Chapter 1

Introduction

The Harvey Mudd Mathematics Clinic is "an innovative approach to higher education that provides advanced students with the opportunity to expand and deepen their knowledge by confronting the challenge of the unknown. Mathematics Clinic teams employ mathematical modeling, statistical analysis, and a host of formidable numerical approaches to concentrate on unsolved problems for industry and government." For this project, the clinic team has partnered with Microsoft to solve problems relating to Microsoft Bing performance.

Microsoft is a multinational technology company based in Redmond, Washington. It is a worldwide leader in personal computer software development with its flagship product, Windows operating system.

Our project focuses on Bing, the search engine owned and operated by Microsoft. Bing powers Bing.com, Yahoo, AOL, Xbox, and Cortana. Bing is the second most popular search engine in the United States.

Microsoft Bing is seeking a solution to quickly identify root causes behind Bing outages and latency spikes. Each second that the Bing site is down costs Microsoft hundreds of dollars; a timely summary of possible causes of an outage are essential in quickly rectifying such a situation.

Towards solving this problem, Microsoft has provided the clinic team an array of time series, with an overall latency 'indicator metric' as well as a group of data from other aspects of Bing.com infrastructure and pipeline that may be responsible for outages and latency spikes. The clinic team has been asked to develop a method to quickly determine which of these processes 'causes', or is correlated with, a spike in the overall latency indicator metric.

Through a year's worth of experimentation, research, and development, we converged on an approach that is useful, easily interpretable, statistically rigorous, and solves the business problem at hand. We identified vector autoregressive methods as this solution. The definition, theory, and applications of vector autoregression are explored in this paper.

Chapter 2

Historical Background

2.1 Time Series

Time series data are usually experimentally observed as ordered pairs. Each observation has a time and a corresponding observed value. This has a structural similarity to the ordered pairs of the form (x, y) we see in classical statistics, save for the fact that here the x coordinate is time. This small difference has large implications, however. Since values are observed close to each other in time, we expect a natural correlation to arise between values. Two data points in a time series are *not* independent. This violates a fundamental assumption in classical statistics, which assumes that data points are independent, a feature that commonly arises in cross-sectional data. This dependence between data points invalidates many statistical models, including regressions. This necessitates the study of time series as its own field.

2.2 Common Applications

To motivate our research in time series, we provide examples of how time series data is present in everyday life, economics, physical sciences, social sciences, etc.

This figure visualizes the number of airline bookings over time. Understanding this data and the phenomena that cause it is central to pricing tickets and ensuring that appropriate flight capacity exists. This time series has obvious business and economic applications.

We note a number of features in this plot that are characteristic of time series. First, we see a **trend**: the average number of passengers increases over time. We also note **seasonality**: in each year, we see a peak in late November and a trough in January and February. This cycle repeats annually, making it seasonal. Finally there is **noise** in the plot. We note small abnormalities in the data that aren't repeated annually and aren't due to the trend of increasing passengers over time. These fluctuations arise because real life rarely follows exact patterns and there is always some element of randomness, or noise, in human behavior.

These three concepts of trend, seasonality, and noise are central in understanding time series and are examined more in depth later in this paper.

Time series also arises in environmental science, as we see in Figure 2.2 which displays ocean heat content over time. The study of time series is central to climate science, energy policy, and understanding the future habitability of our planet.

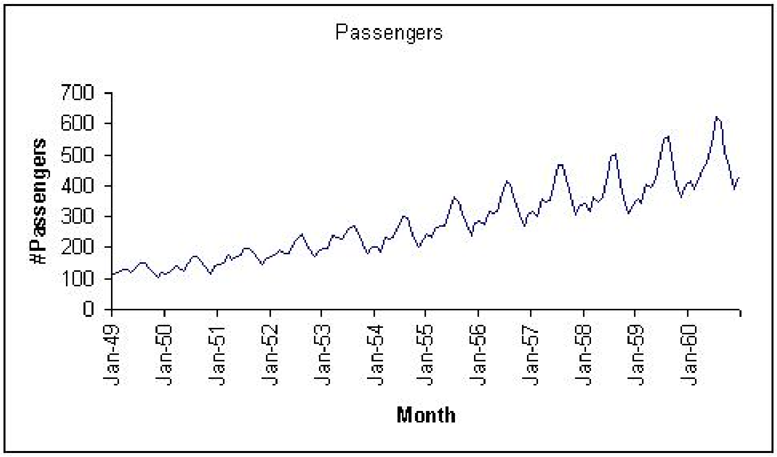


Figure 2.1: Number of airline bookings over time. (from R data set "AirPassengers" visualization with R's plot() function.)

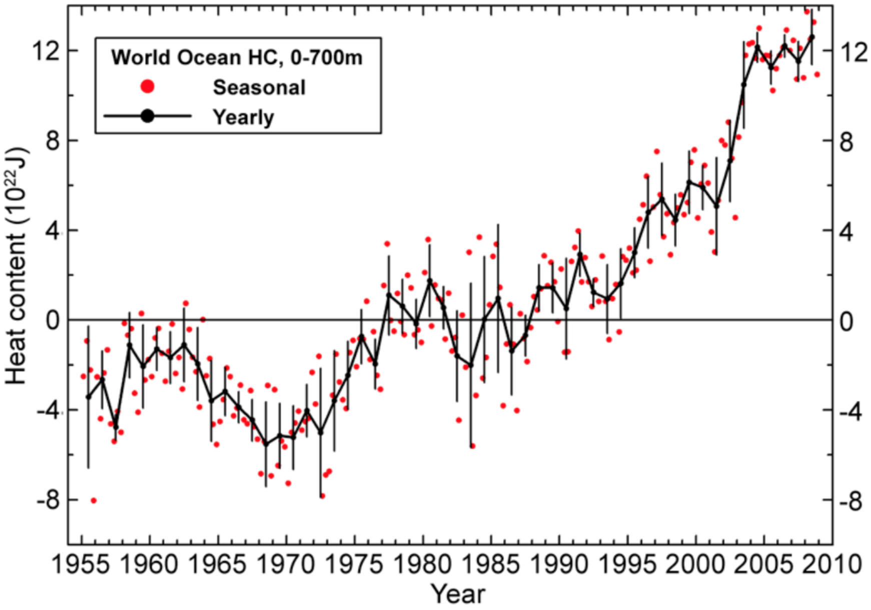


Figure 2.2: Ocean temperature over time. [C,]

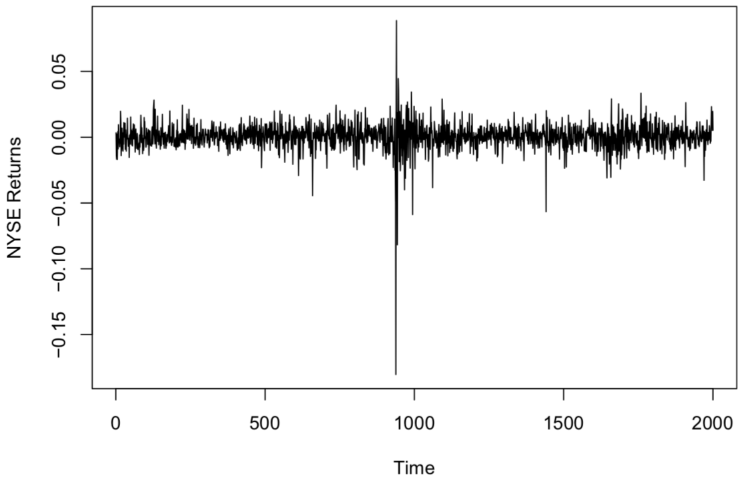


Figure 2.3: New York Stock Exchange returns over time. [Shumway and Stoffer, 2017]

Time series, of course, are central in finance and economics. Figure 2.3 displays the returns on the New York Stock Exchange over time. The crash of October 19, 1987 occurs at t = 938 [Shumway and Stoffer, 2017]. Time series are central to understanding the economics of our society, and predicting things like whether you might be able to find a job if you graduated in 1987.

In this time series, we note that we don't see any trends or seasonality at all. The plot seems to be largely noise, save for the **anomaly** at t=938. Anomaly detection plays a large part in our time series application for Microsoft Bing, and is explored further later in this paper.

For our final example in Figure 2.4, we examine a time series that has *no* trend and *no* seasonality. It is entirely noise. This time series is an example of **white noise**, since the mean of the time series is zero and there is no correlation between values over time.

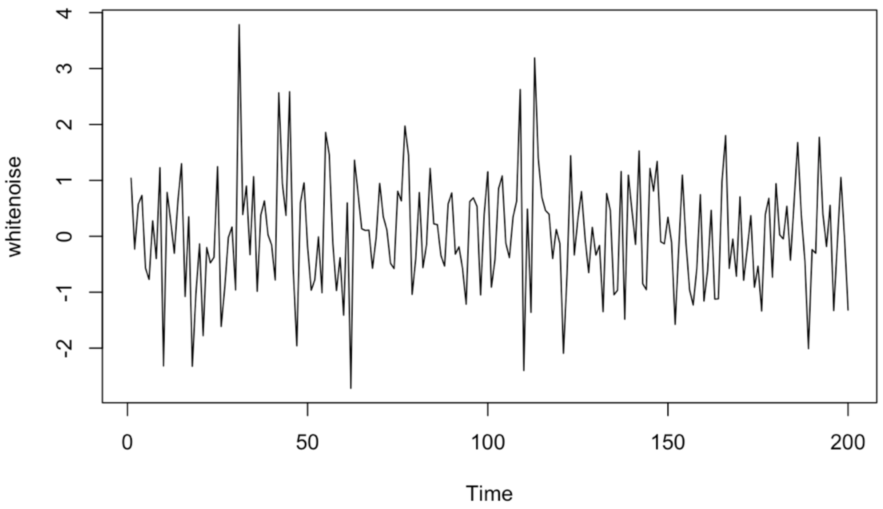


Figure 2.4: White noise generated randomly using the R function arima.sim(list(order=c(0,0,0)),200).

Chapter 3

Theoretical Background

We now define the foundational aspects for examining and understanding time series.

White noise, as introduced in the previous chapter, illustrates a number of key theoretical concepts taken to their extreme. A white noise time series has the least possible *smoothness* in a time series. Smoothness has an intuitive definition: a smoother time series features gradual, continuous curves while a non-smooth time series features jagged, discontinuous spikes. Figure 3.1 is another example of white noise. Notice the extreme lack of smoothness.

To illustrate the concept of smoothness, we also present two other figures, derived from the same white noise time series, but with increasing smoothness.

The time series are smoothed by the technique of *moving averages*. Calculating a moving average smooths a time series by replacing the value at each time point with the average of that time's value and immediately nearby values.

Definition 3.1 *More formally, let w_t be a time series. Let m be the order of a moving average, and let v_t be the smoothed time series resulting from an order m moving average. Then*

$$v_t = \frac{1}{m} * \sum_{i=\lfloor \frac{m}{2} \rfloor}^{m-1} w_{t-i}$$

For example, the order 3 moving average, which was used to derive Figure 3.2 was calculated with $v_t = \frac{1}{3} * (w_{t-1} + w_t + w_{t+1})$.

Calculating the moving average introduces correlation between nearby points in the time series. It follows that if points are a result of averaging neighboring points, then every two adjacent points will be in each others moving average, leading to a degree of correlation between neighboring points. We can think of a smooth time series as having a large degree of correlation between temporally neighboring points, and a non-smooth time series as having a low degree of correlation between points.

This provides a natural transition into more formally defining correlation.

Definition 3.2 *Let X be a time series and s and t be points in time. The autocovariance function is defined as $C = cov(X_t, X_s) = E[(X_t - \mu_t)(X_s - \mu_s)] = E[X_t * X_s] - \mu_t * \mu_s$.*

In smooth time series, the autocovariance remains large even when s and t are distant, while in less smooth time series, the autocovariance decays rapidly as the distance between

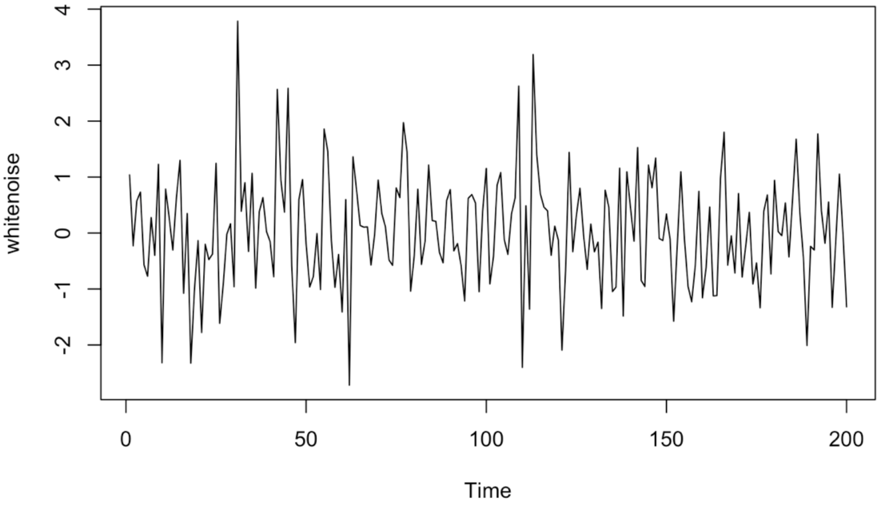


Figure 3.1: White noise.

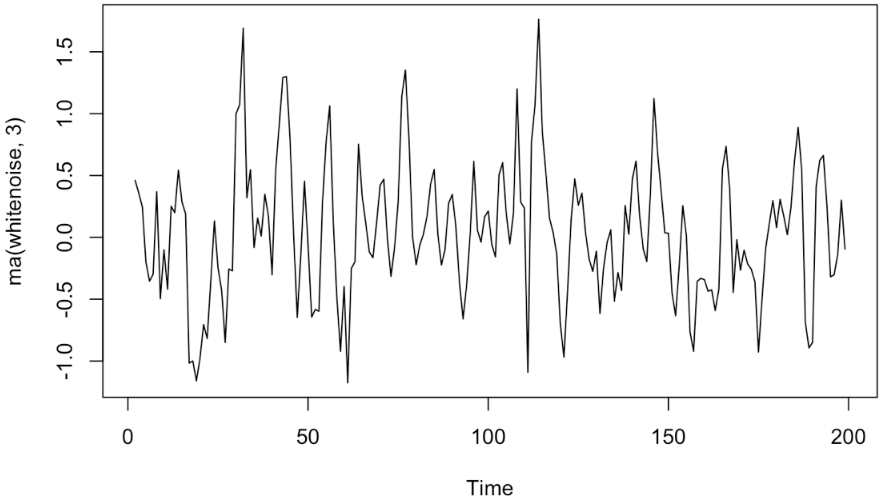


Figure 3.2: The same white noise series, slightly smoothed.

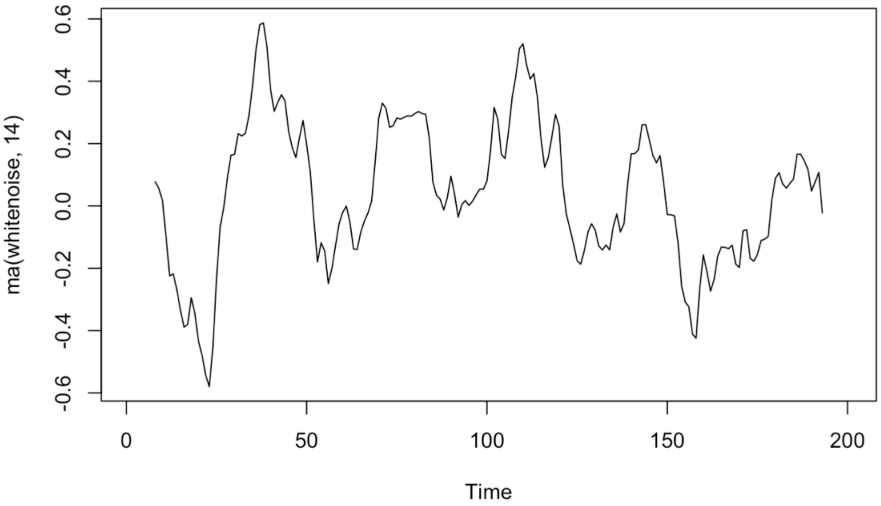


Figure 3.3: The same white noise series, heavily smoothed.

s and t increases.

In most time series, we expect some degree of correlation between immediately past values in time and the present value. If we were to predict the temperature tomorrow, we would first look at the temperature today. This insight of using past values of a time series to predict future values is utilized in the formulation of *autoregressive models*.

Definition 3.3 *An autoregressive model of order p takes the form* $x_t = \beta_0 * x_{t-1} + \beta_1 * x_{t-2} + ... + \beta_p * x_{t-p} + w_t$*, where* x_t *is a stationary time series,* β_n *is a constant, and* w *is a stochastic element of error.*

In autoregressive models, the present value is predicted based on its past lagged values. Estimation of the β values is the crux of time series analysis.

Chapter 4

Data Analysis

In this section we demonstrate the previously covered material through a case study. Time series analysis and decomposition is applied to the Harvey Mudd Clinic's data from Microsoft.

Microsoft had the problem of latency spikes across its services with unknown causes. The time series of the latency metric was supplied by Microsoft, as well as an array of other possibly related metrics. The source of the metrics was not especially well documented internally at Microsoft, leading to unidentified time series being supplied. Further, in the context of this report, non-disclosure agreements limit the amount of transparency we can provide. Given these two conditions, no domain knowledge could be applied to the time series analysis. Although this necessitated much more preliminary exploratory data analysis, it had the useful effect of requiring us to write reusable, extensible code rather than code that was tailor-made to apply only to the initially provided data sets. As such, the project acquired an engineering focus as well as the obvious statistical focus. Of course, the expectations and expertise of the client (Microsoft software engineers) contributed to this engineering focus as well.

4.1 Exploratory Data Analysis

The natural first step in time series analysis is to plot the supplied data. The symptom metric (representing in this case latency) is provided in Figure 4.1.

We notice regularity and some periodicity in the higher troughs. Let's use a smaller time frame to examine this, as pictured in Figure 4.2. The periods with the lower troughs are weekdays (e.g., June 5- June 9). During weekdays, the symptom metric drops to around 2500 at night and peaks at around 3200 during the day. This cyclic behavior remains relatively constant across weekdays.

However, there is a different repeating trend over weekends (e.g. June 10-11). During weekends, the troughs do not dip to the same extremes as during the weekdays. Instead, the symptom metric remains relatively consistent, hovering around 3100 without extreme dips.

We note these features of the time series to motivate the aspects of the time series that ought to be removed, or at the least, accounted for. The mean of the time series increase dramatically over weekends; this forms a trend in the time series that must be removed to achieve stationarity. We also notice seasonality with the period of a day, as there are regular

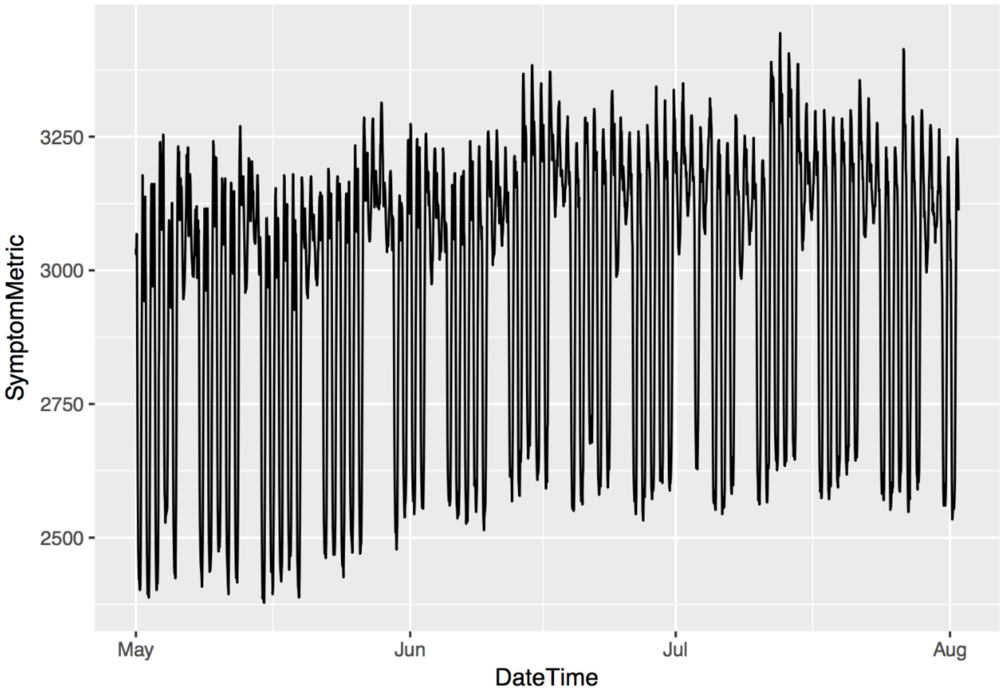


Figure 4.1: The entirety of the given symptom metric.

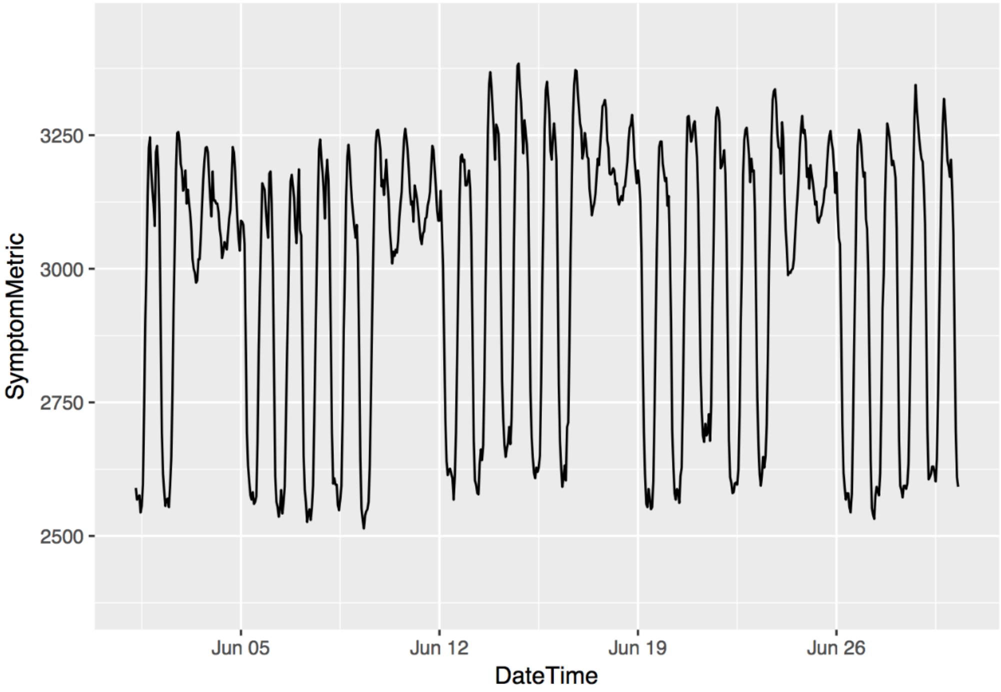


Figure 4.2: The symptom metric at a weekly scale.

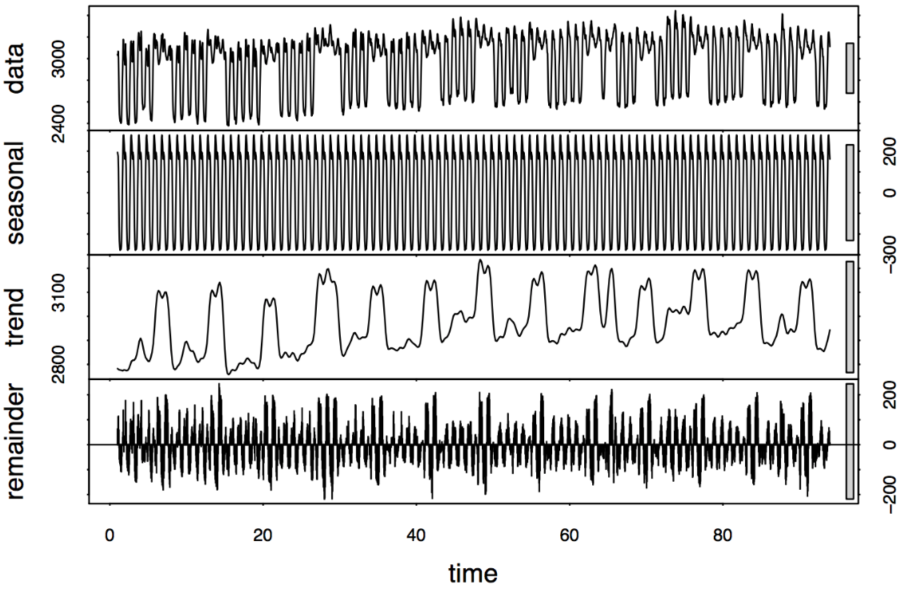


Figure 4.3: The decomposition of the symptom metric

spikes and troughs that correspond to day/night cycles. Finally, there is also the remainder we expect in the time series, given that the data arose from observed stochastic processes and was not deterministic.

Next we apply time series decomposition to extract these aspects (trend, seasonality, remainder) from the given time series.

4.2 Time Series Decomposition

We decompose and detrend a time series for a number of reasons. First, we wish to extract a weakly stationary time series for which it is valid to apply a number of time series analysis techniques (a technique of particular concern for us is calculating cross-correlation). The 'remainder' component of the time series, as we will see, serves as a weakly stationary time series on which we continue the analysis.

We also apply time series decomposition in order to understand the underlying factors and latent information in a time series. The seasonality of the data clarifies the periodicity of the time series and the natural way of comprehending the cycles of the time series (e.g. are they hourly? daily? yearly?). The trend in the data describes how the mean of the time series changes over time. The decomposition of the symptom metric is included in Figure 4.3.

The raw data is plotted in the top pane of Figure 4.3. Below that is the extracted seasonality. Note that in this data, the seasons correspond to days, with latency peaking during the day and decreasing at night. The third pane shows the extracted trend, calculated from the moving average of the means. We see the trend increase over weekends and decrease over weekdays.

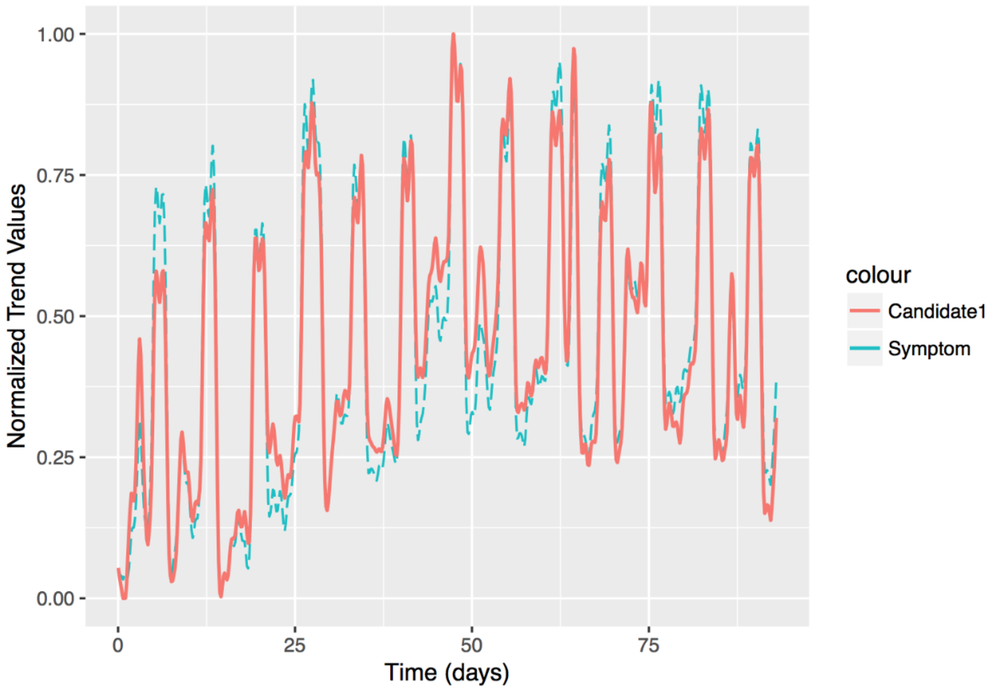


Figure 4.4: The symptom metric trend normalized and plotted alongside the candidate one metric trend

The final pane is the remainder, which represents the residuals of the model created by predicting the raw time series using as predictors the seasonality and the trend. Importantly, the remainder time series is weakly stationary, centered around a mean of 0.

Extracting the trends also provides an intuitive graphical representation for assessing the correlation between time series. The plots of the trends of the symptom overlaid with each metric proved useful in getting an initial sense of the data and their correlations, since the trends are smoothed and stripped of seasonality.

These plots are included in Figure 4.4 and Figure 4.5. From the plot of Candidate Metric One against the symptom, we see they are obviously very highly correlated. From the plot of Candidate Metric Two against the symptom, we see they are likely not correlated at all.

From this, we certainly gain useful information. However, we would like a numerical method that provides a quantity to measure the degree of correlation between two arbitrary time series. A numerical approach allows for the advantages of: objectivity rather than graphical interpretation, quick comparison between time series, and extensibility across arbitrary time series.

4.3 Cross-Correlation

To those ends, we employ the cross-correlation function between two stationary time series. A function was written in R to compute these.

An example of the output of this function is included in Figure 4.6. From this, we extract

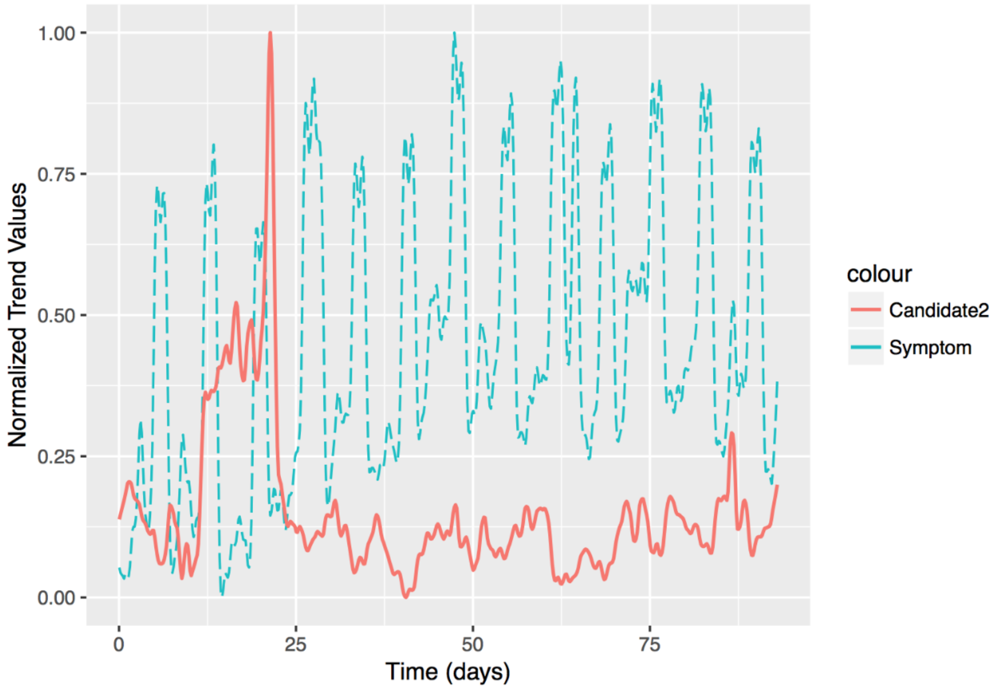
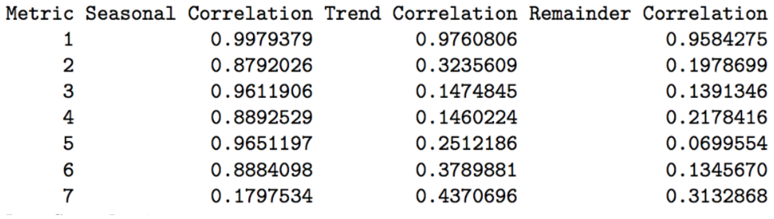


Figure 4.5: The symptom metric trend normalized and plotted alongside the candidate two metric trend



```
Metric Seasonal Correlation Trend Correlation Remainder Correlation
     1            0.9979379         0.9760806             0.9584275
     2            0.8792026         0.3235609             0.1978699
     3            0.9611906         0.1474845             0.1391346
     4            0.8892529         0.1460224             0.2178416
     5            0.9651197         0.2512186             0.0699554
     6            0.8884098         0.3789881             0.1345670
     7            0.1797534         0.4370696             0.3132868
```

Figure 4.6: The output of the R function corr_stl() applied to provided sample data sets.

the information that candidate one is very strongly correlated (across all measures) with the symptom metric. However, the degree of correlation of the other metrics remain less than obvious. For example, take candidate 7, which has the second highest remainder correlation of .31. However, it has the lowest seasonality correlation of .17. Thus it remains unclear whether it is well correlated with the symptom metric or not correlated at all.

We notice other issues with this cross-correlation methodology as well. It provides no information on statistical significance. It remains statistically valid only for the remainders, and only in certain cases where the remainders achieve stationarity.

To move towards a more enlightening methodology, we turn to vector autoregression.

Chapter 5

Vector Autoregression

There were a number of shortcomings with the cross-correlation approach that vector autoregression attempts to overcome. First, the results do not include a measure of statistical significance. The approach outputs only the effect size. We would like to be able to tell apart whether a metric has a large but tenuous effect, or a small but definite effect.

A simplistic cross-correlation approach also does not isolate the impact of one metric with respect to all other metrics. It examines each one in a vacuum. This begins to cause problems when candidate metrics become correlated with each other and not only with the symptom metric (in classical statistical linear modeling, this phenomenon is called *collinearity*).

Another shortcoming is that the cross-correlation approach does not begin to broach the question of causality. Of course, although through statistical methods we never determine causality with finality, we can attempt to find evidence for *Granger causality* with time series data.

Definition 5.1 *A time series X is said to Granger-cause Y if it can be shown, usually through a series of t-tests and F-tests on lagged values of X (and with lagged values of Y also included), that those X values provide statistically significant information about future values of Y.*

An important caveat is that Granger causality can easily fall prey to the *post hoc ergo propter hoc* fallacy. This translates literally to "after this, therefore because of this." Granger causality makes the assumption that if a response is seen after a perturbation in a different time series, then the response is *because of* this perturbation. We can easily imagine cases in which this is not true.

This has an important application to the domain of latency data. Some metrics are 'early warning' indicators in that their latency spikes earlier problems propagate down the infrastructure pipeline and become evident in other metrics. A Granger causality approach would assert that the first, 'early warning' metric *caused* the slow performance in later metrics. This is a strong and erroneous assertion, and is therefore something we wish to avoid providing as a solution.

5.1 VAR Theory

We now formally define vector autoregression.

Definition 5.2 *VAR is a modeling technique to describe the change in a set of k variables over a time period (t=1,2,...,T) as a linear function of their past values. A p-th order VAR, denoted VAR(p) is defined as:* $y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + ... + A_p y_{t-p} + e_t$*, where c is a vector of constants/intercepts of length k, and A is a k by k matrix and e is a vector of length k of error terms.*

As an example, we consider a problem with only two time series, y_1 and y_2, and considering only one time lag value. Applying vector autoregression to these two variables produces the output below:

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} * \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \end{bmatrix}$$

Vector autoregression is a technique to estimate the value for c and A.

5.2 Estimation of Vector Autoregression Parameters

The parameters for A and c are estimated using multiple ordinary least squares. We have $y_t = A * Y_{t-1} + E*$, where Y is a matrix of our time series, A are the coefficients we wish to estimate, and E are the error terms. Using least squares, we minimize the sum of squares of E: $\sum_{i=1}^{n} e_i^2 = E * E' = (Y - AY) * (Y - AY)'$.

We minimize this distance by projecting the Y_{t-1} vector onto the column space of A matrix. This projection is our least squares estimate for $Y_{\hat{t-1}}$. The matrix A is equivalent to $A = (Y'_{t-1} * Y)^{-1} Y'_{t-1} y$.

5.3 Impulse Response

When VAR is applied, we are interested in capturing interactions between time series. We may be interested in how a slight change in one variable affects the others in the system. Quantifying this effect is done by calculating the *impulse response function.* In this technique, we create an “impulse” or a small shock in a single variable at a single point in time (say, increase it by 10%). We then calculate, through our vector autoregression model, the “response” of the other metrics due to this shock.

In the simple case of observing the effects of the shock in the next time period, $t + 1$, we can take the observation y_t, increase the value of one variable by 10% to obtain $y_{shock,t}$ and use that shocked vector as an input into the vector autoregression model to obtain a prediction for $y_{\hat{shock,t+1}}$. The multiplicative difference between the predicted value of $y_{\hat{shock,t+1}}$ and the actual value for y_{t+1} is the *impulse response value* for a lag of one. Impulse response values for larger time lags can be found through a recursive process in this manner.

Chapter 6

Vector Autoregression Applied

6.1 Case Study

We now apply vector autoregressive methods on system performance data supplied by Microsoft Bing to approach the problem of identifying possible root causes for site errors.

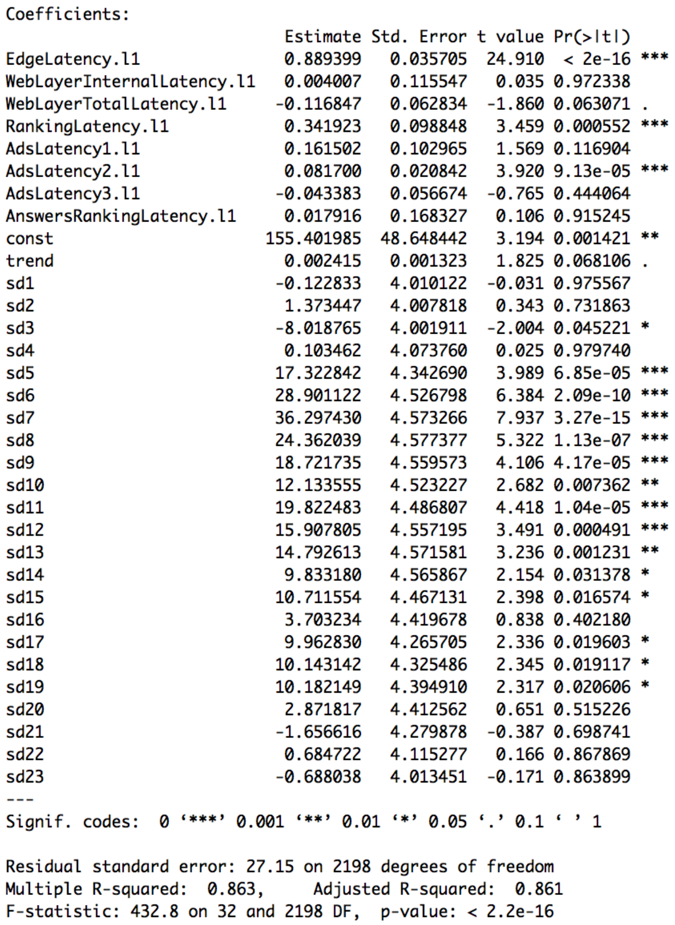
The results of the model applied to a typical, paradigmatic example are included in Figure 6.1. In this model, the symptom metric is 'Edge Latency', a measure of total latency of the system, and the candidate metrics are latency measures from smaller components of the Bing infrastructure. For vector autoregression, we model Edge Latency predicted by all of the candidate metrics. In this analysis, we see reasonably good performance from vector autoregression.

The seasonality constants (denoted by the prefix sd for 'seasonal dummy' variables) indicate the hour of day for an observation. They enter the model as predictors, which prevents seasonal variation being attributed to other metrics rather than merely the hour. This serves a similar function as detrending the time series in order to obtain weakly stationary ones, a method we have applied in previous approaches.

As for performance, we notice the adjusted r^2 value is .861. This can be interpreted as the model is powerful enough to explain 86% of the variation in our symptom metric, Edge Latency. In statistical modeling, this is considered a relatively high r^2 value and provides reassurance that the model has some validity.

Some of the candidate metrics are significant predictors (they explain an amount of variation in the symptom metric too large to be due to random change), and some candidate metrics fail these tests of significance. The model would not be noticeably different were these metrics excluded and the model re-fit. If this is the case, we can reasonably conclude that if a candidate metric has no bearing on the symptom metric, it is not a root cause. By applying this test, we narrow our set of possible root causes from the entire set of candidate metrics to only the set of statistically significant metrics. In this data set, that set includes only RankingLatency and AdsLatency2 (as indicated by their p-values being below .05). Microsoft employees with knowledge of the malfunctions from the time this data set was recorded confirmed that these two metrics were indeed the root causes they were looking for. This, ultimately, was the gold standard test we were looking to pass.

We've properly identified root causes, but we would like to quantify the size of their



```
Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
EdgeLatency.l1              0.889399   0.035705  24.910  < 2e-16 ***
WebLayerInternalLatency.l1  0.004007   0.115547   0.035 0.972338    
WebLayerTotalLatency.l1    -0.116847   0.062834  -1.860 0.063071 .  
RankingLatency.l1           0.341923   0.098848   3.459 0.000552 ***
AdsLatency1.l1              0.161502   0.102965   1.569 0.116904    
AdsLatency2.l1              0.081700   0.020842   3.920 9.13e-05 ***
AdsLatency3.l1             -0.043383   0.056674  -0.765 0.444064    
AnswersRankingLatency.l1    0.017916   0.168327   0.106 0.915245    
const                     155.401985  48.648442   3.194 0.001421 ** 
trend                       0.002415   0.001323   1.825 0.068106 .  
sd1                        -0.122833   4.010122  -0.031 0.975567    
sd2                         1.373447   4.007818   0.343 0.731863    
sd3                        -8.018765   4.001911  -2.004 0.045221 *  
sd4                         0.103462   4.073760   0.025 0.979740    
sd5                        17.322842   4.342690   3.989 6.85e-05 ***
sd6                        28.901122   4.526798   6.384 2.09e-10 ***
sd7                        36.297430   4.573266   7.937 3.27e-15 ***
sd8                        24.362039   4.577377   5.322 1.13e-07 ***
sd9                        18.721735   4.559573   4.106 4.17e-05 ***
sd10                       12.133555   4.523227   2.682 0.007362 ** 
sd11                       19.822483   4.486807   4.418 1.04e-05 ***
sd12                       15.907805   4.557195   3.491 0.000491 ***
sd13                       14.792613   4.571581   3.236 0.001231 ** 
sd14                        9.833180   4.565867   2.154 0.031378 *  
sd15                       10.711554   4.467131   2.398 0.016574 *  
sd16                        3.703234   4.419678   0.838 0.402180    
sd17                        9.962830   4.265705   2.336 0.019603 *  
sd18                       10.143142   4.325486   2.345 0.019117 *  
sd19                       10.182149   4.394910   2.317 0.020606 *  
sd20                        2.871817   4.412562   0.651 0.515226    
sd21                       -1.656616   4.279878  -0.387 0.698741    
sd22                        0.684722   4.115277   0.166 0.867869    
sd23                       -0.688038   4.013451  -0.171 0.863899    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27.15 on 2198 degrees of freedom
Multiple R-squared:  0.863,     Adjusted R-squared:  0.861 
F-statistic: 432.8 on 32 and 2198 DF,  p-value: < 2.2e-16
```

Figure 6.1: The output of the R function VAR() from the ‘vars’ R package applied to a typical data set from Bing, with parameters of lag=1 and seasonality=24.

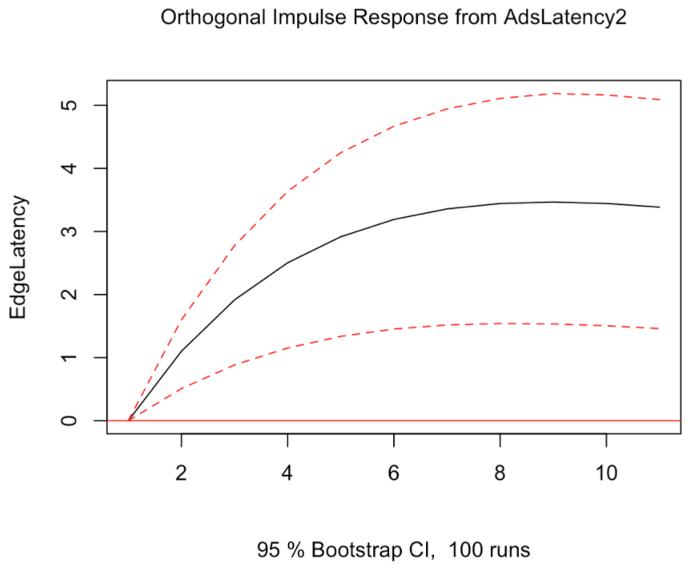


Figure 6.2: The plot for the impulse response function of AdsLatency2. The x-axis indicates the time since the impulse, and the y-axis indicates the size of the response in the symptom metric, Edge Latency.

influence. Although identifying root causes can be done in a heuristic, time-consuming manner by human analysts, calculating their impact is a much more challenging task. However, leveraging the impulse response functionality allows us to do that.

Included are the impulse response function plots for the two significant metrics, AdsLatency2 and RankingLatency, as well as an arbitrary example of an insignificant metric, AdsLatency3. These can be found in Figure 6.2, Figure 6.3, and Figure 6.4, respectively.

In these plots, the x-axis indicates the time since the impulse on the candidate metric of interest, and the y-axis indicates the size of the response in the symptom metric (expressed as a multiplier from what the Edge Latency otherwise would have been). The red dashed lines indicate the 95% confidence interval for the size of the impulse response, constructed from bootstrapping the data and re-fitting many models and impulse response coefficients. Notice that for the significant metrics, zero is nowhere within the 95% confidence interval. That is, we can be reasonably confident that there is a non-zero effect from shocking this candidate metric. However, for the non-significant metrics such as AdsLatency 3 in Figure 6.4, zero always lies within the 95% confidence interval, indicating that we cannot be confident that this metric has any effect on the symptom metric.

From these, we can also quantify the effect size. The impulse used in this analysis was a 10% increase in the symptom metric. We can glean from the plots, for example, that 4 hours after a 10% increase in AdsLatency2, the symptom metric would have increased by a factor of 2.5.

Though this is only one typical example of applying vector autoregression and impulse response analysis to a data set, it is representative of the analysis we have done on the many data sets provided by Microsoft.

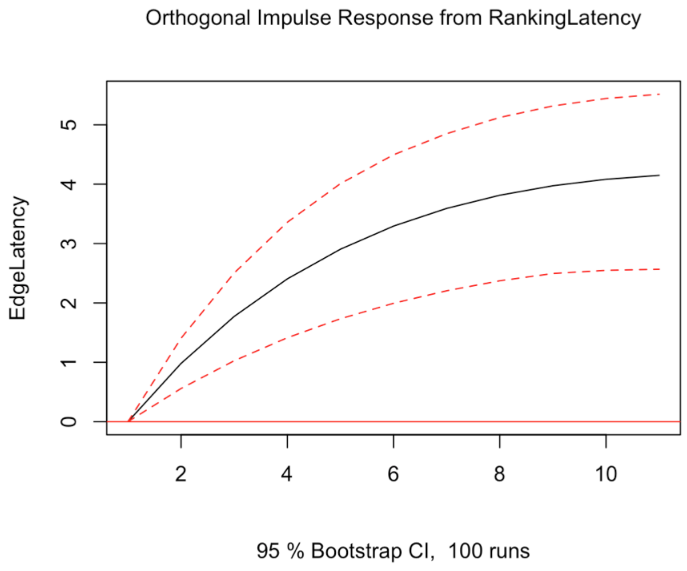


Figure 6.3: The plot for the impulse response function of RankingLatency. The x-axis indicates the time since the impulse, and the y-axis indicates the size of the response in the symptom metric, Edge Latency.

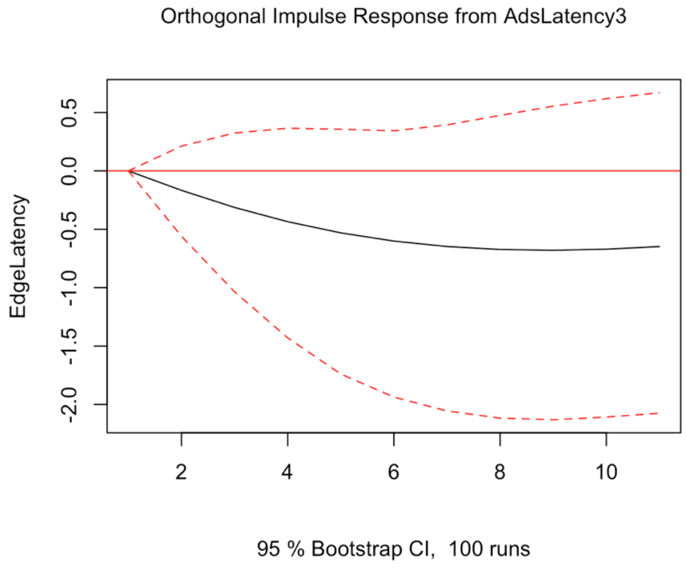


Figure 6.4: The plot for the impulse response function of AdsLatency3. The x-axis indicates the time since the impulse, and the y-axis indicates the size of the response in the symptom metric, Edge Latency.

6.2 Discussion

Though vector autoregression proved to be the most fruitful method explored and applied during this project, it was not a flawless model.

There were many reasons for selecting vector autoregression as our final delivered model. Most important was transparency and interpretability. A goal was to select a model with clear, exposed coefficients rather than a black box model. Vector autoregression, unlike some machine learning and artificial intelligence methods, fit transparent models with coefficients that can be interpreted and explained. For example, we made much use of citing p-values for a predictor's significance in determining whether it was a possible root cause. Further, we use the coefficient estimates to assess the effect size of each metric.

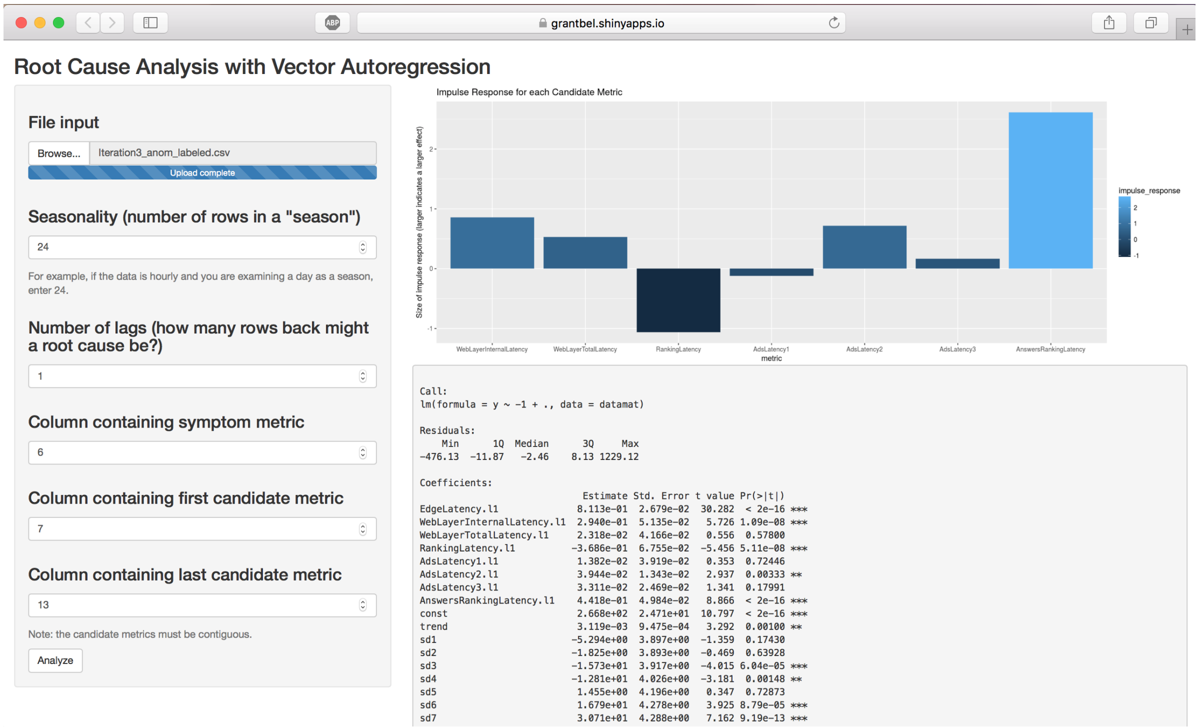
The model also provided interpretability, which was key in developing a method and tool to empower non-statisticians to interpret data and assess root causes. The model was built atop linear regression, which is a relatively accessible method for non-expert, but technical, users. The model also allowed for distilling the output to simple, accessible solutions: a yes/no for significance (whether a p-value was greater than .05 or not), and a single value for the impulse response coefficient after one hour.

Some drawbacks of the method is that it is sensitive to parameters provided by the end user. For example, the number of lags to examine is a critical user choice that modifies the results of analyses significantly. Even more critical is the choice for seasonality; controlling for hourly cycles produces drastically different results than controlling for daily cycles, for example. These drawbacks can be ameliorated through user education.

A more troublesome aspect of the vector autoregression approach is its sensitivity to time span and its treatment of anomalies. In the analysis, we are primarily concerned with the root cause of *anomalies* in the time series, rather than root causes of the time series as a whole. This is because anomalies in the symptom metric corresponds to a site malfunction or error, and we are only interested in finding the causes of these incidents. If we input data from a large time span with few anomalies, the anomalies lose leverage and the regressions are pulled away from these points towards the mean of the rest of the data where the systems are functioning regularly. In effect, this causes the analysis to ignore the site incidents and only give us information on the root cause of regular system functioning, which isn't helpful in this case. This problem is always present, but can be mitigated by supplying a data set with a larger proportion of time corresponding to site incidents.

6.3 Shiny Application

After implementing and refining the methodology described above on many data sets, I developed a Shiny application to automate this process. The application takes a csv file as an input, as well as a set of parameters entered by the user such as periods for seasonality (hourly, daily, weekly, etc) and the number of lags to be considered. The app then runs vector autoregression on the metrics of interest and displays the output of this model for more technical users to interpret. Most importantly, it outputs the impulse response coefficients in an easy-to-interpret bar plot, where the largest and brightest bar indicates the most possible root causes. This app has empowered our liaisons to run statistical analyses themselves and



Root Cause Analysis with Vector Autoregression

File input

Browse... Iteration3_anom_labeled.csv

Upload complete

Seasonality (number of rows in a "season")

24

For example, if the data is hourly and you are examining a day as a season, enter 24.

Number of lags (how many rows back might a root cause be?)

1

Column containing symptom metric

6

Column containing first candidate metric

7

Column containing last candidate metric

13

Note: the candidate metrics must be contiguous.

Analyze

```
Call:
lm(formula = y ~ -1 + ., data = datamat)

Residuals:
    Min      1Q  Median      3Q     Max
-476.13  -11.87   -2.46    8.13 1229.12

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
EdgeLatency.l1              8.113e-01  2.679e-02  30.282  < 2e-16 ***
WebLayerInternalLatency.l1  2.940e-01  5.135e-02   5.726 1.09e-08 ***
WebLayerTotalLatency.l1     2.318e-02  4.166e-02   0.556  0.57800
RankingLatency.l1          -3.686e-01  6.755e-02  -5.456 5.11e-08 ***
AdsLatency1.l1              1.382e-02  3.919e-02   0.353  0.72446
AdsLatency2.l1              3.944e-02  1.343e-02   2.937  0.00333 **
AdsLatency3.l1              3.311e-02  2.469e-02   1.341  0.17991
AnswersRankingLatency.l1    4.418e-01  4.984e-02   8.866  < 2e-16 ***
const                       2.668e+02  2.471e+01  10.797  < 2e-16 ***
trend                       3.119e-03  9.475e-04   3.292  0.00100 **
sd1                        -5.294e+00  3.897e+00  -1.359  0.17430
sd2                        -1.825e+00  3.893e+00  -0.469  0.63928
sd3                        -1.573e+01  3.917e+00  -4.015 6.04e-05 ***
sd4                        -1.281e+01  4.026e+00  -3.181  0.00148 **
sd5                         1.455e+00  4.196e+00   0.347  0.72873
sd6                         1.679e+01  4.278e+00   3.925 8.79e-05 ***
sd7                         3.071e+01  4.288e+00   7.162 9.19e-13 ***
```

Figure 6.5: The UI for the shiny app. The results on the right pane visualize the impulse response coefficients in a bar plot.

obtain results quickly.

6.4 Future Work

The most glaring area for potential future work is in model validation, performance assessment, and model tuning. Over the course of the year-long development process, Microsoft liaisons were unable to provide labelled data in which a true root cause was known. Hence, it was incredibly difficult to assess the performance of the model and validate results.

We did, however, construct a primitive solution to "label" data algorithmically. If there was an anomaly in the symptom metric, we check if there are anomalies in the same time period in other metrics, and if so, assign the 'root cause' label to the anomalous candidate metric 'furthest' from the symptom metric (determined from domain knowledge of the way the metrics represent Microsoft infrastructure). This algorithm approximates the heuristic way that Microsoft analysts are currently approaching the issue.

Despite this method being suggested and approved of by Microsoft liaisons, there are a number of difficulties in this approach. First, we expect root causes to not be contemporaneous with their effects. Second, we have imperfect knowledge of the system of metrics (due to the scale and complexity of the Microsoft infrastructure), making it difficult to assign root cause labels when there are multiple anomalies at the time. Making this designation often relies on a priori knowledge from our liaisons, which does nothing more than give the liaison's assumptions, biases, and intuitions a veneer of mathematical certainty and objectivity.

Finally, this approach gave rise to a methodological and ethical problem of giving us, the data analysts, control over both the "truth" and the tool for converging onto that "truth." In most data science cases, true labels are observed from reality and therefore correspond to a level of truth free from the scientist's tampering. However, in this case, the scientists are given the power to construct an algorithm to create that truth in the first place. This allows implicit assumptions, preconceived notions, and biases to seep into the data. By fitting a model to these constructed truths, we merely fit the model to preconceived notions of what we expect to find. This precludes the possibility of discovering novel insights from the data.

These concerns did not deter the liaisons from pressing to continue development.

Chapter 7

Time Series Analysis and the Historical Development of Vector Autoregression

Working with time series similar to the sort Microsoft provided and attempting to uncover relationships between variables is not a novel task. The work explored in this paper is particularly indebted to the work of many mathematicians, statisticians, and economists throughout the twentieth century. Many of the methods applied in this paper were conceived for the purpose of understanding economic data, which is naturally expressed in the form of time series (for example, GDP over time). In this chapter, we briefly explore the field of econometrics and the historical development and context for vector autoregressive and time series methods.

7.1 Econometrics and Time Series

Econometrics is a subfield of economics that uses mathematical methods and statistical models to describe economic systems [Pesaran,]. Econometrics asserts that, beneath the cruel randomness and disorder of occurrences in the world, there exists some regular non-varying structure. Statistics attempt to describe this structure, by conceiving of the world and its happenings as composed of a systemic and deterministic component plus a stochastic, random component.

This paradigm of statistical thinking, when applied to economics, has led to the discovery of many regularities in behavior across millions of apparently self-willed, free-acting individuals. For example, Wal-Mart has sufficiently modeled the demand of products and arranged its supply in a way to maximize profits. Demand of, say, crackling cornflakes is a product of millions of people deciding to wander to Wal-Mart, remember they need breakfast in the morning, examining a great wall of various cereals, and ultimately deciding that what they need is a certain brand of noisy cornflakes. The act of a single hungry shopper may be erratic, but in aggregate, the demands of the average American can be sufficiently modeled and predicted.

To perform feats such as these, economists build models to understand and forecast time series data.

7.2 Econometrics before 1980

Econometrics before 1980 was largely a manual, slow, and heuristic affair. Computational power proved to be an extremely limiting factor: a single univariate regression might take dozens of hours to calculate on a medium-sized data set. As a result, econometricians entered variables into models with great restraint in order to minimize complexity. This, of course, had the effect of introducing significant bias into models: a model can only infer within the limits of data provided to it. Typical models ranged from simple univariate time series models with a single variable to larger "systems of equations" models with hundreds of equations, to single deterministic equation models that focused on interactions of few variables chosen by the economist. The limitations of these methods was a limiting factor in the progress of the field of economics, and discontent with the available models grew throughout the 1970s.

7.3 The Advent of Vector Autoregression

In the seminal, grandly titled paper "Macroeconomics and Reality" (1980), Christopher Sims voiced the limits and pitfalls of the most commonly used methods in econometrics. Most troubling, he writes, is the amount of *a priori* "knowledge" that econometricians bake into their models [Sims, 1980]. An important and unavoidable choice that economists and statisticians must make when constructing models is choosing endogenous and exogenous variables. *Endogenous* variables are explained, influenced by, and influence other variables in the system. *Exogenous* variables are independent from other variables being examined and may be excluded from the model. Excluding variables simplifies the model and decreases complexity and computational cost. Sims writes that "if every variable is allowed to influence every other variable with a distributed lag of reasonable length, without restriction, the number of parameters grows with the square of the number of variables and quickly exhausts degrees of freedom" [Sims, 1980].

However, the knowledge to decide that a variable is exogenous is a priori, determined by prior knowledge rather than the model at hand. These forms of "knowledge" are inaccurate assumptions at best, as theories guiding economies and policy can swing wildly and defy our "knowledge" and expectations, as recent history is a litany of "things the experts got wrong" (for evidence of this, we must merely look at the instability in economics in 2001 and 2008, and politics in 2016). At worst, these forms of "knowledge" encode biases and inequity into the models. In a paper entitled *Vector Autoregression and the Study of Politics*, John Freeman summarizes that "VAR modelers assume that our theories are relatively underdeveloped, or that our understanding of social reality is severely limited. VAR modelers maintain that we have a poor understanding of how variables are related and that our intuitions in this regard should not be trusted. Theories are loose collections of causal claims about the existence and direction of certain relationships. Only a weak set of restrictions about what variables to include in the model and about the contemporaneous relationships among variable innovations are theoretically justified" [Freeman et al., 1989].

As an alternative, Sims suggest that “it should be feasible to estimate large-scale macromodels as unrestricted reduced forms, treating all variables as endogenous. Of course, some restrictions, if only on lag length, are essential, so by ‘unrestricted’ here I mean ‘without restrictions based on supposed a priori knowledge”’ [Sims, 1980]. This is the fundamental insight that led to the development of vector autoregression. Sims then outlines a small manageable example and demonstrates the early stages of the methodology that will become central to VAR methods. In his conclusion, he writes “a long road remains, however, between what has been displayed here and models in this style that compete seriously with existing large-scale models on their home ground-forecasting and policy projection.... But though the road is long, the opportunity it offers to drop the discouraging baggage of standard, but incredible, assumptions macroeconometricians have been used to carrying may make the road attractive” [Sims, 1980]. This road eventually leads to the methods of vector autoregression. This road has proved fruitful and indeed, incredibly attractive. Since this publication, vector autoregression has become the dominant tool used in American empirical macroeconomics [Demiralp and Hoover, 2003].

In this paper, we apply the same tools to different ends. There is sparse existing literature on applying vector autoregressive methods outside the domain of economic data. Our work implies that these methods are equally successful in describing time series data from other domains, specifically high technology.

Chapter 8

Conclusion

A large part of the project was operationalizing the intuition behind how a ‘root cause’ in a system malfunction is defined. We have had reasonable successes in using the impulse response function to determine a ‘root cause.’ This methodology has a large range of application beyond latency and site error data, and extends to any domain that utilizes interacting time series.

Vector autoregression serves as a useful starting point in understanding time series and communicating insights to a general audience. P-values for variables in the regression, for example, help narrow the set of related metrics as we have seen. Impulse response coefficients can quantify the size of ‘root causes.‘ The methodology is flexible enough to be applied to any set of time series data.

Bibliography

[Box and Jenkins, 1990] Box, G. E. P. and Jenkins, G. (1990). *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated.

[C,] C, J. G. State of the climate 2008.

[Demiralp and Hoover, 2003] Demiralp, S. and Hoover, K. D. (2003). Searching for the causal structure of a vector autoregression*. *Oxford Bulletin of Economics and Statistics*, 65:745–767.

[Freeman et al., 1989] Freeman, J. R., Williams, J. T., and min Lin, T. (1989). Vector autoregression and the study of politics. *American Journal of Political Science*, 33(4):842–877.

[Haugh, 1976] Haugh, L. D. (1976). Checking the independence of two covariance-stationary time series: A univariate residual cross-correlation approach. *Journal of the American Statistical Association*, 71(354):378–385. Again, the paper's title is exactly what we are looking to do for the clinic project. The methods in this paper agree with the previous Quenouille paper. Both papers fit univariate models on each of the two series, and then seeing if there is a correlation between the two series of residuals.

[Hong, 1996] Hong, Y. (1996). Testing for independence between two covariance stationary time series. *Biometrika*, 83(3):615–625.

[Pesaran,] Pesaran, M. H. Econometrics. *The New Palgrave Dictionary of Economics*, page 1.

[Podobnik and Stanley, 2008] Podobnik, B. and Stanley, H. E. (2008). Detrended cross-correlation analysis: A new method for analyzing two nonstationary time series. *Phys. Rev. Lett.*, 100:084102. This paper also analyzed correlations between two time series, but in the case where they are non-stationary. The previous papers examined only stationary time series. Both methods will be examined. First, of course, I have to actually get the data from Microsoft and see whether it is stationary or not.

[Quenouille, 1949] Quenouille, M. H. (1949). Approximate tests of correlation in time-series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(1):68–84. This paper has an ostensibly useful title (indeed, its title is exactly the request from the clinic liaisons). It is a somewhat dated paper at 1949, so newer and more elegant, computational, or sophisticated methods may have arisen. The level of mathematic exposition is somewhat beyond my comprehension at this point, since I just began study of the field earlier this week when the clinic changed direction to time series analysis.

[Shumway and Stoffer, 2017] Shumway, R. H. and Stoffer, D. S. (2017). *Time series analysis and its applications: with R examples*. Springer. This is an introductory textbook on time series analysis. As such, it explains the field of time series analysis and its common applications and modifications. It also includes code and examples in R. It has been cited over 3,500 times (according to Google Scholar), and seems to be a widely-respected starting point for the field of time series analysis. I'll be reading it to gain a background on the subject, and reference it as a starting point for discussions in my thesis.

[Sims, 1980] Sims, C. A. (1980). Macroeconomics and reality. *Econometrica*, 48(1):1–48. A seminal paper first advocating for VAR in econometrics.