TIME SERIES ANALYSIS OF TWITTER DATA

MATTHEW TIGER

1. Introduction

With the advent of social media, the way in which society communicates has evolved. Popular social media networks document these new forms of communications and as a result, a rich set of data surrounding these interactions emerges. In particular, Twitter remains one of the most popular social media platforms to date. Twitter was first launched in July 2006 as a social networking service designed to allow its users to communicate via short 140-character messages called "tweets". These tweets are sometimes affixed by the sender with a meta-label called a "hashtag", denoted by a string leading with the # symbol, that is meant to categorize the information contained in the message.

In this report, we will analyze data pertaining to a popular television show collected from Twitter's streaming API over the course of three weeks. This analysis will consist of measuring the number of tweets that contain a certain hashtag sent every hour over this timeframe. We will then fit a time series model to these measurements and provide a forecasting model of the next week's projected data in order to to predict the number of tweets that will occur during the show's next airing.

2. Data

In an effort to help researchers glean insight from tweets, Twitter offers a streaming API that streams all tweets that contain a certain string. We leverage this service to gather data surrounding the popular television show *The Walking Dead* and then measure the number of tweets that occur each hour over a time span.

Using the programming language Python and the library `tweepy`, a popular wrapper for the Twitter streaming API, we collected all tweets containing the hashtag "#thewalkingdead" for three weeks from 2015-11-07 21:00 EST to 2015-11-23 21:00 EST. As this television show airs Sundays at 21:00 EST, of particular importance is the tweets occurring around this time. We therefore restrict our measurements to 24 hours prior to and 24 hours after the television show's airing for the above three week time frame.

These measurements give rise to the time series presented in Figure 1. From this plot it is clear that, due to the weekly episodic nature of the television show, there is a seasonal component to this time series. It also appears that there is a trend component. Thus, in order to fit a stationary time series to the underlying data, we will need to first apply transformations to the data.

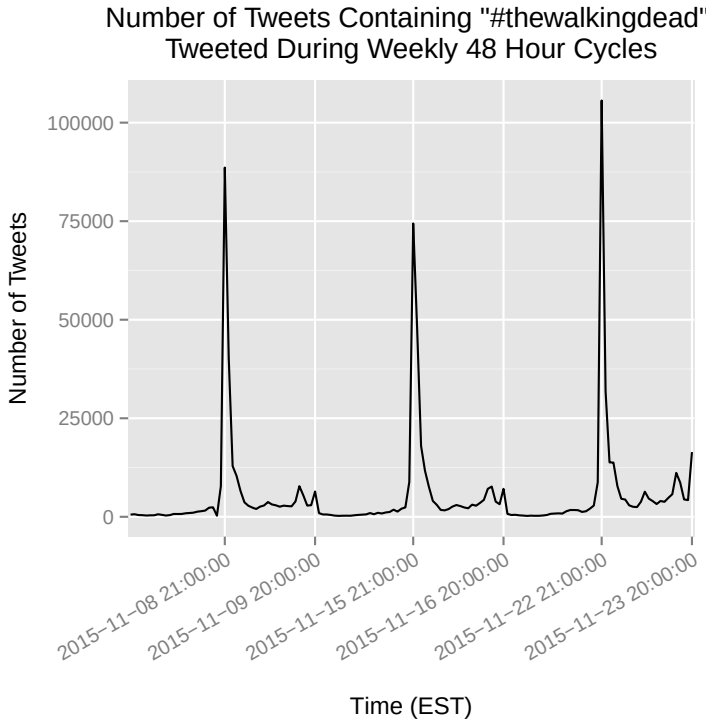

FIGURE 1. Time series plot of number of tweets containing the hashtag #thewalkingdead over three 48 hour cycles.

The details behind the process of fitting a stationary time series model to this data is handled in Section 3.

3. Model Fitting

We now wish to fit a time series model to the data described in Section 2. As can be seen from Figure 1, there are seasonal and trend components present in the underlying data set and what also appears to be non-constant variance.

Thus, we will need to apply transformations to the time series in order to determine the underlying time series model.

3.1. Transformations Let $\{X_t\}$ for $t = 1, 2, \ldots, 144$ denote the observations of the time series described in Section 2. To remove the non-constant variance we apply the Box-Cox transformation log to the observations $\{X_t\}$ to arrive at the mean-corrected data $Y_t = \log(X_t) -$

$E(\log(X_t))$ for $t = 1, 2, \ldots, 144$. As can be seen from Figure 2, this transformation has made the variance of the data constant.

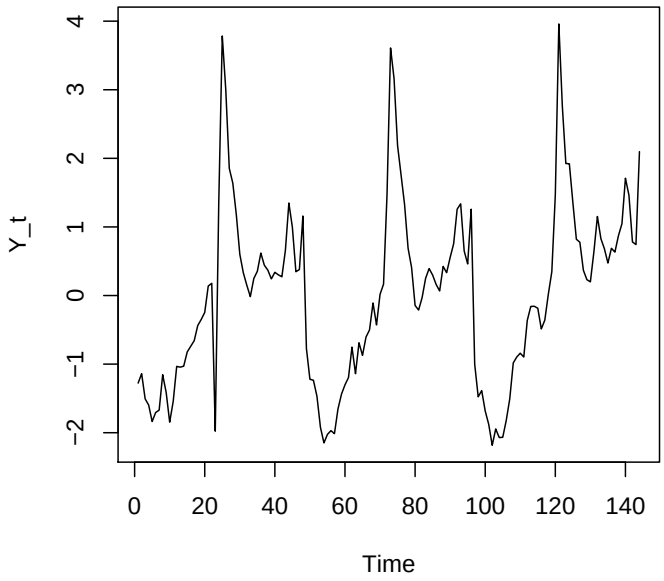

FIGURE 2. Plot of Box-Cox transformed data $Y_t = \log(X_t) - E(\log(X_t))$.

The seasonality and trend noticed with the untransformed data is still present in the transformed data. Through differencing, we hope to characterize the underlying seasonality and trend of the data.

Knowing that the observations come from data with a period of 48 hours, it makes sense to remove the seasonality by applying the differencing operator $(1 - B^{48})$ to Y_t. Applying this transformation results in Figure 3.

As can be seen from the figure, the seasonality has been removed. The plots of the ACF and the PACF of $(1 - B^{48}) Y_t$ are found in Figure 4.

These plots suggest that a seasonal AR model of order 1 or 3 may be appropriate as the ACF slowly decreases to 0 while the PACF stops abruptly after lag 3.

However, as can also be seen in Figure 3, there is still a trend component present. This trend appears linear suggesting that applying the

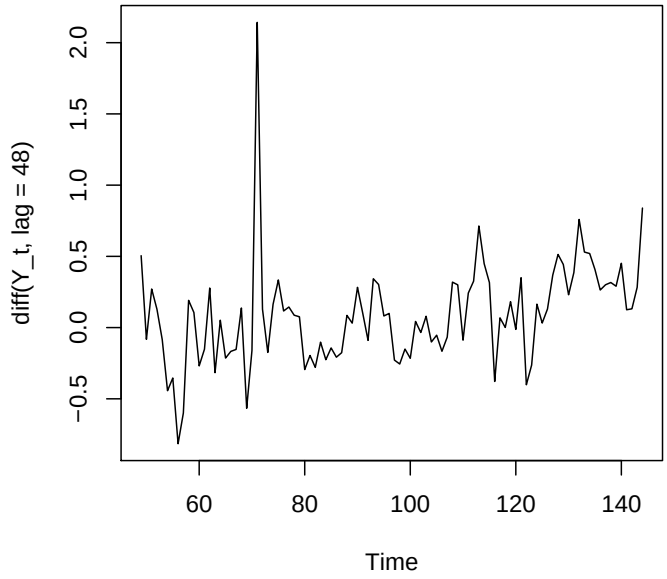

FIGURE 3. Plot of data $(1 - B^{48})\, Y_t$.

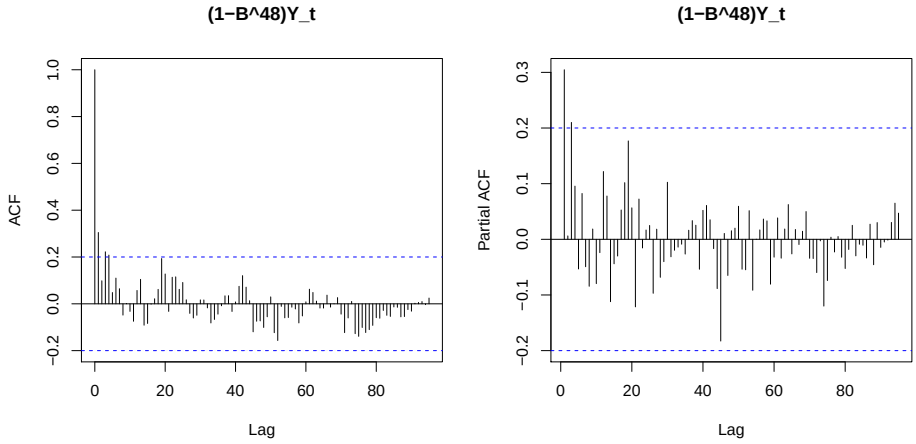

FIGURE 4. ACF and PACF of $(1 - B^{48})\, Y_t$.

difference operator $(1 - B)$ to $(1 - B^{48})\, Y_t$ will remove this trend. The results of applying the operator are found in Figure 5.

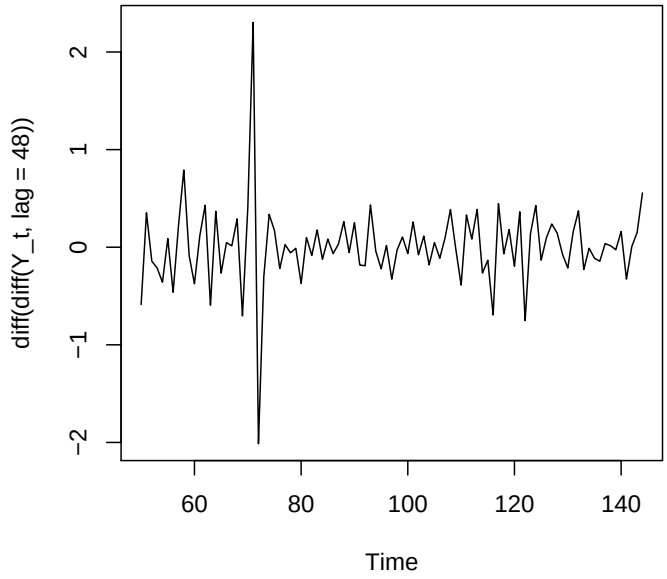

FIGURE 5. Plot of data $(1-B)\left(1-B^{48}\right)Y_t$.

The plot in Figure 5 shows that the trend component has now been removed after applying this difference operator. The plots of the ACF and the PACF of $(1-B)\left(1-B^{48}\right)Y_t$ are found in Figure 6.

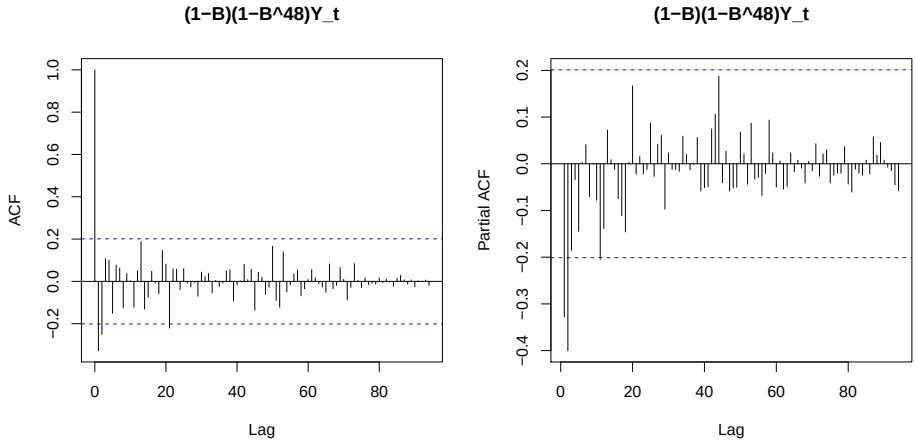

FIGURE 6. ACF and PACF of $(1-B)\left(1-B^{48}\right)Y_t$.

These plots suggest a possible AR model of order 2 for the trend component as the ACF slowly decreases to 0 while the PACF stops abruptly after lag 2. It is also possible for an MA component to be present so we will investigate models of the form ARMA(p, q) for $p, q = 1, 2$ for the trend component.

3.2. Model Selection The findings from Section 3.1 suggest fitting the data $\{Y_t\}$ to a seasonal ARIMA model. Specifically, a model of the form

$$\text{SARIMA}(p, 1, q) \times (P, 1, 0)_{48}$$

where $p, q \in \{0, 1, 2\}$ and $P \in \{1, 3\}$.

Using the programming language R, we examine the AIC statistic associated to each of the different possible models suggested and choose the model that minimizes this statistic. Due to the limitations of the available hardware, we were unable to fit SARIMA models to the data in which $P = 3$ for such a large period. Thus, we present the results of testing for $P = 1$ alone.

p	q	AIC
0	1	77.00203
0	2	75.06627
1	0	99.46614
1	1	`NaN`
1	2	78.04194
2	0	85.23537
2	1	77.33328
2	2	78.92953

TABLE 1. AIC values for different possible SARIMA$(p, 1, q) \times (1, 1, 0)_{48}$ models fitted to the data $Y_t = \log(X_t) - E(\log(X_t))$.

Note that hardware limitations also prevented us from fitting a model of the form SARIMA$(1, 1, 1) \times (1, 1, 0)_{48}$ to the data so that has been omitted from consideration as well. Table 1 suggests that, if we are to select a model based on minimizing the AIC statistic, we should choose to fit the data to a SARIMA$(0, 1, 2) \times (1, 1, 0)_{48}$ model.

Doing so produces the following output in R:

```
Coefficients:
          ma1      ma2      sar1
      -0.6768  -0.2257  -0.1586
s.e.   0.1097   0.1100   0.1441

sigma^2 estimated as 0.1152:  log likelihood = -33.53,
aic = 75.07
```

The above gives the associated p-values for the coefficients of the model:

```
        ma1          ma2          sar1
6.878613e-10 4.021464e-02 2.709825e-01
```

Using a significance level of $\alpha = 0.05$ suggests that the SAR(1) coefficient is not significant. However, removing the seasonal component and fitting the model to an MA(2) model results in a worse fit. Thus, we elect to keep the seasonal component in the model.

Thus, we fit our data $Y_t = \log(X_t) - E(\log(X_t))$ to the model

$$Y_t = \Phi_1 Y_{t-48} + Z_t + \theta_1 Z_{t-1} + \theta_2 Z_{t-2} \quad Z_t \sim \mathrm{WN}(0, \sigma^2)$$

$$\Phi_1 = -0.1586, \quad \theta_1 = -0.6768, \quad \theta_2 = -0.2257, \quad \sigma^2 = 0.1152. \tag{1}$$

We verify that these residuals are indeed a white noise process by examining the plot of their ACF in Figure 7.

As almost all of the values fall within the 0-bound, we conclude that this process is most likely a white noise process.

4. Forecasting

Now that we have fitted our data $\{Y_t\}$ to a suitable model, the next reasonable step is to use the model to provide a forecast of the next 48 hour cycle ,i.e. for the time period of 2015-11-28 21:00 EST - 2015-11-30 21:00 EST.

The programming language R provides many tools to forecast data. One such tool is the `forecast` package, which we employ. Using the model provided by (1), we provide the model in an acceptable form to the function `predict` as well as the desired number of lags, e.g. 48, and arrive at the forecasted values. As the model is for the transformed data $Y_t = \log(X_t) - E(\log(X_t))$, we must apply the inverse transformation to the forecasted values. Doing so gives us the table of values found in Appendix A along with 90% confidence intervals. A plot of these forecasts combined with the original data can be found in Figure 8.

The shape of the forecast seems to follow the same shape as the previous cycles suggesting that the forecasts seem plausible. Also note

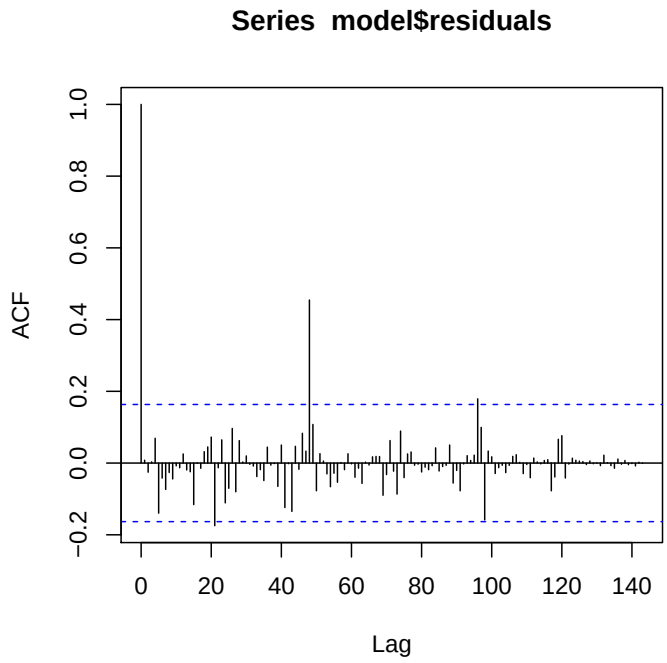

FIGURE 7. Plot of ACF of Z_t in (1).

the sensitivity of the forecast at the time of interest, i.e. at 2015-11-29 21:00 EST.

5. Conclusion

Twitter's streaming API along with our custom program allowed us to gather tweets containing the hashtag "#thewalkingdead" over a three week time period. Isolating that data to 48 hour cycles centered at the show's air time gave us a time series to analyze, namely the number of tweets containing the hashtag over that three week period.

Utilizing the time series plots of our data $\{X_t\}$ combined with differencing techniques allowed us to determine a potential family of seasonal ARIMA models to fit to the transformed mean-corrected data $\{Y_t\} = \{\log(X_t) - E(X_t)\}$. Choosing the model that minimizes the AIC statistic, we arrived at the SARIMA model in (1) with period 48.

With a model describing the transformed data, we applied R's forecasting software to create a forecast of the next 48 hour cycle for the data. The shape of the forecast seems plausible, however. The forecast is not very accurate in predicting the number of tweets that will occur during the show's airing due to the large variance between the bounds

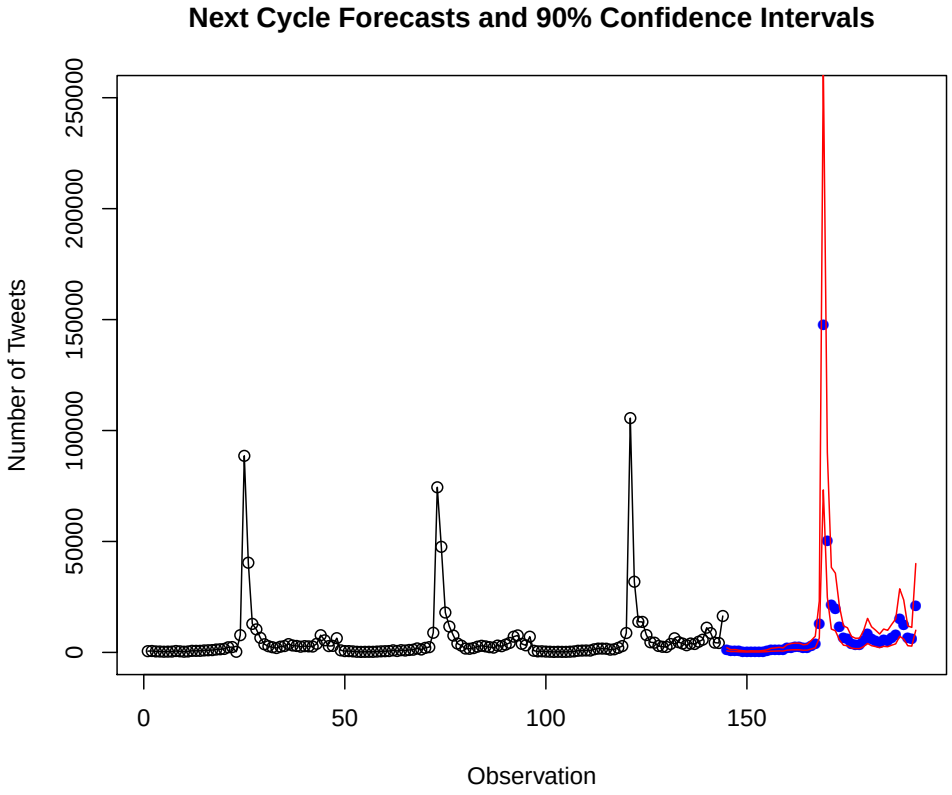

FIGURE 8. Plot of forecasted data for the next 48 hour cycle for data X_t.

of the 90% confidence interval as can be seen from Figure 8 and entry 169 of the table in Appendix A.

Thus, the model currently does not achieve its main goal of accurately predicting the number of tweets at the show's future airing. However, the techniques outlined in this report can be repeated with more Twitter data to derive a more accurate model.

A. Future Forecasts of Twitter Data

The first 24 hours of the forecasted 48 hour cycle.

Period	Lower Bound	Forecast	Upper Bound
145	696.661029707395	1289.74016300334	2127.93027856002
146	371.205179144431	707.042650629641	1200.19457955701
147	398.990757929047	761.886863969782	1296.55993960765
148	298.389024387854	571.218000037578	974.530391505982
149	235.260017809285	451.496034618654	772.207861151707
150	175.104938912735	336.888687926832	577.62891424388
151	217.633776492968	419.751298882938	721.493233577037
152	197.028439827803	380.949968593457	656.419432190296
153	195.834380936897	379.574340784209	655.660796189112
154	254.261169783136	494.029058309949	855.459417118332
155	342.348756627779	666.809028053149	1157.4670876602
156	541.839965322973	1057.93693787983	1840.86944832303
157	591.074352049692	1156.86506892861	2017.89128345538
158	662.938385055221	1300.65307779798	2274.17689679211
159	593.618750247393	1167.45337812907	2046.18994117121
160	996.764813380513	1965.00756178743	3452.31342121765
161	1145.93011950655	2264.46317381743	3987.92381850244
162	1195.64781213398	2368.32912746162	4180.76062731866
163	1182.90156687488	2348.63292164139	4155.81441111673
164	974.897576330122	1940.21306877937	3441.23493944191
165	1029.04480314982	2052.79131978179	3649.46988969812
166	1513.72135344513	3026.72364118822	5393.54198603356
167	2039.88049439694	4088.31301699942	7302.27219528937
168	6431.44443826672	12919.8309270562	23130.2113965612

The latter 24 hours of the forecasted 48 hour cycle.

Period	Lower Bound	Forecast	Upper Bound
169	73234.5161866536	147458.349966062	264604.597838768
170	24830.3114773495	50111.4945676466	90129.5094293664
171	10504.7941538785	21249.1126483882	38306.1508328149
172	9750.07661699606	19767.6849806556	35717.2536402957
173	5657.00381745176	11495.4318154739	20817.9792381468
174	3248.31018616337	6615.82315271962	12008.3889027812
175	2988.56663035889	6100.6131816249	11098.352076634
176	1936.31134917626	3961.56240087113	7223.24381900657
177	1703.46912831153	3493.03541272558	6383.31348424044
178	1702.6180029853	3499.12965123463	6408.80874021746
179	2567.99280062991	5289.4123102931	9709.46956476303
180	4042.13442983339	8344.3387520691	15351.3850092768
181	3026.74054804063	6262.11015363218	11546.2424872823
182	2611.40115790704	5414.77289548684	10006.0199303995
183	2153.93111344956	4476.06570543426	8289.63788420334
184	2726.04897907648	5677.44795273481	10537.7315794995
185	2558.65415211628	5340.51159724128	9934.10168548733
186	3233.09489592031	6762.98509283016	12607.6130659856
187	3855.33164127782	8082.15895610075	15099.6713928233
188	7286.0431135438	15307.3510306129	28660.4371177222
189	5975.47118800499	12581.1442799613	23607.0974232007
190	3014.16940383741	6359.93056229604	11959.4595514178
191	2826.93700075681	5977.70742669783	11264.9078461764
192	10006.7275494762	21205.1579967342	40046.578166815