UNIVERSITY OF CALIFORNIA

Los Angeles

Time Series Analysis
on AMP-enabled Website
Advertisement Revenue

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Statistics

by

Boyu Zhang

2019

ABSTRACT OF THE THESIS

Time Series Analysis
on AMP-enabled Website
Advertisement Revenue

by

Boyu Zhang
Master of Science in Statistics
University of California, Los Angeles, 2019
Professor Frederic R. Paik Schoenberg, Chair

AMP is a web page format that Google has developed for optimizing search results and mobile display. In this thesis, we will analyze the daily data for revenues, pageviews, and RPMs of a particular website that uses both AMP and regular web pages from 2016 to 2018. After data cleaning and transformation, we will utilize spectral analysis, regressions, multiplicative decomposition using LOESS, ARIMA modeling and other methods to analyze the time-series data, separate the seasonality and trend components, and build two different models and generate predictions. We will also analyze the specific impact of suspending AMP-enabled pages on the advertisement revenue of a website, and provide casual inference on the behavior of total revenue immediately after suspending AMP-enabled pages.

The thesis of Boyu Zhang is approved.

Hongquan Xu

Ying Nian Wu

Frederic R. Paik Schoenberg, Committee Chair

University of California, Los Angeles

2019

To my dear family and friends . . .

who have offered a tremendous amount of distraction

while I was trying to finish this thesis

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Time series analysis is one of the most commonly-utilized statistical tools in the real world. In the field of digital media, it is crucial for the management team of a website to learn from historical data and establish prediction models that can be used to provide scenario testings and support business decisions. This can be especially hard when new features are constantly rolled out by both the websites and the platforms they are built on. For example, Google has developed the Accelerated Mobile Pages(AMP)' format for web pages, which are optimized for displaying on mobile devices and mobile Google searches.[1]

In this analysis, we will establish time series projection models for a website that both has these AMP-enabled web pages and regular ones, and attempt to gain a better understanding of the impact that AMP-enabled web pages have in terms of total pageviews and total revenues.

CHAPTER 2

Background and Raw Data

2.1 Background Information

This analysis will focus on the traffic and revenue data from one particular news website, which will be referred to as 'The Website' from now on. The Website was originally set up only in the normal HTML format back in 2010; On 07/12/2017, The Website started adding AMP (Accelerated Mobile Pages) pages for their new articles, which is a Google-developed web page format optimized for mobile display and Google searches (For convenience, the normal HTML format will be called 'WEB pages', and the AMP-enabled version will be called 'AMP pages'). AMP pages generally load faster than WEB pages on mobile devices, show up at the very top of mobile Google search results, and are prioritized by apps like Apple News. However, WEB pages offer much more freedom in terms of page design and advertisement placement, while the AMP format is somewhat restrictive in that aspect.

The Website ran both WEB and AMP pages until 08/16/2018, when it stopped providing AMP versions for their new articles. The decision was made mainly based on Revenue Per Mille (RPM) performance, which is an important indicator in digital marketing that measures the average revenue per 1000 impressions. The website's team observed that the RPM for WEB pages are higher than that of AMP pages, probably due to the fact that each WEB page has 4 advertisement spots while each AMP page only has 3. The team expected the total pageviews to drop after stop using AMP pages; but since now all pageviews would go to WEB pages, which had higher RPMs, it was argued that the total revenue may not decrease as a result. Also, WEB pages' flexibility in page arrangement are much more suitable for elaborate marketing campaigns. Unfortunately, the revenue did drop in August and September after stopping using AMP, so The Website resumed using both WEB and AMP again in 10/25/2018.

2.2 Definitions

The setup of this project is relatively complicated and requires some industry-specific knowledge. A summary of key terms used in this thesis is provided below:

The Website: a news website, whose pageviews and website data I have access to;
The Team: the team in charge of managing The Website; in most cases this refers to the team member that tracked data trends;
AMP page: a version of the web page optimized for mobile display and Google's mobile searches; it contains 3 advertisement slots per page, and is generally more restrictive in page design;
WEB page: the normal HTML version of the web page, which is not optimized for mobile devices and generally loads slower comparing to the AMP version; it contains 4 advertisement slots per page, and offers more freedom in page design;
Pageview: number of times a particular page is visited and loaded;
Impression: number of times a particular advertisement is loaded;
RPM: average revenue per 1000 impressions;

2.3 Goals

The Website's team wanted to figure out why their initial expectation was not achieved, and how much of the decrease in revenue from August to September is due to the pause on AMP pages. The two main goals of this analysis are:

1. Build a time-series model based on the historical data from 2016 to 2018 that can be used for predictions and scenario testing;

2. Analyze the data when AMP was paused, and provide insight on whether it is possible to keep a stable revenue when potentially pausing AMP next time for major marketing campaigns.

2.4 The Dataset

The dataset I was provided with covers the daily traffic and from 01/01/2016 to 10/31/2018. The raw dataset is a 119245 × 6 matrix. The 6 columns are:

DATE: dates from 01/01/2016 to 10/31/2018, ordered in a sequence from 2016.00000 to 2018.830137;
DEVICE: a label for which device the visit was registered on; mobile for mobile phones, ipad for Apple iPad, NA for devices that failed to be identified, and non-mobile for visits to the mobile version of the webpage from a non-mobile computer;
ORDER: advertiser and marketing campaign name responsible for the particular advertisement; recorded as strings;
TYPE: a label for the version of webpage that particular advertisement was displayed on; either 'AMP' or 'WEB';
VIEWS: impression counts for that particular advertisement, recorded as integers;
REV: revenue (in USD) generated from that particular advertisement, rounded to 2 decimal places;

Since The Website started using AMP pages in 07/12/2017, there are no AMP entries before that date. The dataset is imported into R for further analysis through a .cvs file.

2.5 Observations before Analysis

Before starting the data analysis project, I talked extensively with the team member that tracked data trends for The Website. He is a veteran in the digital marketing business, and has shared with me his insight into the inner workings and decision making process of the digital media industry. He provided the following observations about the data:

1. He believed that very strong seasonality exists in the pageview counts of The Website. He was very convinced that there would be a weekly cycle, and also thought there would probably be a monthly cycle and an annual cycle as well;
2. He told me that the data for pageviews and revenues in September 2018 had been significantly impacted by an one-time event. An important event in the industry occurred on 09/21/2018, which had driven an 'unusually high amount of traffic' to The Website. He believed that impact was contained within the month of September, and that I should try to remove the effect of this event before establishing the seasonality;
3. From previous experience, he believed that the RPM for WEB pages would be generally higher than that of AMP pages; he thought the difference here was mainly caused by the AMP pages' restriction in page design; he believed that since WEB pages were aesthetically better designed and accentuated the advertisements more, the readers were more likely to click on the ads and generate more revenue for The Website, resulting in higher RPM.

It is important to clarify that none of these observations will be used as assumptions for my analysis, as they were mostly based on empirical knowledge rather than analysis on the actual dataset. However, they do provide good starting points in terms of cleaning and transforming the dataset, and could be very useful when drawing conclusions from my own data analysis.

CHAPTER 3

Data Cleaning and Transformation

After importing the raw data into R, the first step was to observe the data. Since we were only interested in the effect of AMP pages on mobile devices, 76 entries with the 'non-mobile' label in DEVICE were removed. After discussion with The Team, I also removed all 6 entries with the 'ipad' label, and 4 entries with the 'NA' labels. In total, 86 entries were removed because of the category, which constitutes less than 0.1 percent of the raw data, and would not have any significant impact on the model.

For the next step, I grouped the data into daily totals. Using daily data as the increment seems to be the most appropriate measure in this analysis, as the weekly cycle was supposed to be very strong according to the team, and the revenues for each advertisement slot was calculated daily.

By adding up the pageview and impression numbers with the same DATE number, I generated a new matrix of data, now with 7 columns:

DATE: dates from 01/01/2016 to 10/31/2018, ordered in a sequence from 2016.00000 to 2018.830137 with each date only appear once; 1035 entries in total;
A_VIEWS_W: total impression counts for all the advertisements displayed on WEB pages on that day, recorded as integers; 1035 entries in total;
A_REV_W: total revenue (in USD) generated from all the advertisements displayed on WEB pages on that day, rounded to 2 decimal places; 1035 entries in total;
A_VIEWS_A: total impression counts for all the advertisements displayed on AMP pages on that day, recorded as integers; 1035 entries in total; 447 entries in total;
A_REV_A: total revenue (in USD) generated from all the advertisements displayed on AMP

pages on that day, rounded to 2 decimal places; 447 entries in total;
A_VIEWS_T: total impression counts for all the advertisements displayed on that day, recorded as integers; 1035 entries in total; =A_VIEWS_W + A_VIEWS_A;
A_REV_T: total revenue (in USD) generated from all the advertisements on that day, rounded to 2 decimal places; 1035 entries in total.

Since The Website started using AMP on 07/12/2017, all entries for A_VIEWS_A and A_REV_A before 'DATE' 2017.52603 (07/12/2017) are missing. To fit the dataset into one matrix, we will replace these missing values with zeros. The effect of this strategy will be discussed later in this thesis.

By observing the data, I noticed that several A_VIEWS_W entries in February 2016 are extremely high. After tracking the outliers from the daily sums, I isolated 4 entries in the raw data that are more than 100 times higher than the other entries. They also all have 'T mobile campaign' in the ORDER entries. After communicating with The Team, we confirmed that these are data entry errors, and I decide to exclude these 4 entries when calculating the daily total impressions.

After finishing the data cleaning process, we can start working on adding more variables.

Since RPM is an important parameter in digital marketing and contributed significantly to the decision to stop using AMP, I will add the average daily RPMs as new variables in this analysis. There are 4 different RPMs for each day that we can calculate; we will add them all into the matrix as 4 new columns:

A_RPM_W: average revenue per 1000 impressions of an advertisement on WEB pages in that particular day;
P_RPM_W: average revenue per 1000 impressions of a WEB page in that particular day;
A_RPM_A: average revenue per 1000 impressions of an advertisement on AMP pages in that particular day;
P_RPM_A: average revenue per 1000 impressions of an AMP page in that particular day;

It is important to discuss the difference between ‘A_RPM’ and ‘P_RPM’ before moving on with our analysis. In the case of WEB pages, for examples, A_RPM_W calculates the average RPM of an advertisement in that day, while P_RPM_W calculates the average RPM of a web page in that day. Thus the actual P_RPMs should be calculated as:

$P\_RPM\_W = \frac{WEB\ Revenue}{WEB\ pageviews} * 1000$
$P\_RPM\_A = \frac{AMP\ Revenue}{AMP\ pageviews} * 1000$

But unfortunately, the ‘pageview’ data I had access to does not distinguish between AMP pages and WEB pages, and only provides a total pageview count. This means that we cannot analyze the RPMs for AMP and WEB pages separately, rendering the whole analysis ineffective. To solve this problem, we will approximate the pageview numbers by adding these two new variables:

$Pageview\_W \approx \frac{A\_VIEWS\_W}{4};$
$Pageview\_A \approx \frac{A\_VIEWS\_A}{3};$

This is based on the fact that there are 4 advertisements on each WEB page, and 3 advertisements on each AMP page. However, there are several scenarios in which this assumption will not hold:

1. The reader closes the web page before seeing all the advertisements; this will make $Pageview\_W > \frac{A\_VIEWS\_W}{4};$
2. With poor internet connection, the reader refreshes the web page, loading new advertisements without actually seeing the previous ones; this will make $Pageview\_W < \frac{A\_VIEWS\_W}{4};$
3. The reader uses some forms of browser plugin that blocks the display of advertisement on that web page; this will make $Pageview\_W > \frac{A\_VIEWS\_W}{4};$

I have consulted with The Team about these concerns, and they believed that these will not affect the behavior of RPM in a significant fashion. The first two cases are impossible to avoid with the current technology, and the third case is extremely rare, since advertisement-blocking plugins on mobile devices are very niche. This approximation, though still a compromise, is the most reasonable method in calculating RPMs. The Team had suggested to build the predictive model on the total pageviews first, so we will add it as a new variable too.

Thus the RPMs for each version of the web pages are calculated as:

$$P\_RPM\_W = \frac{4 * A\_REV\_W}{A\_VIEWS\_W} * 1000;$$

$$P\_RPM\_A = \frac{3 * A\_REV\_A}{A\_VIEWS\_A} * 1000;$$

Note that since DVIEWS_A contains zeros before 07/12/2017, the definition of A_RPM_A and P_RPM_A are not applicable. We will manually set them to 0.

CHAPTER 4

Preliminary Analysis

We will start by plotting all variables and make some preliminary observations. In the following plots, the dashed vertical lines mark the end of every year, while the red vertical lines mark the three relevant event on the time-line: from left to right, they respectively mark the date on which AMP pages started to be in use (07/12/2017), the date on which AMP pages were paused (08/16/2018), and the date on which AMP pages were resumed again (10/25/2018). Data related only to the AMP pages will be plotted with blue lines, and data related to the WEB pages will be plotted with green lines. The totals will be plotted with black lines.

First, we will take a look at the total impressions and pageviews data:

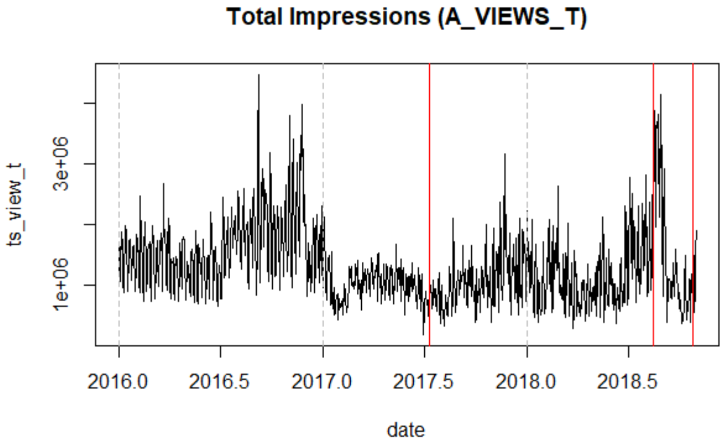


Figure 4.1: Total Impressions

As we can see here, the total impression seems to be going slightly downwards overall, but sharply increases starting around June 2018. The impression numbers seem to fluctuate around 1,500,000 in 2016, with a very high peak in the second half year that reaches over 4,000,000. In 2017, however, the impression numbers center around 1,000,000, with a much smaller peak during the second half. 2018 starts relatively at the same level as 2017, but shows a much strong peak starting in June.

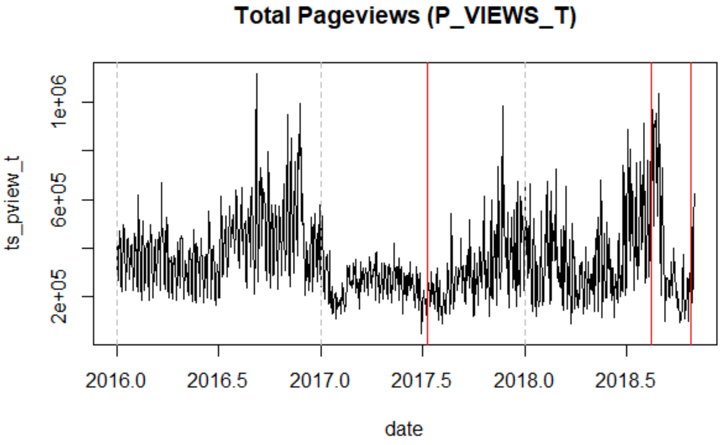


Figure 4.2: Total Pageviews

However, the total pageview plot shows that the actual viewership, though constantly fluctuating, is not declining as a general trend; and, unlike the total impression plot, the numbers in 2017 and the first half of 2018 seem to be roughly at the same level as that of 2016. This is likely caused by the fact that each AMP page has one fewer advertisement slot comparing to the WEB page. Before the second half of 2017, only WEB pages are available to the readers; this means that during that time, we have:

A_VIEWS_T = 4×P_VIEWS_W = 4×P_VIEWS_T;

after AMP pages were available, we have:

A_VIEWS_T = 4×P_VIEWS_W+3×P_VIEWS_A < 4×P_VIEWS_T;

Thus even though the total pageviews are roughly at the same level, the total impressions will still decrease.

We also notice that there is very strong seasonality involved in these data. A relatively short cycle can be clearly identified even without any quantitative analysis, which seems to be the weekly cycle that was mentioned by The Team. A monthly cycle cannot be easily recognized, but an annual cycle seems to exist, with pageviews going relatively stable in the first 6 months of the year, and peaking around September or October. But since the data only covers fewer than 3 years, it is very likely that the perceived annual cycle is just an coincidence.

There are also multiple outliers with very extremely values in the dataset for pageviews. A quick search shows that there are 20 entries that are higher than 800,000, and 2 entries that are higher than 1,000,000. One of the highest entry corresponds to the 'one-time event' that was mentioned by the team, who suggested that I should remove this outlier for better performance; but we can see now that these high peaks occur relatively often, and thus should not be removed.

Now we will compare the performance of AMP pages and WEB pages in terms of impressions and pageviews:

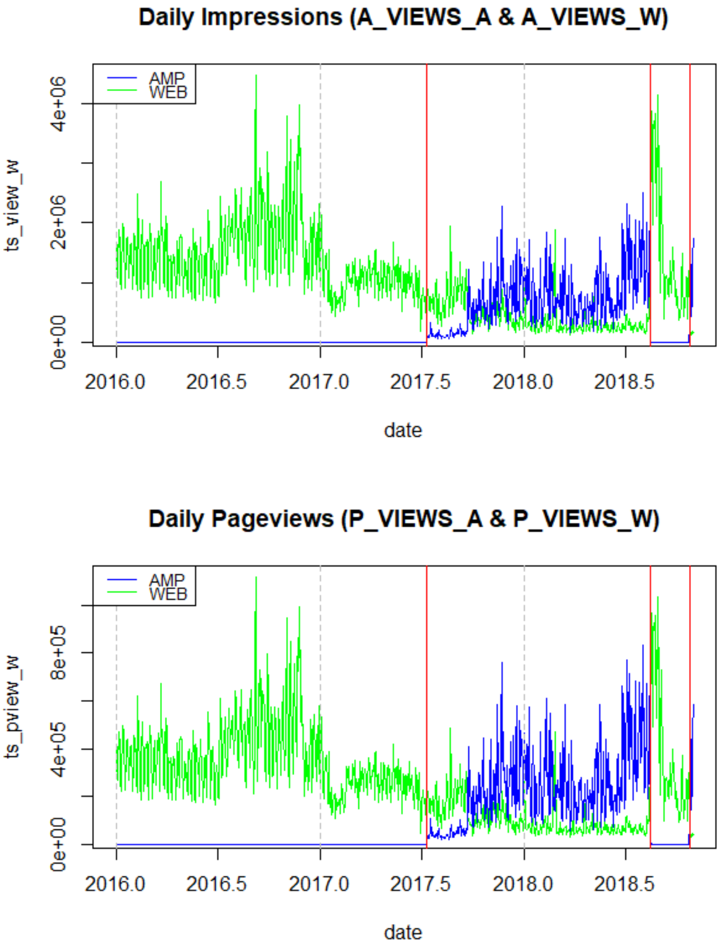


Figure 4.3: WEB and AMP Impressions (top) and Pageviews (bottom)

We can see that in both plots, WEB pages' impression/pageview number is decreasing, while AMP pages' is increasing; this is to be expected, as they share the same potential population of viewership. As the technology in smart-phones and high-speed wireless service develops in recent years, more and more users are using mobile devices to perform Google searches. Since AMP pages are optimized for Google searches, they will appear at the top of the search results instead of the WEB version, taking more and more viewership from the WEB version. Because of the way we are calculating pageviews, the AMP line is the pageview plot is a vertically stretched version of the AMP line in the impression plot, showing that the growth in AMP page viewership is actually faster than what it seems to be when looking at the raw data for impressions.

We then move onto the revenue plot:

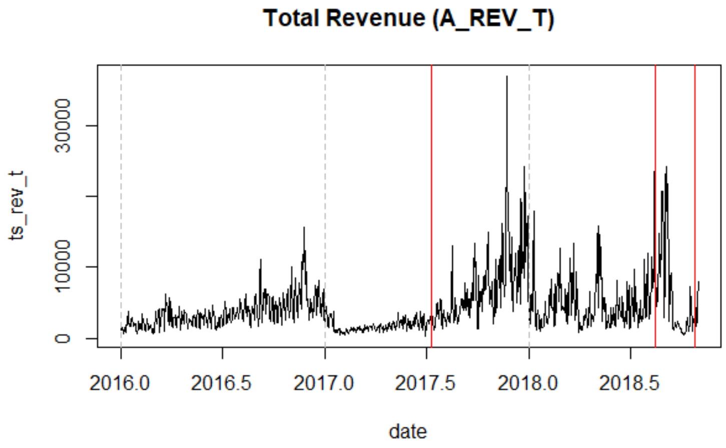


Figure 4.4: Total Revenue

The total revenue plot shows an increasing variance as the time progresses, which means we will probably need to detrend the data before further analysis. We also can notice that seems to be a relatively short cycle that is very strong, which should also be removed. An annual cycle may also be present, but there are only 2 complete years of data and it is hard to determine.

We then can check the revenue from WEB and AMP pages separately:

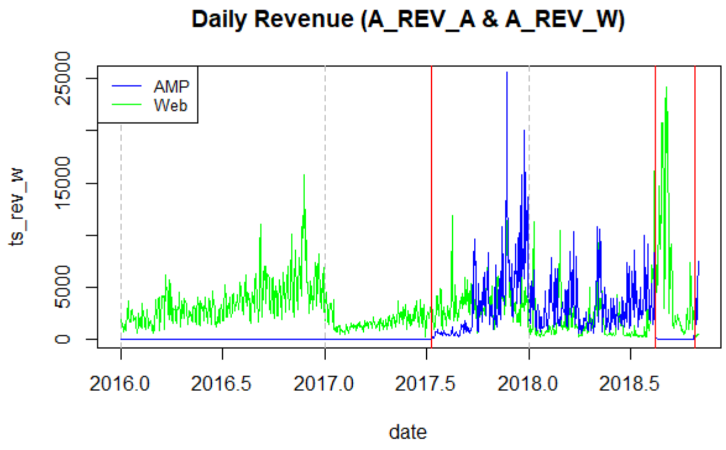


Figure 4.5: WEB and AMP Revenues

Both sets of data seems to have the strong cycles with a relatively short period, and both does not show any dominating upward or downward trend throughout the 3 years. After the detrending process, Both the total revenue data and the revenue data for each version of the web page should be good predictors to build the prediction models on.

Finally, we check the plot for both the advertisement RPMs(A_RPM) and the page RPMs(P_RPM):

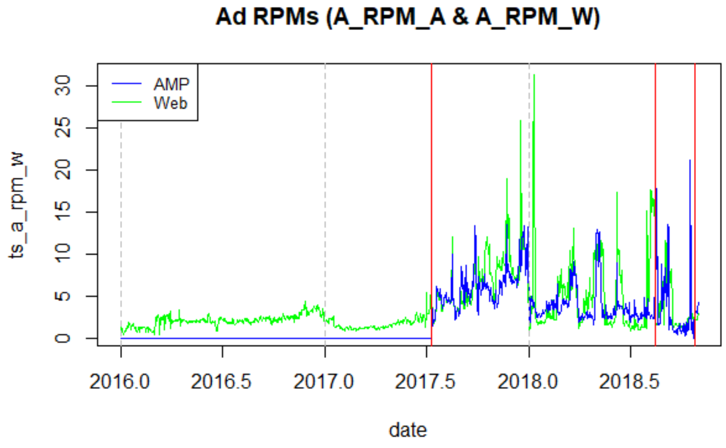


Figure 4.6: WEB and AMP Advertisement RPMs

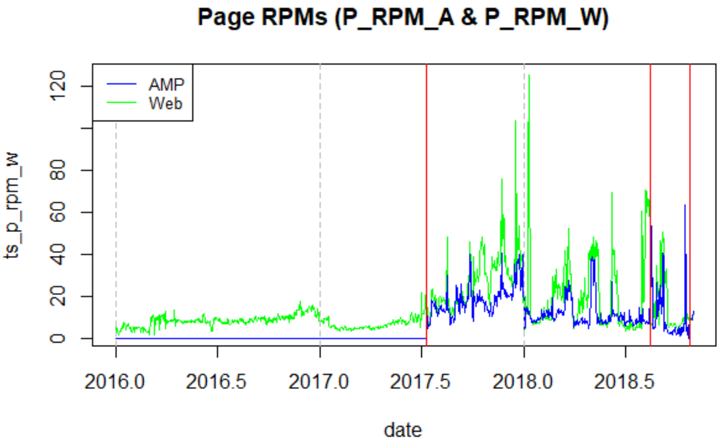


Figure 4.7: WEB and AMP Page RPMs

It is interesting to note that if we look at the ad RPM, The Team's claim that WEB pages have higher RPMs are actually false; when both versions were active, the RPMs for WEB pages is not constantly higher than the RPMs for AMP pages. Though it does seem that WEB RPMs have a much bigger variance, and more extremely values, which may be a indication that the viewership of the two versions of web pages behave differently.

In terms of trend and seasonality, it is very hard to recognize any cycle or general trend in the data. The variance for WEB RPM also seems to dramatically increase after AMP pages were implemented, showing that the data is definitely not stationary without detrending. It seems that RPMs are not a very good predictor to build our prediction models on.

CHAPTER 5

Choice of Models

One of the goals of the analysis is the prediction of revenue, and there are multiple potential approaches to build the predictive model. In this thesis, we will test out two different approaches, and compare the results afterwards:

Model I: We will use one single variable, the total revenue, to build the predictive model.
Model II: We will build 2 independent predictive models for AMP revenue and WEB revenue respectively, and add up the prediction afterwards.

Model I is the most simple and straightforward in structure, but Model II provides much detailed parameters and can be easily tweaked to simulate various scenarios. The Team showed the most interest in Model II, since building a model based on different versions of the web page would probably provide them with more insight into the management of the website. In this thesis we will test these models accordingly, and compare the results afterwards.

CHAPTER 6

Model I: Using Total Revenue As the Only Predictor

6.1 Detrending

Spectral Analysis

The first step to build this model is to separate the trend from the seasonality, so we can apply further analysis on the residuals.

We start by transforming all relevant data into time series. Checking the ACF and PACF plot for total pageviews and we can see that there is a high possibility that there is a cycle of 7 days; this observation matches The Team's assumption that there would be a very strong weekly cycle.

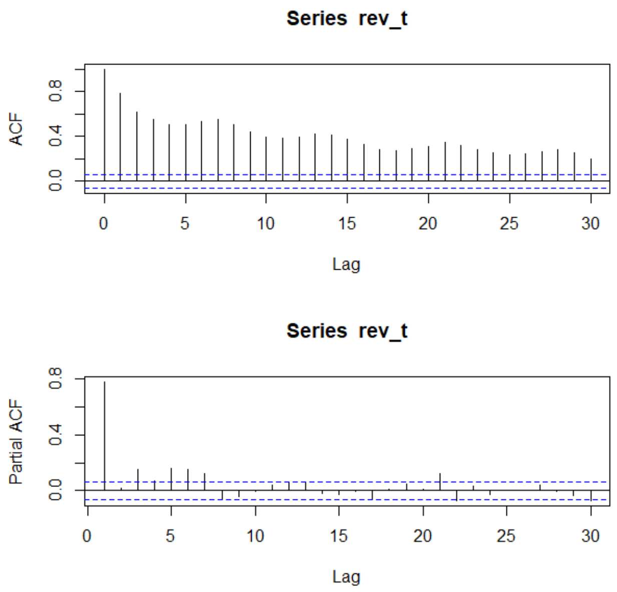


Figure 6.1: ACF and PACF for Total Revenue

We can also perform spectral analysis on the total revenue data to determine a possible cycle. First we'll look at the raw periodogram using the 'mvspec()' function. The periodogram shows the most dominant frequencies, after transforming the data into a combination of cosine waves with various amplitudes and frequencies.[2] In this case, we can see that the values starts very high when frequency is low, and quickly decreases with fluctuation to a relatively small value. We also notice that there is a sudden increase in values around 0.14.

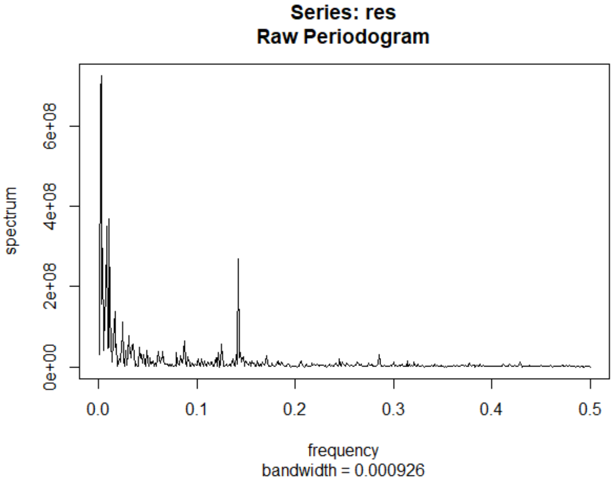


Figure 6.2: Raw Periodogram for Total Revenue

To better analyze the periodogram, we will apply a Daniell kernel with 6 dimensions to smooth the plot, using the 'kernel()' function. The Daniell kernel is a method of applying kernel onto vectors, matrices, and, in this particular case, time series data. It utilizes a two-sided average and the Daniell Spectral Window $W_m(k)$:[3]

$$W_m(k) = \frac{1}{2m+1} for - m \leq k \leq m.$$

The Smoothed Periodogram is shown below:

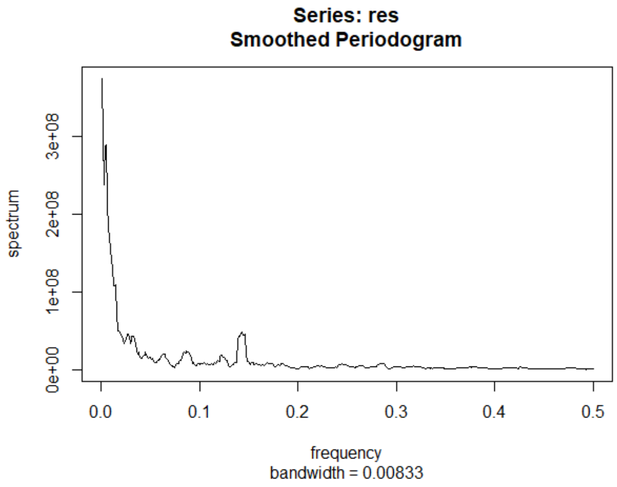


Figure 6.3: Smoothed Periodogram for Total Revenue

We can see that the spectrum values drop linearly at the beginning, and becomes very stable after 0.05 frequency. But there is a peak at around 0.14, where the spectrum values increases and then drops sharply.

We can get a further smoothed version of the raw periodogram by checking the AR plot, generated wit the ‘ar()’ function. This function applies autoregression to the time series data, and chooses the frequency by choosing the largest Akaike information criterion (AIC) values.[4] AIC is based on the information theory, and measures the amount of information that is lost by the model.[5] The plot is shown below:

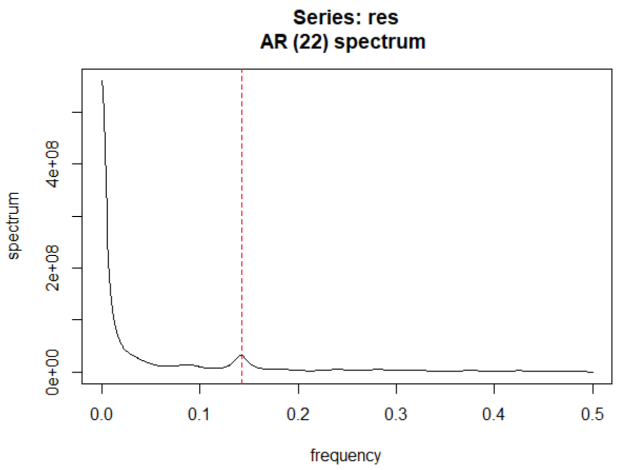


Figure 6.4: AR Plot for Total Revenue

The AR plot tells the same story as the raw and smoothed periodogram, and confirms that even after smoothing, the high values at around 0.14 frequency is still very high. Converting the peak frequency to period, it turns out to be a cycle of 7.028169 days, which is extremely close to a weekly cycle. Thus for our first step in the detrending process, we will set a period of 7 days.

Weekly Cycle

First, we convert all data into the time series format, with a period of 7. Then we will attempt to separate the trend and seasonality by applying classical seasonal decomposition by moving averages to the time series through the R function 'decompose()'. This function first generates a 'trend' component through moving averages in a symmetric window with equal weights, and removes it from the time series. Then the 'seasonal' component is generated through averaging each time unit over all the period. The last component, 'random', is the residual after removing both 'trend' and 'seasonal'.[6]

There are two ways to remove each component:

1. Additive decomposition, where $data = trend + seasonal + random$;
2. Multiplicative decomposition, where $data = trend * seasonal * random$;

In this case, since we are analyzing a website's revenue growth, the seasonality is more likely to be affected by the trend exponentially; thus, it makes more sense to use the the multiplicative decomposition model.

In the case of the total pageview, for the example, the three components after applying multiplicative decomposition are shown below:

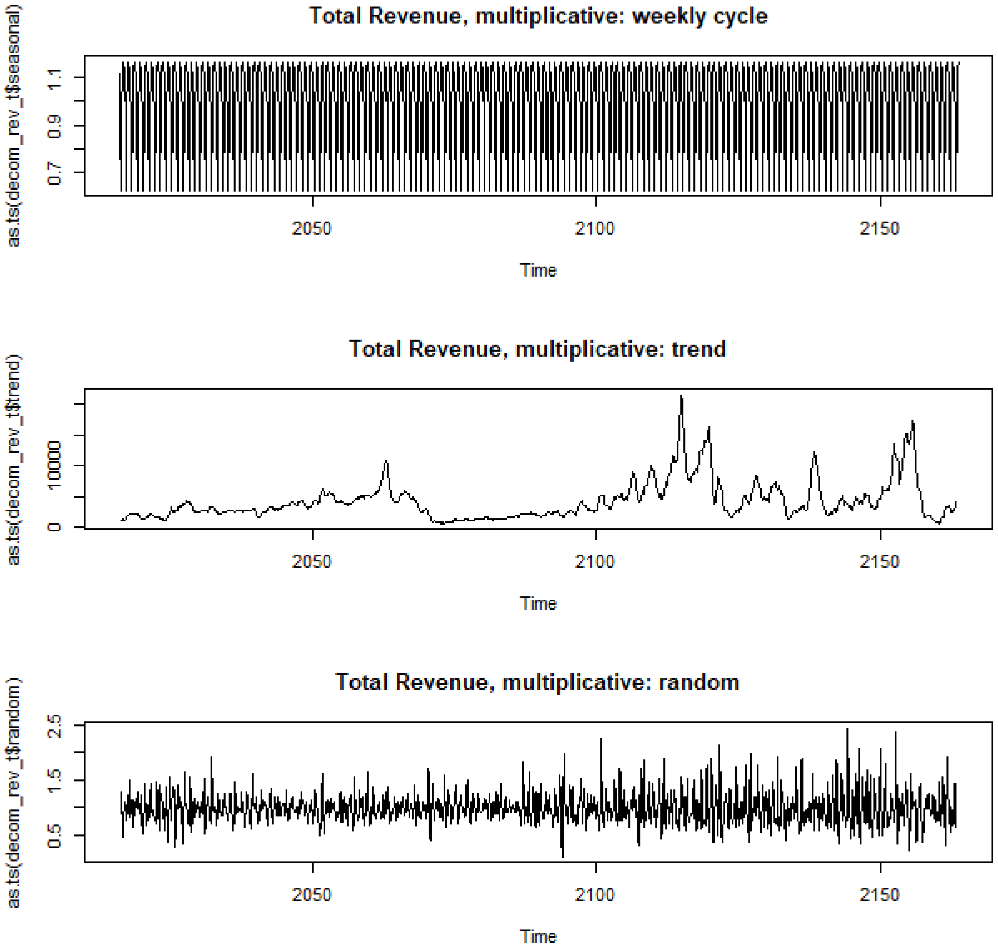


Figure 6.5: Multiplicative Decomposition on the Total Revenue

We can see that the weekly cycle is indeed very strong in the data; and after separating the seasonality and trend, the residual looks reasonably random. However, we notice a problem when trying to check the ACF and PACF plot for the residual: the first 3 and the last 3 entries of the residual are ‘NA’. Upon further inspection, this is not caused by any problem in the dataset. The ‘decompose()’ function uses moving averages in a symmetric window with equal weights, and thus it can only be applied when there are at least half of the cycle on either side of the data. We chose a cycle of 7 in this case, so the first 3 and the last 3 entries cannot be fitted with a trend with the function.

This is, in most cases, not a significant problem, as we can simply discard the entries with ‘NA’ as long as we have a large enough dataset. In this case, however, we are particularly interested in the behavior of pageviews right before and after AMP pages were paused, and using the ‘decompose()’ will force us to discard the data for the last 3 days before the pause on AMP pages. Also, since there are also a monthly and an annual cycle involved in the data according to The Team, we will attempt to apply multiplicative decomposition again on the residuals; this will result in an even bigger loss of usable data around the end of the dataset.

To avoid these potential problems, we will test out some other detrending methods that preserve the entire dataset. After trying out the 7th derivative and the backshift operator, we found out that detrending using the ‘stl()’ function, which utilizes locally estimated scatterplot smoothing (LOESS) to separate the trend and seasonality, is the most optimal solution. LOESS is a common method for local polynomial regression, and fits the trend on each point locally, using points in the neighborhood of each point weighted by their distance from that particular point.[7] Thus LOESS provides a much more localized fit for the data point-by-point, without the need to define a global function. (REFERENCE!!!) It also preserves the entire length of data after the decomposition, since it does not require a symmetric window for each data. The ‘stl()’ function uses LOESS to smooth the remainder after removing seasonality. Then the overall level is removed from seasonality, and is added to the trend component. This process will be iterated several times.

The result of applying the ‘stl()’ function on the total pageview data, with a period of 7, is shown below:

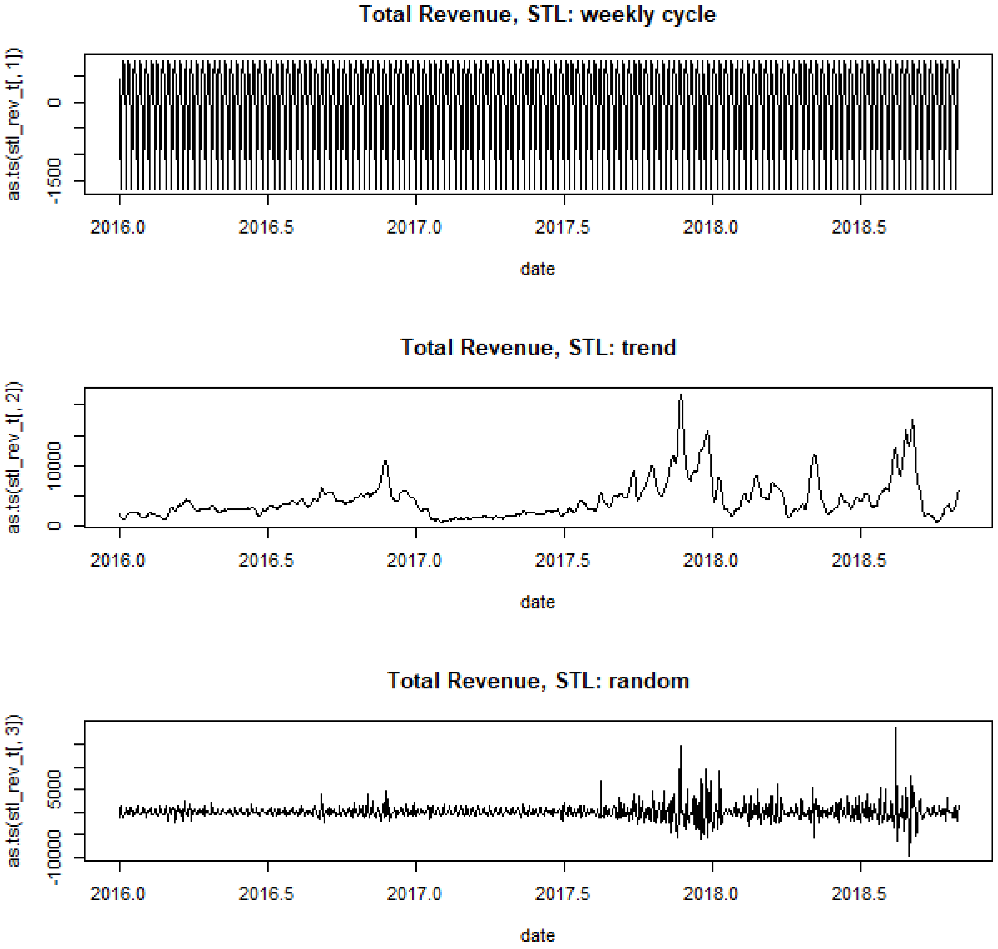


Figure 6.6: STL Decomposition on the Total Revenue

We can see that the trend and seasonality obtained from 'stl()' look very similar to the results of multiplicative decomposition with moving average, but the residual is now less random: the variance during the later half of 2017 is significantly higher than the rest. This is caused by the fact that 'stl()' is additive in nature; thus the seasonality was not adequately removed when the trend is very high. To deal with this problem, we will apply a log transformation to the data, and then use 'stl()' on the results:

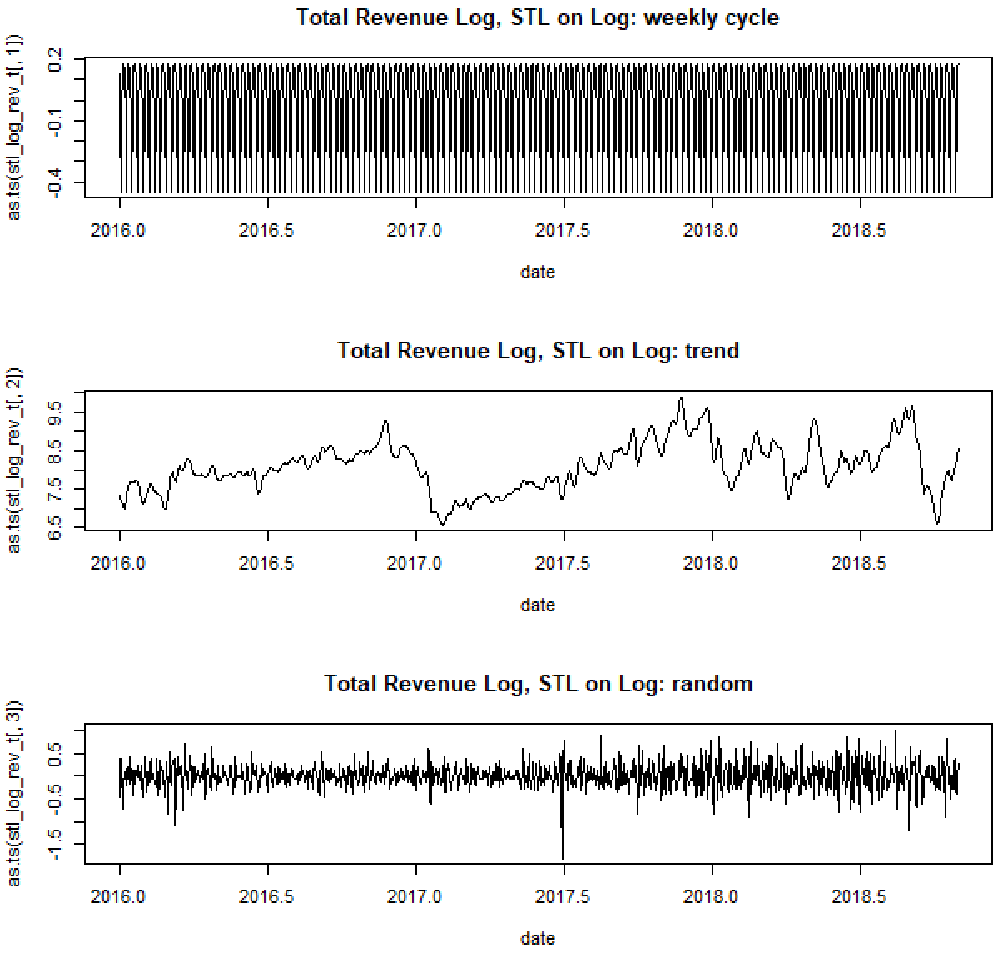


Figure 6.7: STL Decomposition on the Log Total Revenue

The data after taking the log are compounded additively, so when we undo the log and reconstruct the components, they will be compounded multiplicatively:

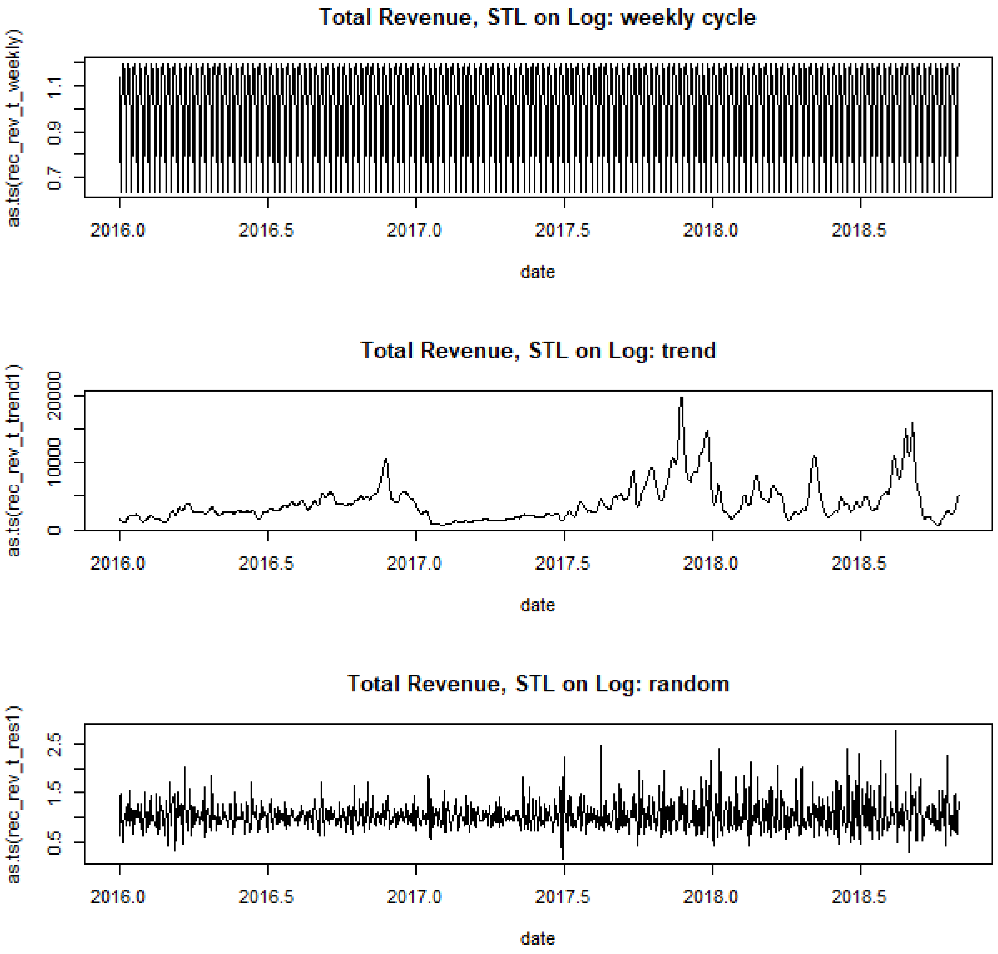


Figure 6.8: STL Decomposition on the Total Revenue, After Log Transformation

Now the residual looks much more random, and the variance remains generally uniform throughout the series, satisfying the stationary assumption. This is the best result among the 4 detrending methods we have tested, so we will use the 'stl()' function on the data after taking the log for all the other variables in this analysis.

Monthly Cycle

After removing the weekly cycle, we will now attempt to remove the monthly cycle. First we will check the ACF and PACF plots for the residual after transforming back from the log data:

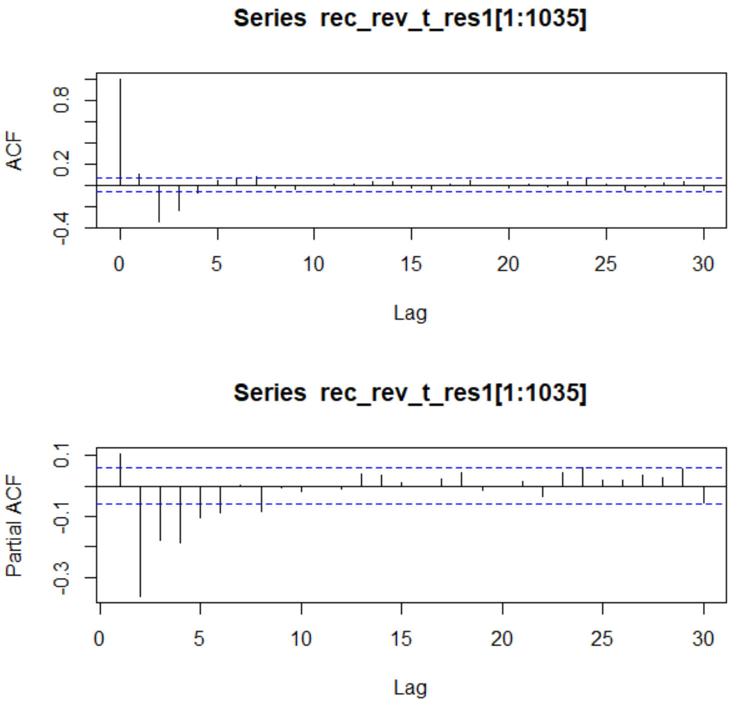


Figure 6.9: ACF and PACF for the Residual of Total Revenue

As the plots show, there's no clear indication of any possible period in Lag 30 or Lag 31. We can also check the raw and smoothed periodograms for the residual:

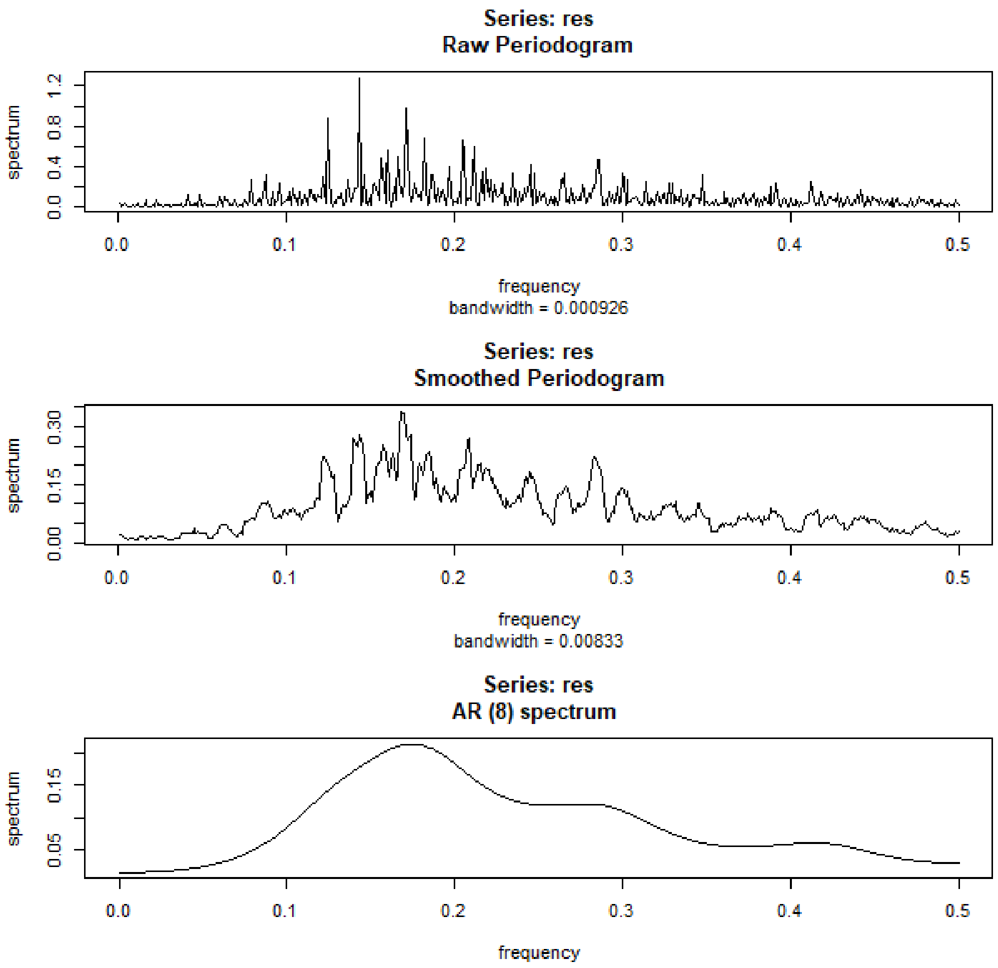


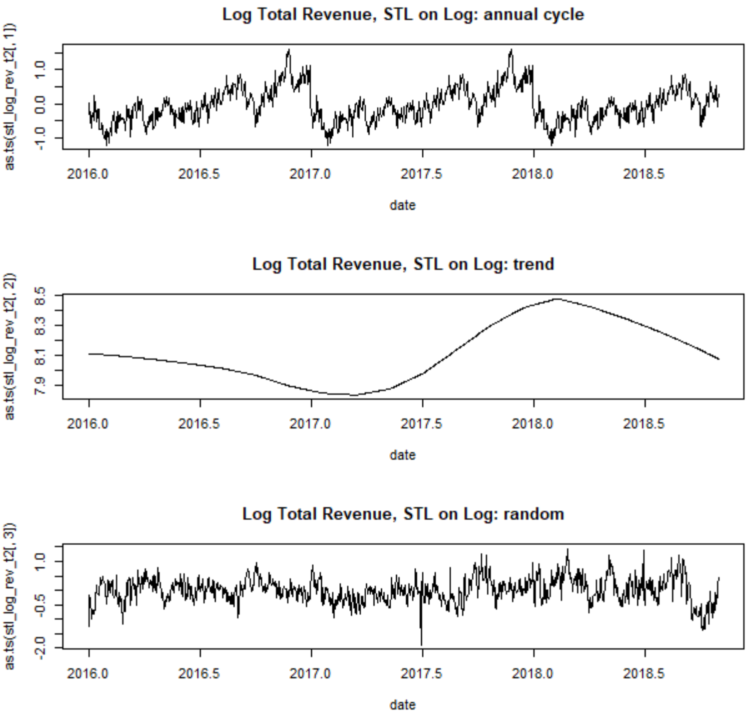
Figure 6.10: Raw Periodogram, Smoothed Periodogram, and AR Plot for the Residual of Total Revenue

The values at the beginning is very small, showing that the existence of a monthly cycle is not very likely. This appears to contradict The Team's first claim, so I conducted some further research and communicated with The Team. I had assumed that the monthly cycle could exist because important conference and new product announcement tend to happen at the beginning of the month. This belief, after some research, proves to be false, as there are no apparent clustering of dates within a month when important events in this particular industry happen. The Team also confirmed that they did not have any substantial evidence to support the existence of a strong monthly cycle; the idea was probably facilitated by the fact that the website's performance was evaluated at the end of each month. Thus we have no reason to believe that there is a monthly cycle and needs to be removed.

Annual Cycle

The Team also claims that there's an annual cycle. Looking at our data, we can see that for both pageviews and revenue, there seem to be a peak at the end of each year. However, since our data only spans 2 years and 10 months, it is also very likely that the 'pattern' is mere coincidence. Checking ACP and PACF plots is also not helpful, as the annual cycle is lag 365.

Since it is very hard to determine if there's an annual cycle, we can attempt to remove it and see if the residual still looks stationary. After recombining the trend and residual from the weekly cycle decomposition, I used 'stl()' again on the data after taking the log. The decomposition results are shown below:


Log Total Revenue, STL on Log: annual cycle

Log Total Revenue, STL on Log: trend

Log Total Revenue, STL on Log: random

Figure 6.11: STL Decompostion of Log Total Revenue, 2nd Round

Then we undo the log transformation and check the four components together:

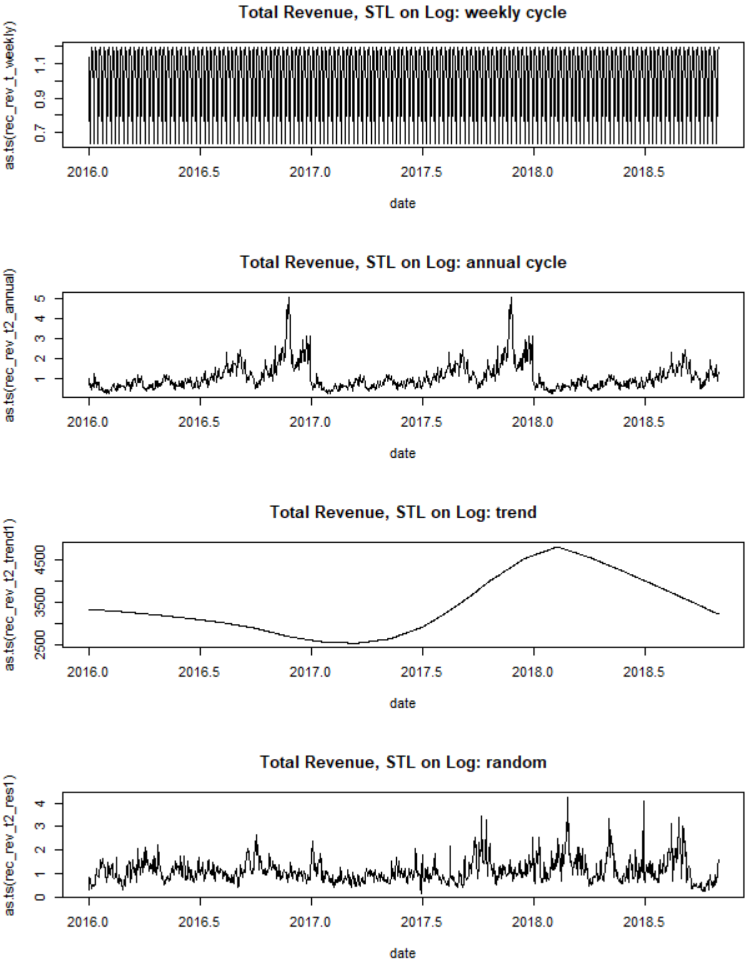


Figure 6.12: STL Decompostion of Total Revenue, 2nd Round

It is interesting to note that, after removing an annual cycle, the residual actually looks less stationary. The variance seems to increase in 2018, and the data spreads on the both sides of mean unevenly. This is probably caused by overfitting, which is highly likely since there are only 2 cycles. Since the residual doesn't see improvement in terms of being stationary, we will not attempt to remove the annual cycle in further studies.

6.2 ARIMA Model

After separating the seasonality and trend by first taking the log and then applying the 'stl()' function, we can see that the residual generally satisfies the univariate assumption. Thus we can then use the autoregressive integrated moving average (ARIMA) model to further analyze the residual.

The ARIMA model is a class of forecasting model that can be applied to stationary time series data, which has constant mean and variance. It is a generalization of the autoregressive moving average (ARMA) model, and utilizes both lags of the dependent variable and the lags of forecast errors as predictors. The equation for ARIMA consists of two main components: the autoregressive component (AR), in which regressions on its own lagged values are applied to the variable; and the moving average model (MA), which is the linear combination of both contemporary and past errors. In the particular R package that we will be using in our analysis, the equation is represented as:[8]

$$X_t = a_1X_{t-1} + ... + a_pX_{t-p} + e_t + b_1e_{t-1} + ... + b_qe_{t-q}$$

We will be denoting the non-seasonal ARIMA models in the form of ARIMA(p,d,q), where p is the order of the autoregressive model in the AR part, d is the degree of differencing, and q is the order of the MA part.

To pick the optimal parameters for the ARIMA model, we will be checking the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) to assess the quality of each model. In order to recognize over-fitting, these two criterion both include penalty terms for the number of parameters in the model. AIC is used to pick frequencies in the 'ar()' function mentioned before; BIC is closely related to AIC, and put more penalty on the number of parameters.[9] In this analysis, we will be choosing the model with the lowest AIC value, and the model with the lowest BIC value, and compare the results.

We have already obtained the residual for total revenue after applying the 'stl()' function and removing the trend and seasonality, so we can test the ARIMA models with different parameters on the residual. One of the goals of this analysis is build the model based on historical data from 01/01/2016 to 08/16/2018, and estimate what the revenue could have been if AMP wasn't paused for 2 months. Thus we will use the first 958 entries in the reconstructed residual as the training data, and attempt to forecast the revenue in August, September and October 2018 based on historical data. We will test 49 models in the form of ARIMA(p,0,q), where $p, q < 8$, and record their AIC and BIC values. Note that we use d=0 here, because the residual has already been detrended and should not need differencing.

The results are shown in the table below:

p/q	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	374.7251	217.0967	211.9529	207.4155	195.2627	188.4762	195.3084
2	196.6151	197.8494	195.6214	197.6214	195.3124	197.2498	196.5653
3	198.1373	193.6841	195.3225	196.1797	197.1270	199.1098	196.0123
4	196.1916	195.4317	183.6389	186.0018	187.8083	200.9007	196.4771
5	198.1659	196.5115	185.9812	192.4668	189.3137	190.8592	191.8193
6	199.5903	198.0556	199.3094	189.5928	178.6963	181.0362	182.7255
7	200.7506	199.6336	201.0430	190.9811	180.2813	182.8504	184.7136

Table 6.1: AIC Values for ARIMA Models on Total Revenue

p/q	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	-2.476957	-2.631577	-2.638491	-2.644845	-2.658624	-2.667626	-2.662469
2	-2.644756	-2.645493	-2.649609	-2.649609	-2.653851	-2.653900	-2.656483
3	-2.638508	-2.644790	-2.645141	-2.646258	-2.647290	-2.647307	-2.652329
4	-2.635634	-2.638328	-2.652178	-2.651724	-2.651890	-2.640796	-2.647642
5	-2.628951	-2.632517	-2.645054	-2.640662	-2.649428	-2.649855	-2.647279
6	-2.622806	-2.626254	-2.626972	-2.642373	-2.655571	-2.655369	-2.655690
7	-2.616923	-2.619948	-2.620521	-2.636316	-2.649160	-2.648686	-2.648735

Table 6.2: BIC Values for ARIMA Models on Total Revenue

As we can see here, the lowest AIC value is 183.6389 from ARIMA(6,0,5), and the lowest BIC value is -2.6676 from ARIMA (1,0,6). We will now use these two models to forecast the rest of 2018 data, using the 'sarima.for()' function from the same package. For ARIMA(6,0,5), the predicted residual and reconstructed total revenue using actual trends are shown below.

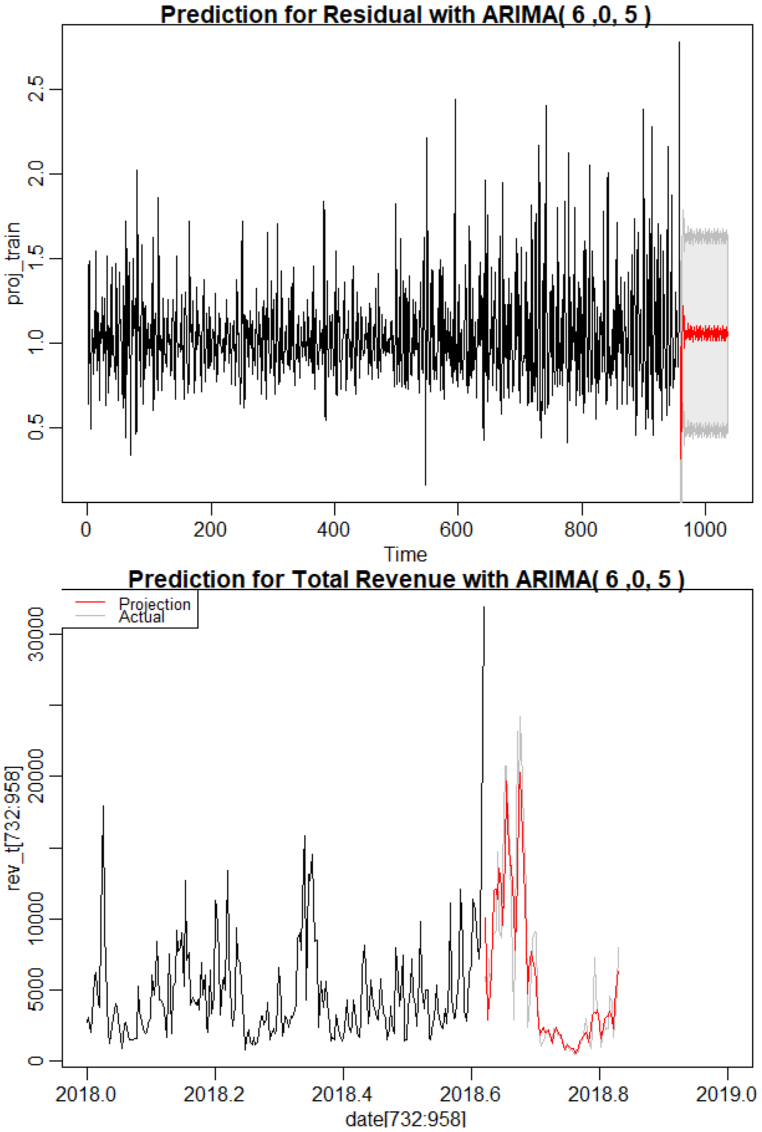


Figure 6.13: Residual and Total Revenue Prediction Using ARIMA(6,0,5)

In the top plot, the grey area represents the 95 percent confidence interval. We can see that the actual prediction for residual quickly decreases in variance, and becomes relatively stable after about 10 entries. After that, the prediction fluctuates in a very small range between 1 and 1.5. The 95 percent confidence interval, however, is very large, and covers the area below and above the mean that most of the historical data fall into. The reconstructed total revenue is very close to the actual revenue, with only some minor differences.

Now we can also look at the results for ARIMA(1,0,6):

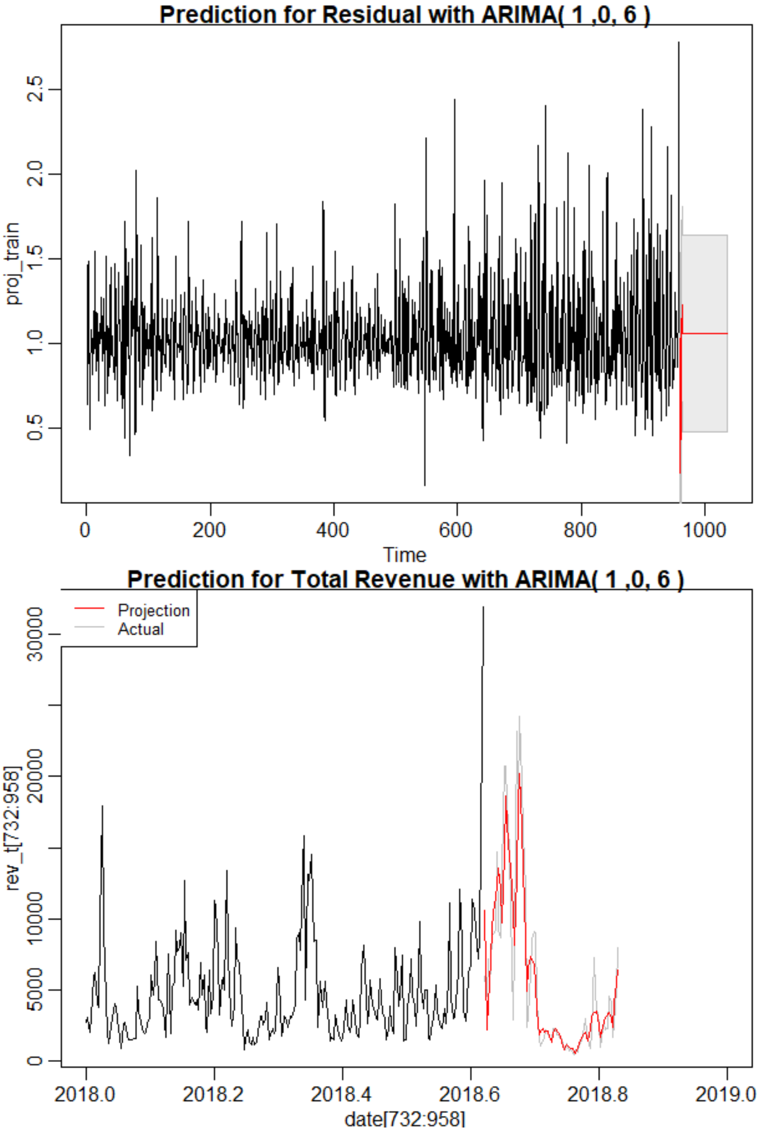


Figure 6.14: Residual and Total Revenue Prediction Using ARIMA(1,0,6)

The predicted residual values generated from ARIMA(1,0,6) also shows significant drop in variance, and, after becoming relatively stable, fluctuates in such a small range that it looks like a straight line at the mean value. The 95 percent confidence interval is also very large. The revenue prediction looks similar to the result from ARIMA(6,0,5), with some minor differences. It is clear that the difference between these two model does not impact the projected revenue in a very significant way. Since the prediction with ARIMA(1,0,6) is basically a constant after the first 10 entries, we will use the results from ARIMA(6,0,5) for the prediction model.

One particular prediction that The Team was very interested in is ‘what the revenues could have been for August, September, and October 2018, if AMP pages was not paused’. Since for Model I we do not distinguish the AMP and WEB pages, it is difficult to separate the effect of the change in web page formats. As a result, we will need three assumptions to produce an estimation:

1. An annual cycle does exists;
2. The first 7 months of 2018 represents the general trend throughout the entire year;
3. The decision to pause AMP pages in 08/16/2018 is the only significant factor that impacts the trend of revenue; if this did not happen, there would have being no change in the trend throughout the rest of the year;

Instead of using the entire dataset, we will only use the data before 08/16/2018 for this prediction model, which has 958 entries in total; we will call this dataset ‘the truncated dataset’.

Using the same method discussed previously, we can remove the weekly cycle and annual cycle from the truncated dataset using the ‘stl()’ function:

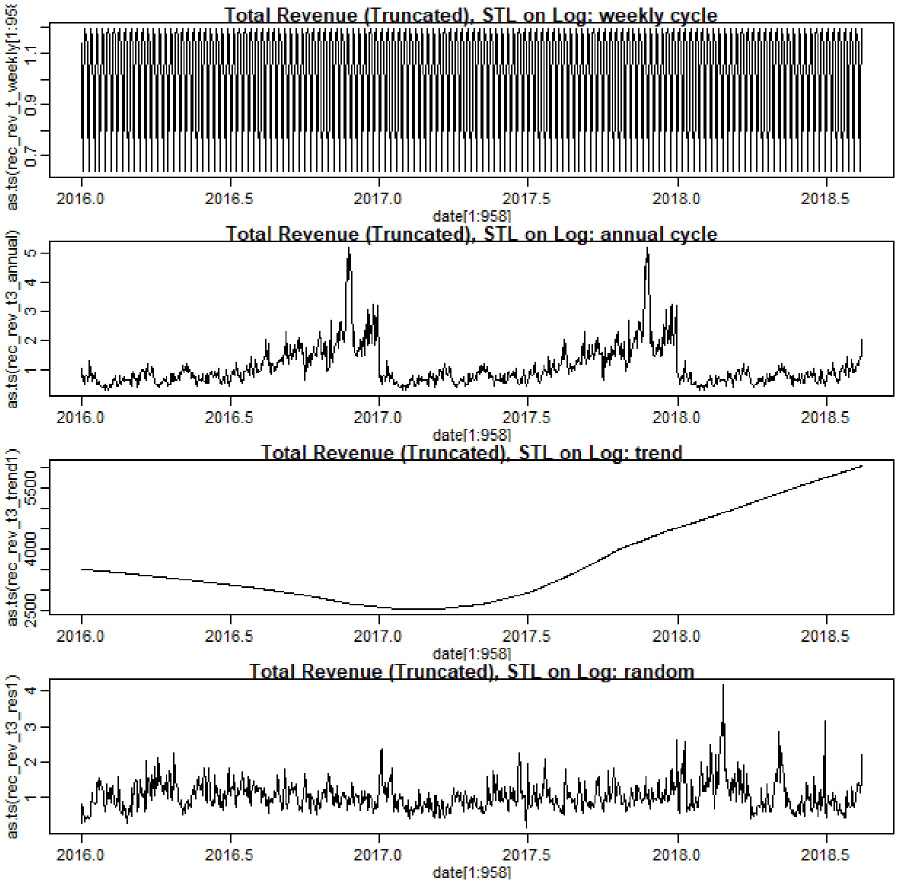


Figure 6.15: STL Decomposition on Truncated Total Revenue

We can see that, comparing to the decomposition on the complete data, the weekly cycle is almost the same, but there are some differences between the annual cycles and trends. Using the truncated data, the overall trend in 2018 is steadily increasing, at almost a linear rate. Thus for the prediction, we can apply a simple linear regression on the trend data, and predict the trend in August, September, and October 2018 if there wasn't a significant shift in strategy. The results from ARIMA(6,0,5) will be used to predict the residual. The predictions for the four components are shown below in red:

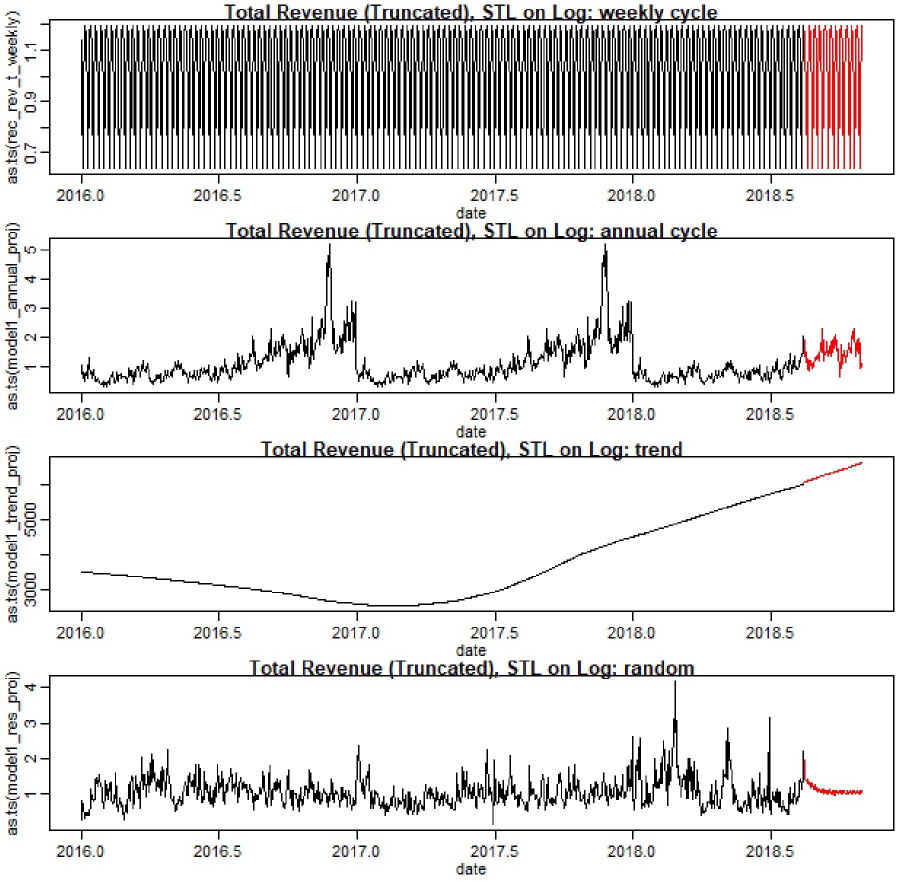


Figure 6.16: STL Decomposition on Truncated Total Revenue with Prediction

Combining them together, we finally have a prediction from Model I for the revenues, if AMP pages was not paused:

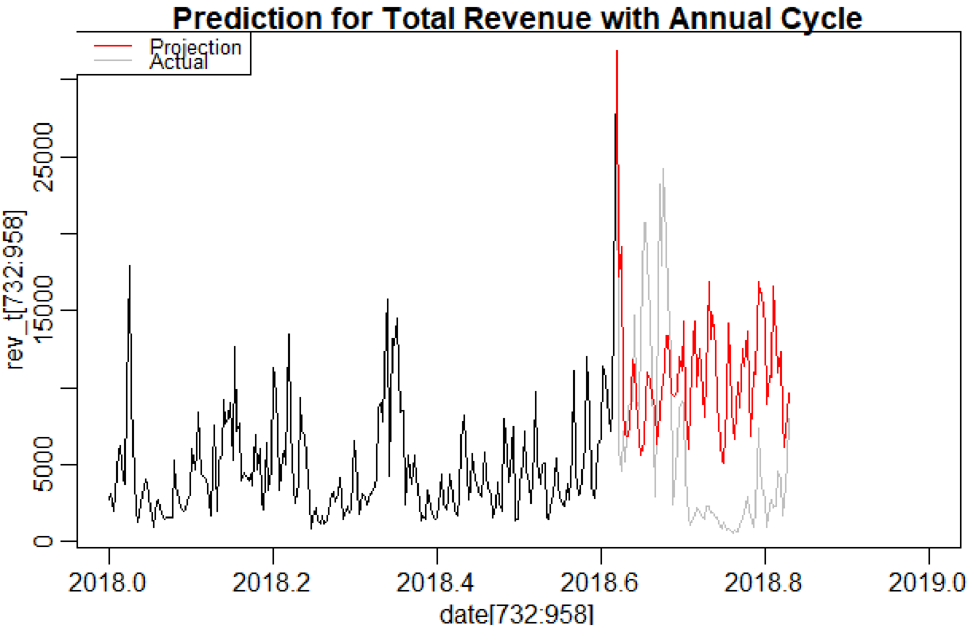


Figure 6.17: Prediction Based on Truncated Total Revenue

The prediction is generally higher than the actual revenues, and follows the upward trend in the first half of 2018. However, immediately after AMP pages were paused, the actual revenues are significantly higher than the prediction; this may be caused by the revenue-boosting one-time event that was mentioned by The Team, or maybe a side-effect of pausing the AMP pages.

CHAPTER 7

Model II: Separate Models for WEB and AMP Revenue Data

For the second method, we will build two separate models based on the WEB and AMP revenue data respectively, and add the results together in the end. Most of the methodology for this model has been discussed in Model I, so we will primarily focus on the results.

7.1 Detrending

First, we will conduct spectral analysis on both data to determine what the dominating cycles are. For the WEB revenue, the results are shown below:

The periodograms for WEB also show a peak at around 0.14 frequency, similar to those for total revenue. Thus we can set a period of 7, and remove the weekly cycle by applying the ‘stl()’ function on the log data. The reconstructed data is shown below:

We can see that the reconstructed trend shows a strong peak when AMP was turned off in 08/16/2018. This is likely caused by the fact that, since there were no AMP pages for new articles, all the readers have to use the WEB pages. The residual seems generally uniform, though the variance may have increases slightly at the end of the plot.

For the AMP data, we will only use the data when AMP pages were available, which ranges from 07/12/2017 to 08/16/2018. The result of spectral analysis on the remaining AMP revenue data is shown below:

In the raw periodogram, the peak at the frequency for the weekly cycle is very obvious. However, after applying the kernel smoothing, that particular peak is actually removed. This

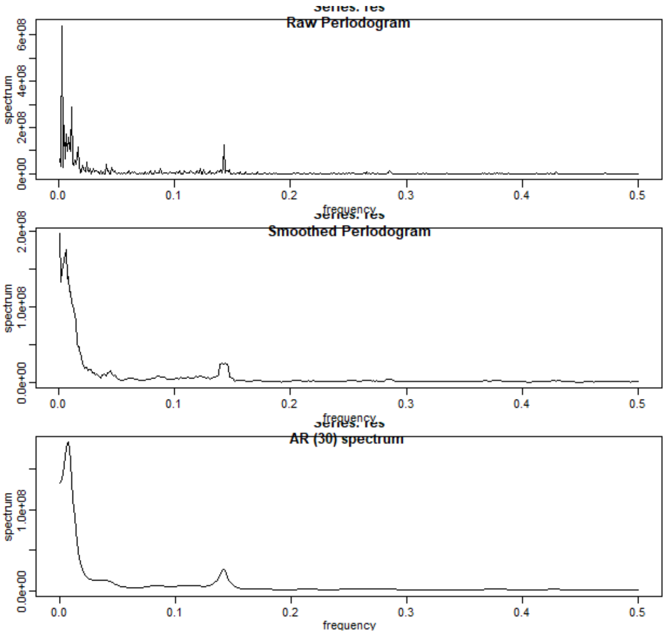


Figure 7.1: Raw Periodogram, Smoothed Periodogram, and AR Plot for WEB Revenue

is likely due to the fewer dataset available for AMP revenue, and is not an indication that the weekly cycle does not exist. We will use the same method to remove the weekly cycle:

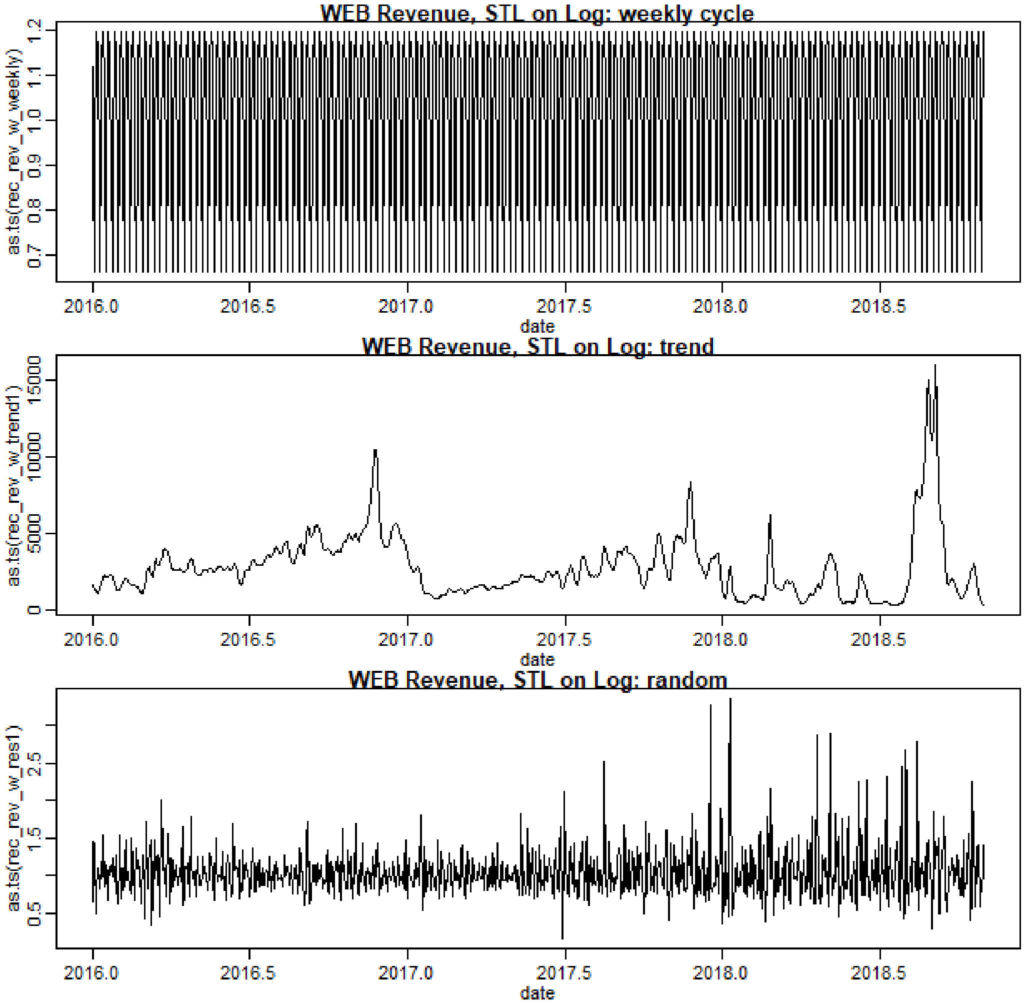


Figure 7.2: STL Decomposition on WEB Revenue, After Log Transformation

We can see here that, with the weekly trend removed, the trend is relatively stable, and the residual looks stationary. We can also see that the one-time event in August did not create a peak in the trend; on the contrary, the revenue actually went down right before AMP pages were paused.

7.2 ARIMA Model for WEB Revenue

After seperating the weekly cycles in both data, we can now apply the ARIMA model to analyze the residuals, using the same approach as in Model I. After testing various ARIMA

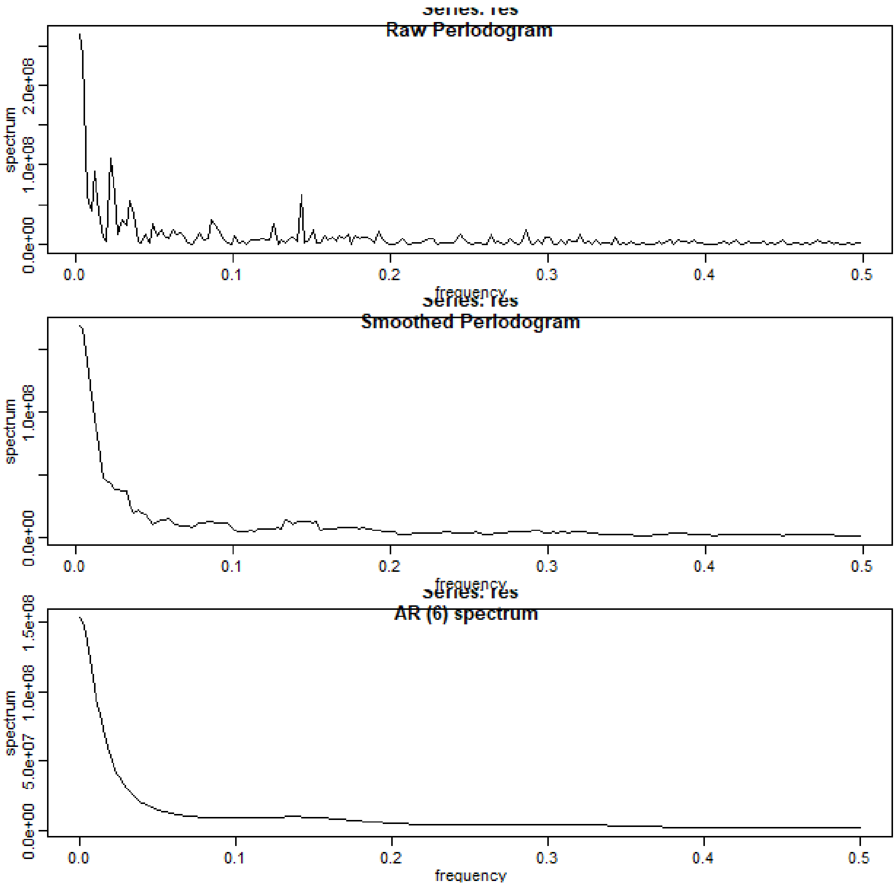


Figure 7.3: Raw Periodogram, Smoothed Periodogram, and AR Plot for AMP Revenue

model parameters, ARIMA(5,0,7) has the lowest AIC value of 415.99, and ARIMA(2,0,7) has the lowest BIC value of -2.44. We can start by checking the prediction using ARIMA(5,0,7):

The result is very similar to the one we saw in Model I. The range in which the predicted value fluctuates quickly shrinks, and the line stabilizes in a small range above and below the mean value. The reconstructed prediction revenues are very close to the actual revenue, showing that the residual component has a relatively small effect on the overall values. Then we can also check the results for ARIMA(2,0,7):

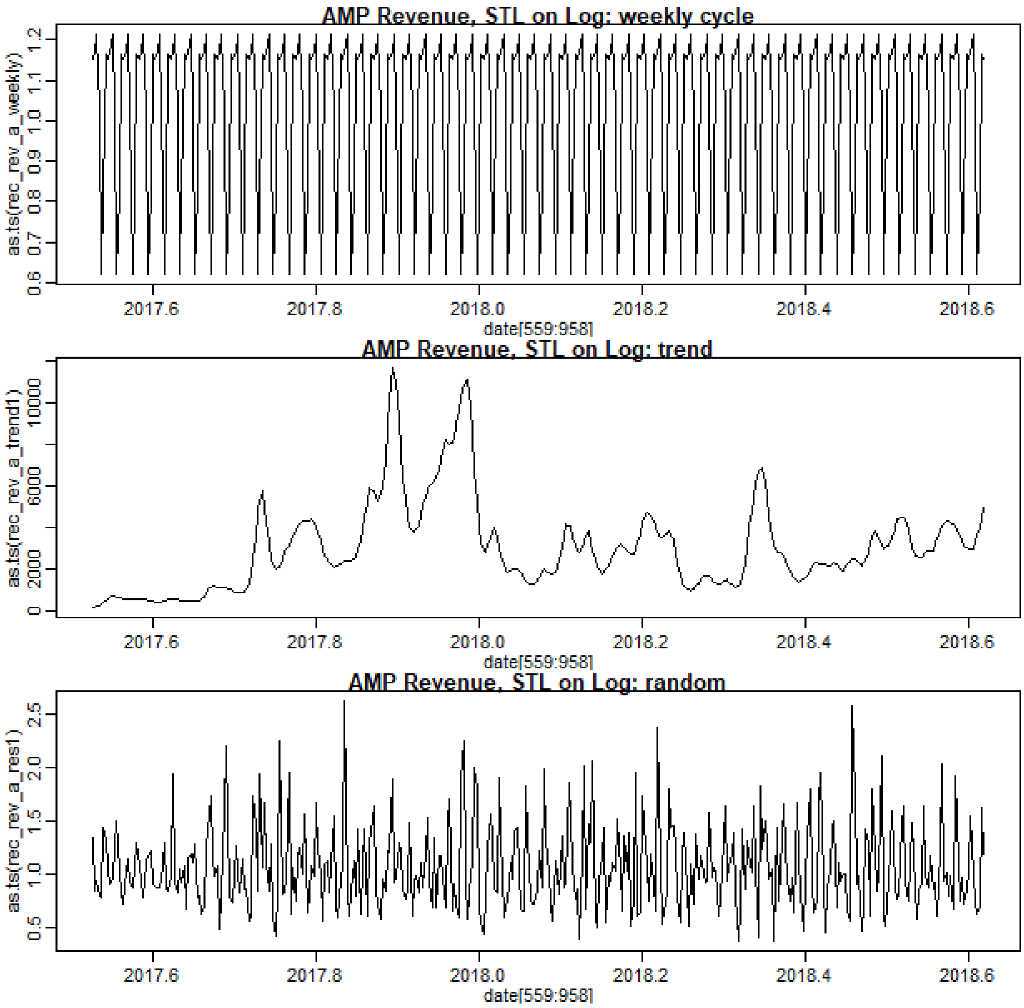


Figure 7.4: STL Decomposition on AMP Revenue, After Log Transformation

Using ARIMA(2,0,7) model, the results are again very similar. Considering that the ARIMA(2,0,7) model has fewer dimensions and a lower BIC value, we will use its prediction for further analysis.

We can then use this model to predict what the WEB revenue would be if the AMP pages wasn't paused. We still would need the three assumptions in mentioned Model I to make the prediction. If there is an annual cycle, we can use the truncated data for detrending, and then reconstruct the weekly cycle, annual cycle, and the trend:

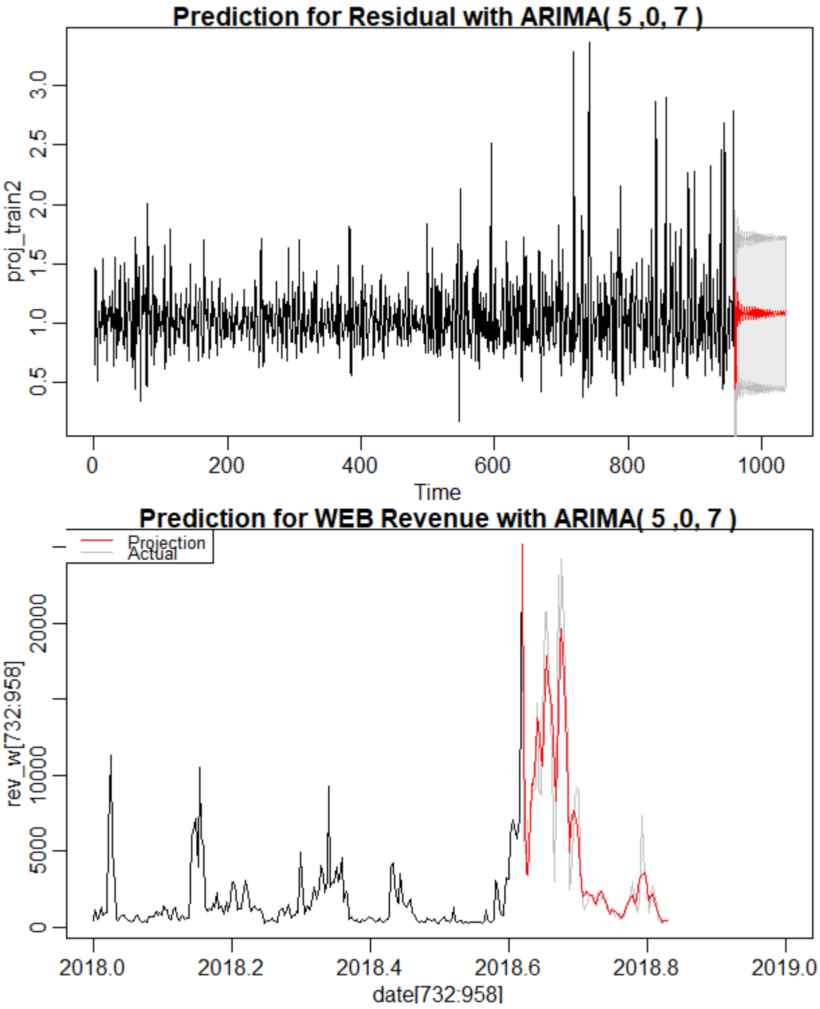


Figure 7.5: Residual and WEB Revenue Prediction Using ARIMA(5,0,7)

As the trend component above shows, the truncated WEB revenue is steadily decreasing in a approximately linear fashion. Thus we can gain use the 'lm()' function to fit a linear regression on the trend, the use ARIMA(2,0,7) to predict the residuals. The predictions for the four components are shown below in red:

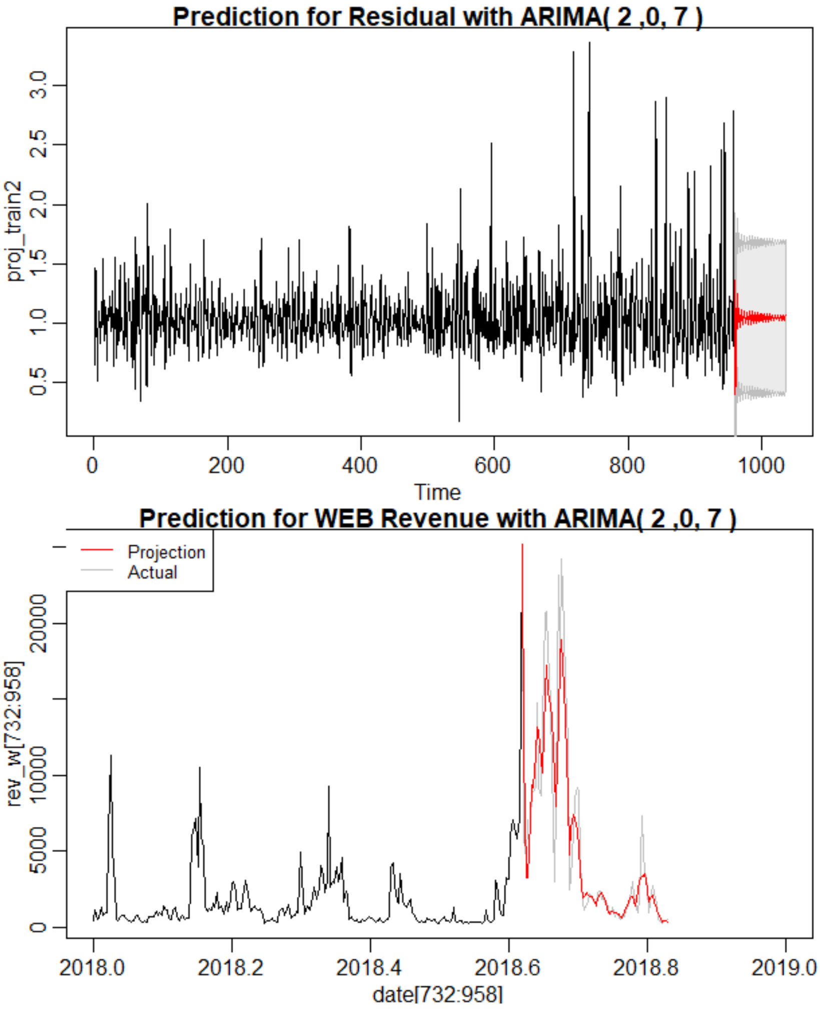


Figure 7.6: Residual and WEB Revenue Prediction Using ARIMA(2,0,7)

Combining them, we have the predicted WEB revenue based on Model II:

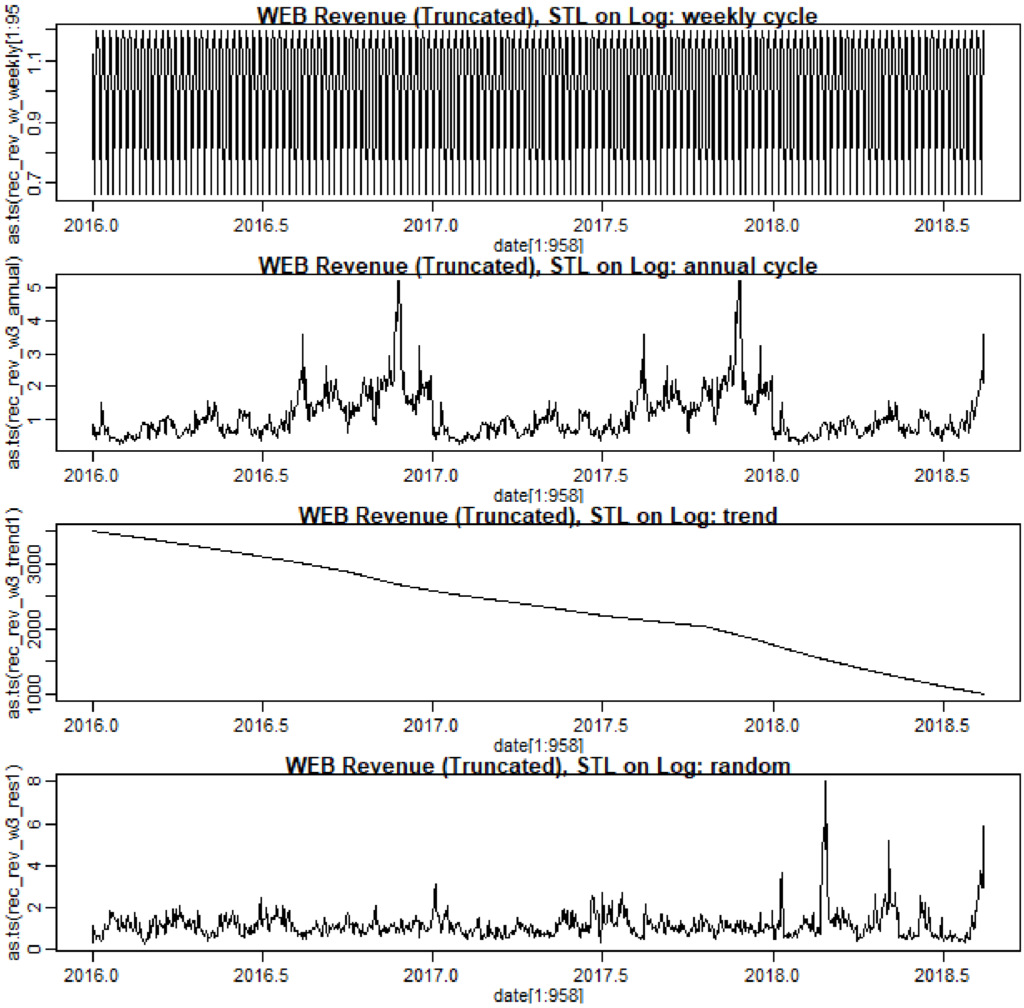


Figure 7.7: STL Decomposition on Truncated WEB Revenue

It seems that without the pause on AMP pages, the revenue for WEB pages would continue to stay relatively low, which has been the overall trend of the WEB pages for the recent years.

7.3 ARIMA Model for AMP Revenue

Now we can move onto building the ARIMA model for AMP revenue data. First, we can analyze the residual from removing the weekly cycle, and test parameters for the ARIMA models. ARIMA(5,0,4) has the lowest AIC value at 328.32, while ARIMA(1,0,7) has the

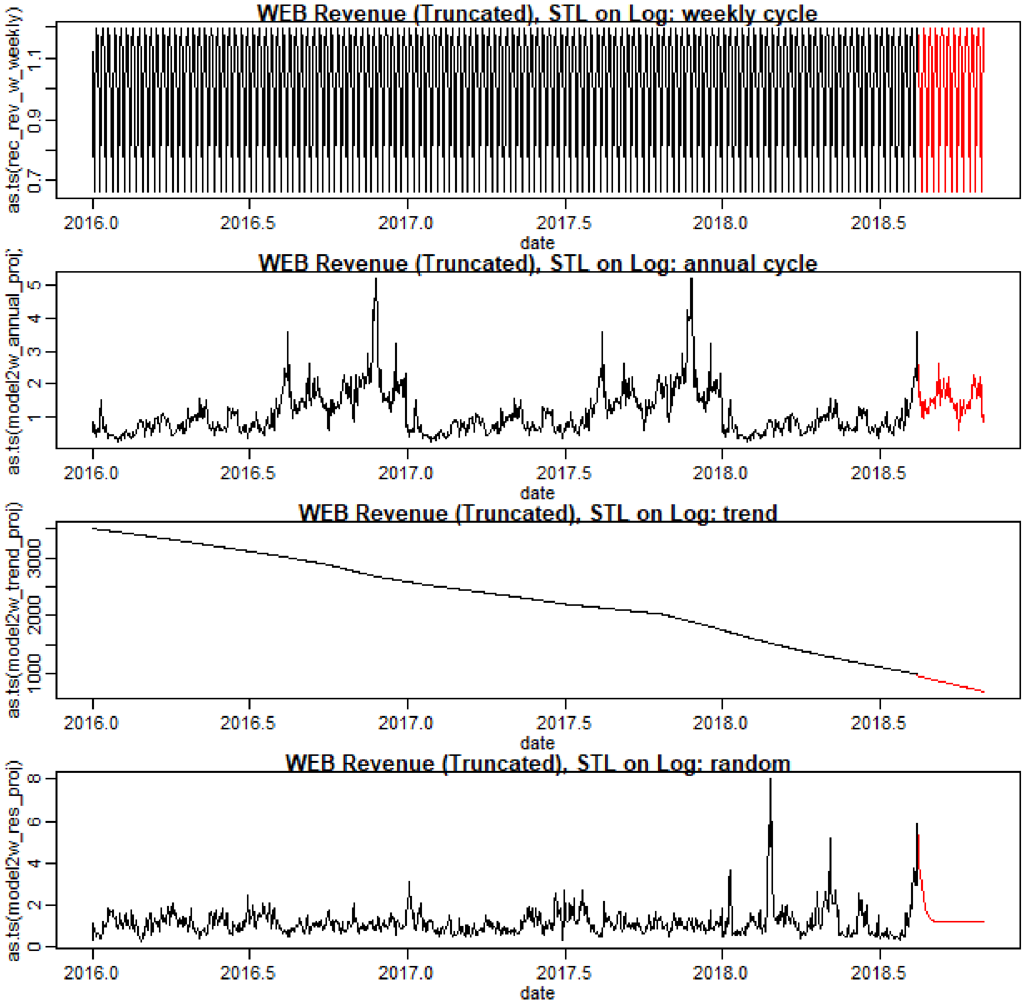


Figure 7.8: STL Decomposition on Truncated WEB Revenue with Prediction

lowest BIC value at -2.04. We can check the predictions generated from these two models:

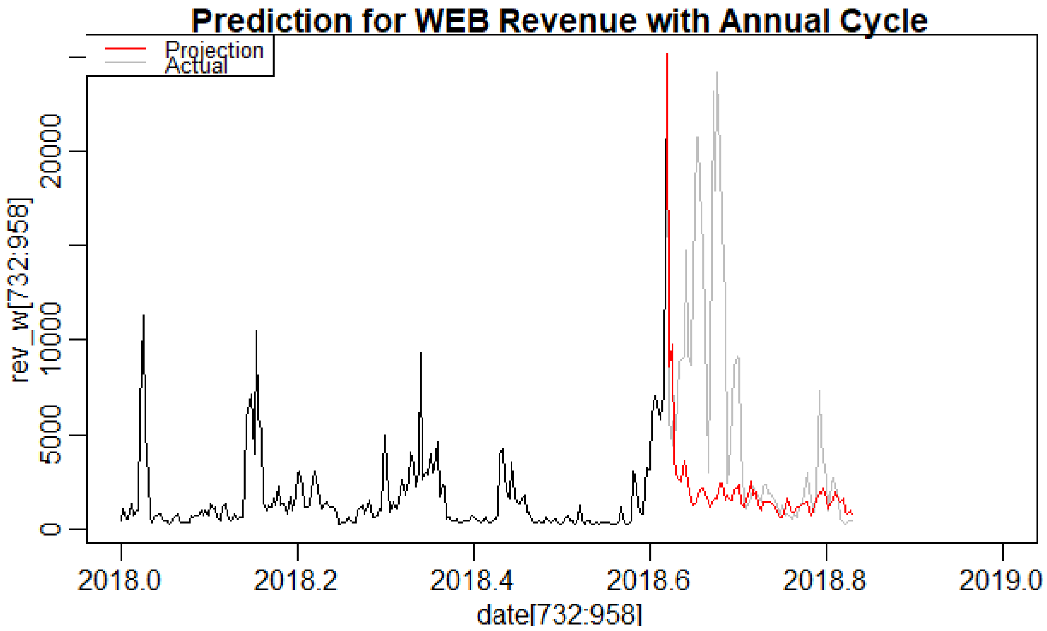


Figure 7.9: Prediction of WEB Revenue Based on Truncated Data

We can see that the predictions from ARIMA(5,0,4) follows a wave pattern with clear cycles, while ARIMA(1,0,7) becomes approximately a straight line on the mean value after some fluctuations. We will use the results from ARIMA(5,0,4) for the purpose of this analysis, since it involves more variation in the predicted values.

To finish this model, we would need to predict the AMP revenue values if AMP pages was not paused. This part is especially difficult, mainly due to the small dataset available. AMP pages was active between 07/12/2017 and 08/16/2018, spanning only about a year. This means that the method we have used for WEB and total revenue prediction does not work here, since we do not have a complete annual cycle even if we assume that it exists. The 'stl()' function in R is designed for at least 2 cycles, so it cannot be applied here.

After understanding the methodology behind the 'stl()' and 'decompose()' function, I generated an annual cycle by using a moving average on the trend component, and then applied a coefficient to each date based on historical data. This is basically using a simplified version of the 'stl()' function, since we only have one cycle and cannot properly assign weights. We apply the annual cycle to a linear regression on the 2018 AMP revenue data, and obtained

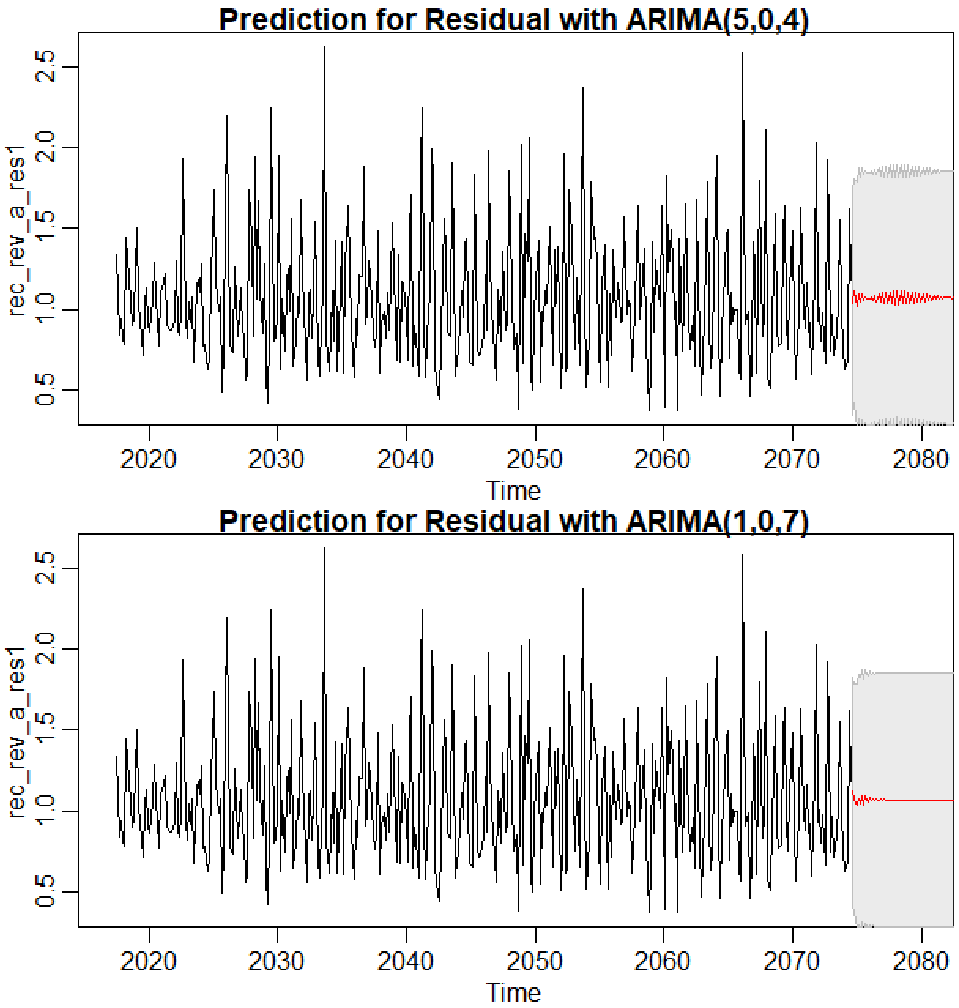


Figure 7.10: Residual Prediction Using ARIMA(5,0,4 (top) and ARIMA(1,0,7)(bottom)

the prediction for the trend. The predictions for the three components are shown below in red:

Combining them, we obtain the prediction for AMP revenue:

The predicted AMP revenue values starts relatively low, fluctuating around $2000 per day. Then it grow dramatically around September and October, even surpassing $10,000 per day at some point. This is based on the assumption that an annual cycle exists, and, following the trend in 2017, the revenue will reach a peak in the second half of the year.

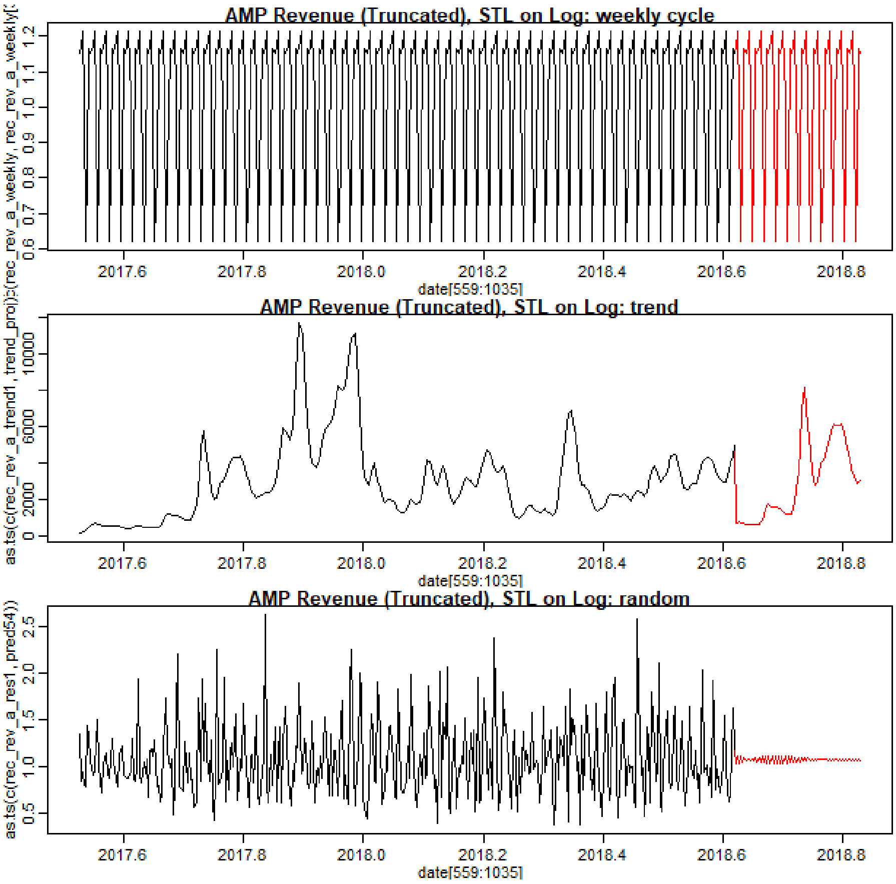


Figure 7.11: STL Decomposition on Truncated AMP Revenue with Prediction

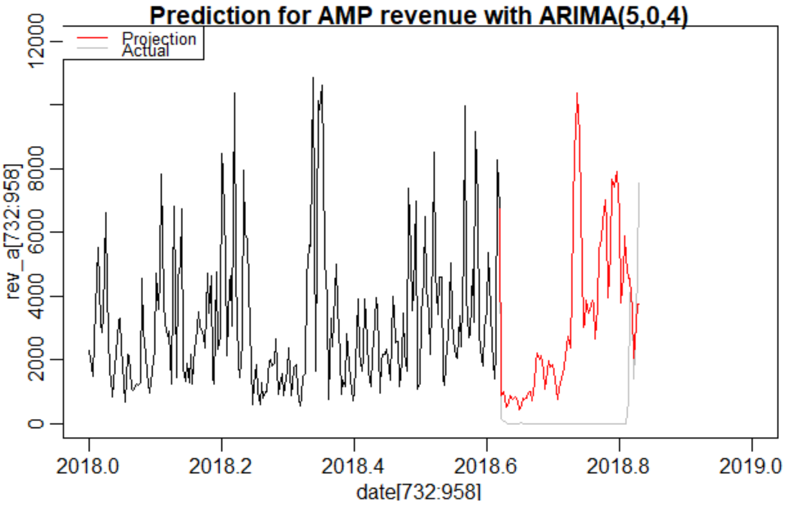


Figure 7.12: Prediction of AMP Revenue Based on Truncated Data

CHAPTER 8

Model Comparison

We can put the final predictions from the two methods into one plot to compare:

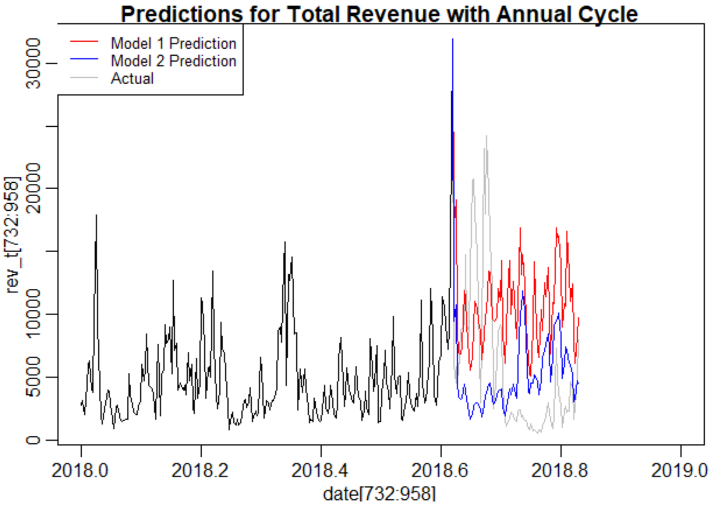


Figure 8.1: Prediction for Total Revenue from Model I and Model II

It is clear that Model II yields much lower predicted revenues comparing to that of Model I. From a financial perspective, this means that The Team would admit to a loss in The Website's revenue if they choose to adopt Model I. In Model I, we only considered the total revenue regardless of source, and, as a result, chose to not involve all data in our model building. After removing the weekly and annual cycle, we found a linearly increasing trend, and used that trend to produce the higher prediction values. In Model II, we learned that when separated, Revenue for WEB pages is dropping, and revenue for AMP pages is growing relatively slowly. This caused the predictions generated from the two models to be lower.

It is hard to access the quality of these two models directly, since one of the goal of this analysis is to generate prediction for a hypothetical scenario, which do not have testing data or accuracy. In terms of model simplicity and fitting the training data, Model I performs better, as it has a complete annual cycle, a generally linear trend, and univariate residual. The WEB revenue model in Model II also fits well, but the AMP revenue model requires more assumptions and also has less stationary residual, probably due to the smaller size of the dataset.

We should also make into account the assumptions required for each model. In order to predict values for a hypothetical scenario, we need assumptions in both methods. The most important one being that the annual cycle exists, which cannot be adequately proven using the current data we have. When calculating the predicted revenue for AMP pages, we made an even stronger assumption, by building a simplified annual cycle based on data from only the second half of 2017 and the first half of 2018. Although The Team claimed that an annual cycle does exist, we cannot draw that conclusion based on two and a half years of data. Model I also involves fewer variables, so it is likely that the prediction form Model II is more biased.

Another difference in assumptions is that in Model I, we are evaluating the total revenue of the website, treating the viewers of The Website as a whole. In Model II, however, the two models are built separately without interaction, which means that we assumed that the AMP and WEB pages are viewed by two separate groups of audience. In reality, the pageview numbers of the two versions of The Website are close connected, and the viewership is treated as one group across platforms and versions of webpages. It is also the general trend in the industry that more and more viewers will browse the AMP version of a webpage instead of the normal WEB version, mainly through the success of Google and its line of products. We cannot separate the effect of that trend from our data when analyzing in Model II.

To summarize, Model I needs fewer assumptions, uses fewer variables, and is simpler in structure and calculation. Model II, on the other hand, offers more freedom in manipulating variables, and offer much more insight into the interactions between two versions of the web pages.

CHAPTER 9

Analyzing the Effect of Pausing AMP Pages

The first goal of this analysis, which is building a model for predicting total revenue, has been accomplished. Now we can attempt to analyze the effect of pausing AMP pages, starting in 06/16/2018. From the previous trend plots for total revenue, it is clear that the decline right after AMP pages were turned off is not likely to be caused by the seasonality. In the decomposition results using the ‘stl()’ function, we can see that there is a change in direction in the middle of 2018, where the trend sharps goes downward. When we use the truncated data to predict the effects, we see that the trend components does not have this peak, and continue to grow throughout the first half of 2018. This shows that the revenue data after AMP pages were paused have a negative impact on the overall trend of the revenue data, after we have removed the weekly cycle.

To understand the cause behind the drop in revenue, we first need to analyze the trend in pageviews for The Website. Using the similar method as before, we can separate the seasonality and trend, again assuming that an annual cycle exist:

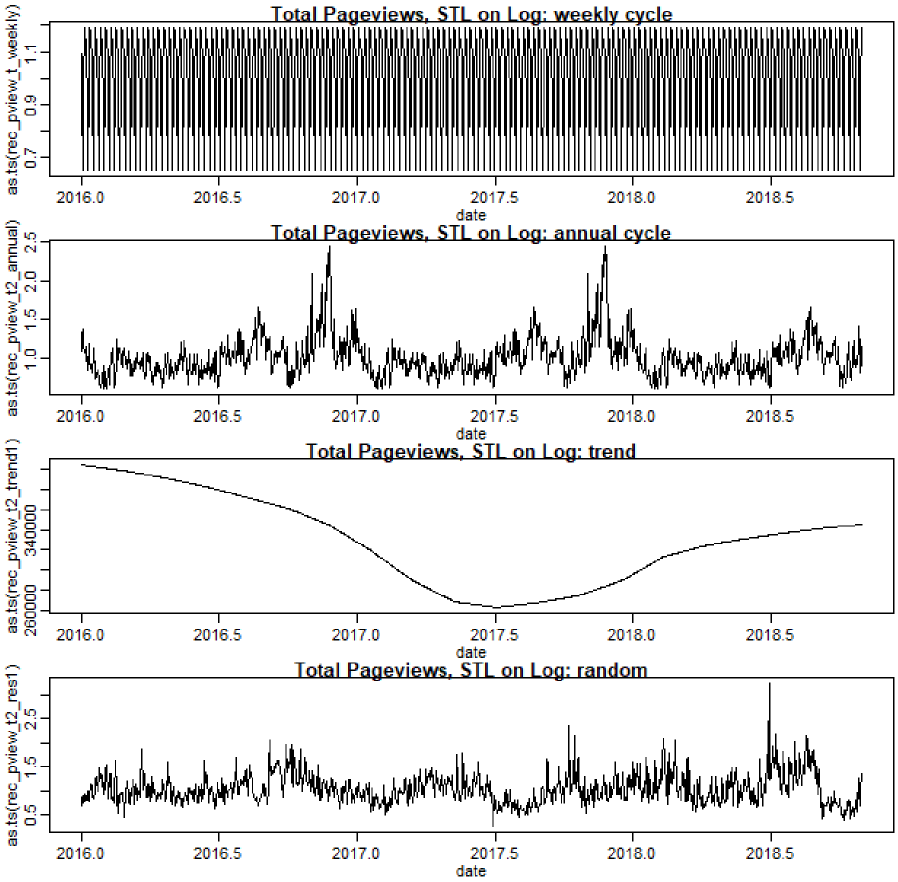


Figure 9.1: STL Decomposition of Total Pageviews, 2nd Round

The seasonality for pageviews has a similar pattern to that of the revenue, with a strong weekly cycle, and relatively high values at the end of each year. The trend plot shows an upward trend starting from the second half of 2017, though the growth rate seems to be slowing down in 2018. We will examine the residual plot more closely:

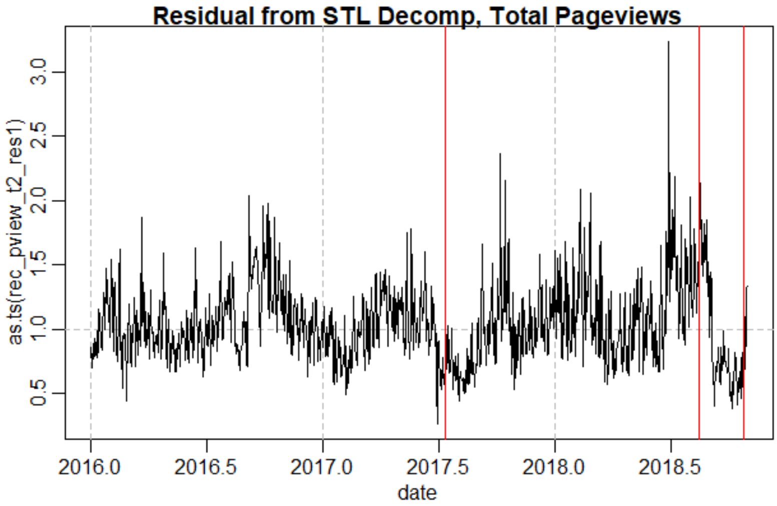


Figure 9.2: Residual from Decomposition of Total Pageviews

In the residual plot, the value generally fluctuates around the mean value 1; but right after AMP pages were paused, the residual drops sharply below 1, and stayed under until the AMP pages was turned back on again. This shows that there is sudden decrease in total pageviews that cannot be explained by the seasonality or the general trend. This is highly likely to be caused by the pause on AMP pages, which are much more optimized by Google searches, and attracts more views than their WEB counterparts.

But the decrease in total pageviews had been expected by The Team before they made the decision to pause AMP pages. They believed that, since WEB pages have higher RPMs, the decrease in total pageviews can be offset by the increase in average revenue per page when calculating total revenues. So we will now analyze the behavior of RPMs in this dataset. First, we would like to separate the seasonality and trend from the data.

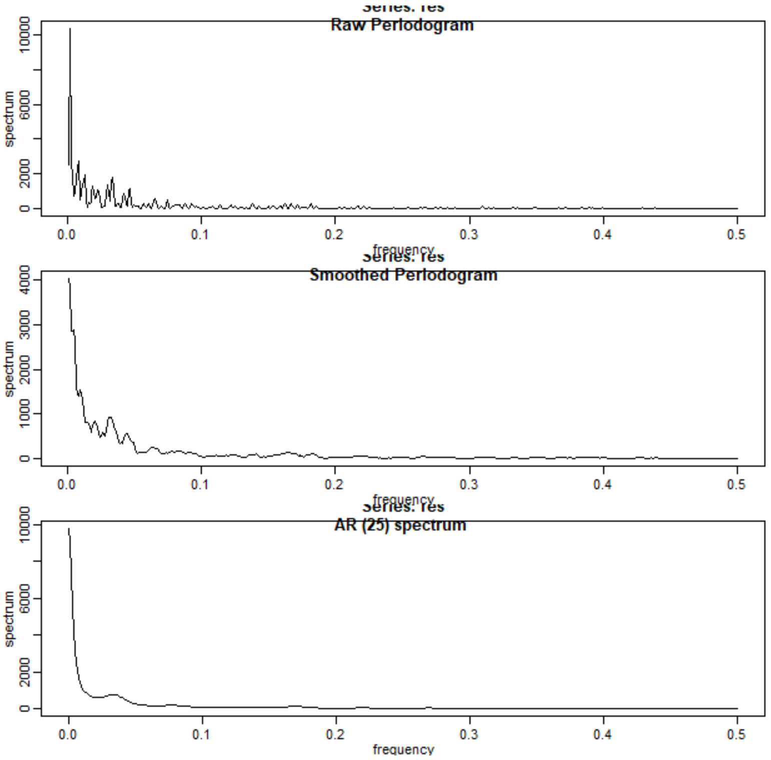


Figure 9.3: Raw Periodogram, Smoothed Periodogram, and AR Plot for the Residual of WEB RPM

By checking the periodogram of the WEB RPM data, we cannot find any strong cycle. Unlike the other data, the RPM does not seem to have a strong weekly cycle. We can try to remove a 7-day cycle anyway, but the resulting residual is very unstationary:

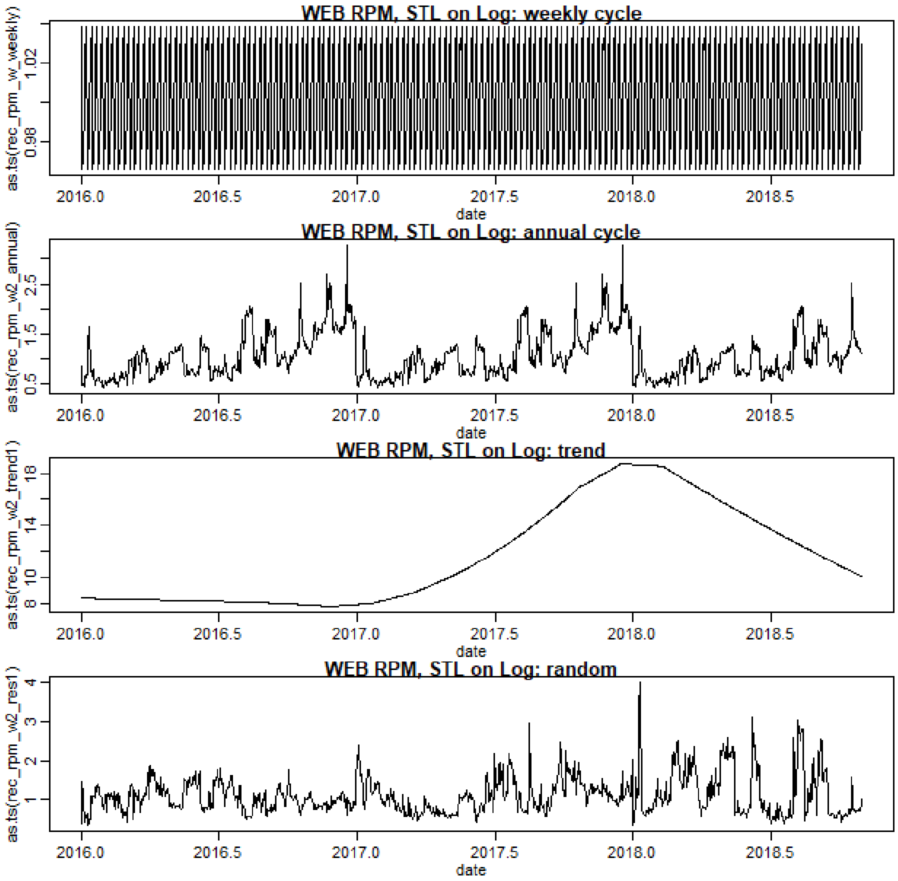


Figure 9.4: STL Decomposition on WEB RPM, After Log Transformation

As a result, we cannot use the time series analysis methods we have used for other parameters to predict the value for RPM. A normal regression would also be ineffective, since the data has a changing variance and many extreme values. Thus RPM is not a good predictor, and should not be used to build prediction models.

From the perspective of The Team, RPM is considered to be an inherent property of the particular web page, closely related to the design and advertisement placement of that web page, and should not be related to time. The WEB version, for example, was believed to have a higher average RPM because the web pages are designed better, and provides more slots for advertisement placements.

However, the idea that 'the RPM is an inherent property of the web page design' cannot explain the sudden drop in WEB RPM when AMP pages were suspended on 08/16/2018.

Since nothing about the WEB pages' layout and design were changed at all, the RPM for WEB pages should not be affected by the pause on AMP pages. The drop is also unlikely to be caused by seasonality, since, as previously mentioned, there are no strong seasonality or trend involved in the RPM data.

To explain this sudden drop in WEB RPM, we have developed an alternative theory: that RPMs are not directly linked to the web page, but linked to the specific group of visitors. When both AMP and WEB versions of a web page is available, you can only get access to the WEB version through navigating directly through the homepage of the Website. All the Google search results, links from apps like Apple News, and external advertisement will take you to the AMP version of the web page instead. So it is reasonable to argue that the visitors of WEB pages either have bookmarked The Website's homepage and checked it for news, or found The Website homepage through Google search and then browse through The Website until he/she landed on that particular page. They have shown relatively strong interest in the subject matter of The Website, having bookmarked or browsed through The Website. As a result, the visitors to the WEB pages are more likely to click through the advertisements and generate revenue for The Website, resulting in a higher RPM for that page. When the AMP pages were suspended, all visitors have to go through the WEB pages, and the group of visitors with strong interest in the subject matter was joined by all the rest of the average people who searched for the web page on Google, lowering the overall RPM of the WEB page.

Because of the current technology limitations and the anonymity of the Internet, we could not discretely prove the alternative theory about RPM to be correct; but it does offer a much reasonable explanation for the sudden drop in WEB RPM when the AMP pages were paused, and also aligns better with the idea of audience targeting, which is very popular in the field of digital marketing. The Team has accepted my theory, and is currently updating their tracking system to record the frequency and repetition patterns of individual visitors to the website, in order to distinguish the group of visitors with strong interest, and design their marketing strategy accordingly.

CHAPTER 10

Conclusion

The analysis project is designed to fulfill two main goals. The first is to establish a prediction model for the daily total revenue of The Website, and produce predictions based on a scenario. After some data cleaning and transformation, we developed two different models for the prediction. Model I uses the historical data of total revenues as the predictor, and contains a weekly cycle, an annual cycle, a piece-wise linear trend, and a residual component that can be predicted with ARIMA(6,0,5) model. They are separated through first taking the log and then using LOESS, and the four components compound multiplicatively to provide a prediction. Model II contains two separate models that predicts the revenue for WEB and AMP pages respectively. Each model is designed similar to Model I, and we can add up the results from the two models to predict the total revenue. The predictions from Model II is more conservative and more biased, but it provides more parameters for scenario testing, and also separate the visitors into two groups.

The second goal of this analysis is to understand the behavior of total revenue when AMP was paused on 08/16/2018. After analyzing the pageview and RPM trends as time series, we argue that the RPM for each web page is directly linked to the group of visitors, rather than the page's design. Pausing AMP pages has driven all the average visitors to the WEB pages, which were previously mostly used by visitors with stronger interest in the subject matter and, as a result, a higher RPM. According to this theory, the decrease in total pageviews can never be offset by the increase in average RPM, since the RPM will decrease too.

From this analysis, we have learned that AMP pages are a vital part of website advertisement strategy, and should not be paused because of its seemingly lower RPM. They bring an increase in the total pageview numbers, and also separate out the high-value audience for the digital marketing team.

For further research, it would be interesting to see if it is possible to design a weighted system based on pageviews that treat the entire potential viewership as one group. For examples, if the AMP pageview is predicted to increase, the weight for WEB revenue prediction would decrease in return, since a large portion of the increased AMP page visitors used to be WEB page visitors. More complicated models could also be developed if we have access to the pageview numbers for each version of the web page, or the number of visitors that browse at least 3 pages before leaving.

Reference

[1] https://blog.amp.dev/2019/05/01/amp-as-your-web-framework. *AMP Development Blog.* May 1, 2019. Retrieved June 2, 2019.

[2] Stoffer, David. *Data sets and scripts to accompany Time Series Analysis and Its Applications: With R Examples (4th ed).* Springer Texts in Statistics, 2017. DOI:10.1007/978-3-319-52452-8.

[3] Bloomfield, P. (1976). *Fourier Analysis of Time Series: An Introduction.* Wiley.

[4] Brockwell, P. J. and Davis, R. A. (1991). *Time Series and Forecasting Methods*, second edition. Springer, New York. Section 11.4.

[5] Akaike, H. (1985). "Prediction and entropy", in Atkinson, A. C.; Fienberg, S. E. (eds.), *A Celebration of Statistics*, Springer, pp.1–24.

[6] M. Kendall and A. Stuart (1983). *The Advanced Theory of Statistics, Vol.3*, Griffin. pp.410–414.

[7] R. B. Cleveland, W. S. Cleveland, J.E. McRae, and I. Terpenning (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Journal of Official Statistics*, 6, pp.3–73.

[8] Durbin, J. and Koopman, S. J. (2001). Time Series Analysis by State Space Methods. *Oxford University Press.*

[9] Bhat, H. S.; Kumar, N (2010). "On the derivation of the Bayesian Information Criterion" (PDF). Retrieved June 2, 2019.