Modeling revenue generated by Chicago's red light cameras

Michael Fagan

1 Introduction

According to the Insurance Institute for Highway Safety, communities in 23 states and the District of Columbia utilize cameras to enforce red light traffic laws. Many studies have been done with regards to the effect of red light cameras (RLCs) on traffic safety (Erke 2009). These studies indicate mixed results in this regard, with a reduction in right-angle "T-bone") collisions offset by an increase in rear-end collisions.

Little research, however, has been done on the efficacy of red light camera systems as revenue-generating engines. In this study, we look at the City of Chicago, which has operated a red light camera system since 2003. Chicago's red light camera program has proven controversial. Three figures instrumental in developing Chicago's red light camera system have pleaded guilty or have been convicted of federal charges of bribery. In addition, in March of 2010, Chicago Alderman Ed Burke told the *Chicago Tribune*, "[The red light cameras] are a money machine, that's all. Period." (Byrne 2010) To that point, Chicago drivers ticketed for red light violations must pay a fine of $100 for each infraction.

To analyze the revenue-generating aspects of Chicago red light cameras, we use time series analysis on the total number of tickets issued by the system and the mean number of tickets issued per camera. In this way, we develop a model that may aid in forecasting the number of tickets issued and, hence, the amount of revenue generated, by the city's red light cameras in future years.

2 Data

Red light camera data was provided to the *Chicago Tribune* by the City of Chicago and is available via the *Chicago Tribune* website (link available in Acknowledgments). The data consists of 4,174,770 red light camera tickets issued between January 1, 2007, and March 2, 2014. Each observation includes the ticket identification number, camera location (by address), timestamp, license plate number, car type, and state in which the vehicle is registered. Latitude and longitude coordinates were appended by cross-referencing red light camera data available via the City of Chicago Data Portal or by the coordinates returned by entering the camera's address into Google Maps.

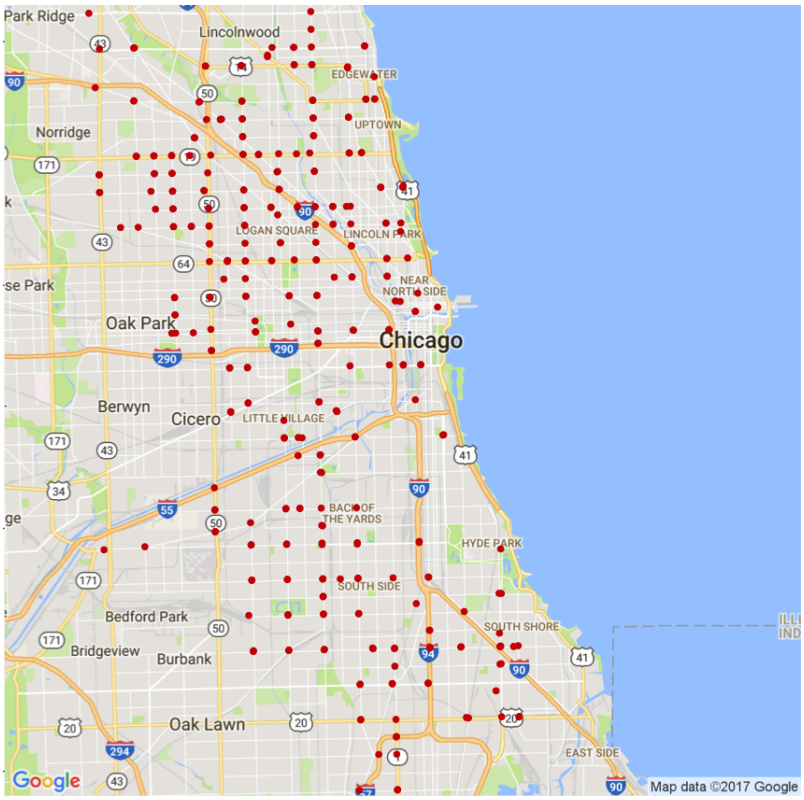

Figure 1: Intersections with at least one red light camera in the Chicago area between 2007 - 2014

By January 1, 2007, the City of Chicago had installed 56 red light cameras at intersections around the city. More cameras were added every month

until January 2010 when 314 cameras had been installed. From January 2010 through February 2014, the number of cameras remained steady, never falling below 291 and never rising above 325. A plot of the 332 of the 336 cameras installed between 2007 and 2014 can be seen in Figure 1.

As seen in Figure 2, the number of tickets issued per month by Chicago's red light camera system shows a clear seasonal pattern. Tickets issued peaks in the summer months and hits a valley in the winter. This agrees with our intuition; we expect more drivers on the road during the more agreeable Chicago summers. A plot of the mean and median tickets issued per camera per month in Figure 3 also shows the same seasonal pattern. Additionally, the mean is consistently greater than the median, suggesting some hot spot intersections pulling the mean up. The steep downward trend in the first two years could suggest that the City of Chicago placed initial red light cameras at the busiest or most problematic intersections and/or that driver behavior began adjusting to the system. The overall downward trend might be indicative of such an adjustment.

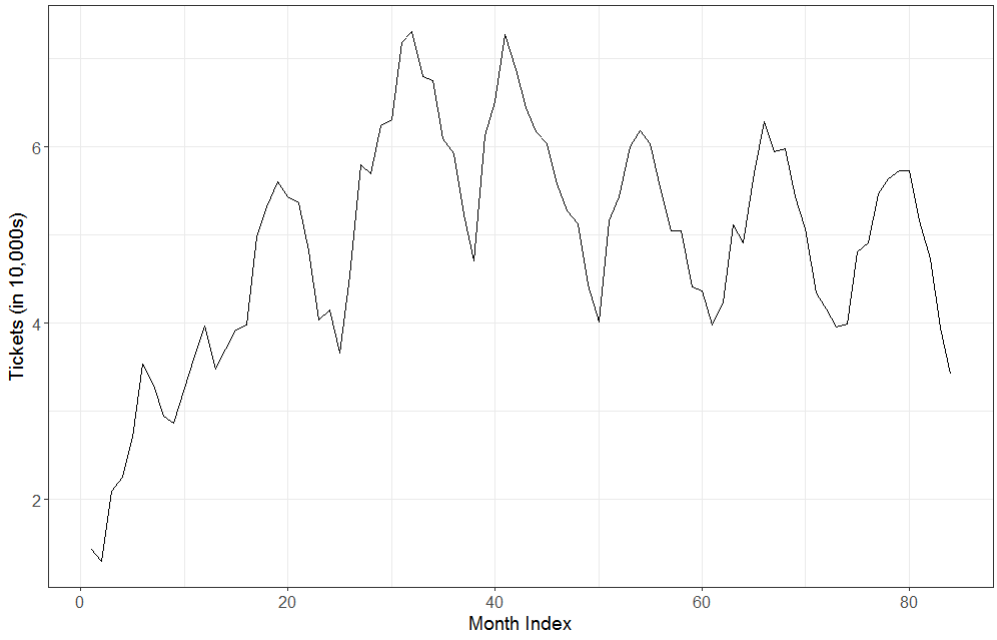

Figure 2: Total tickets issued per month

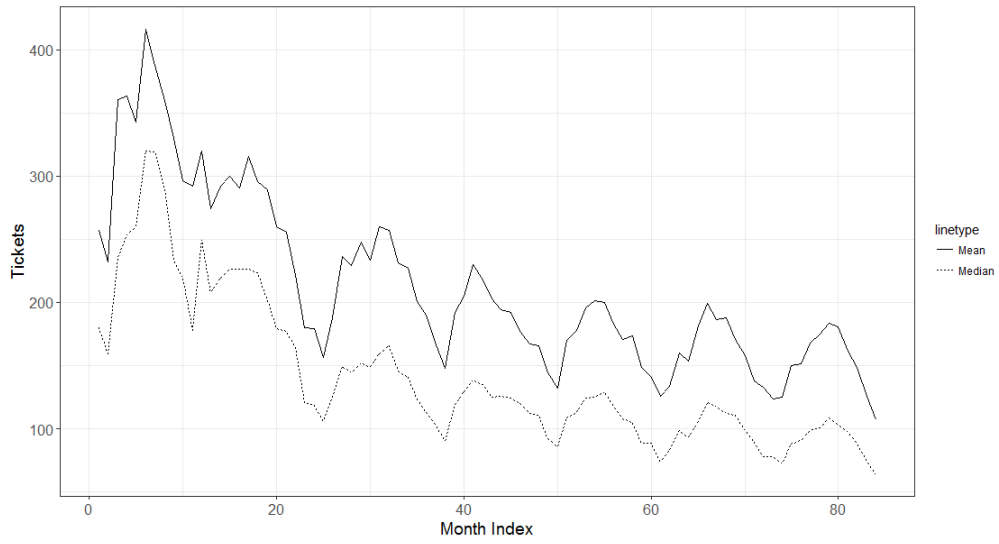

Figure 3: Mean (dotted) and median tickets per camera per month

3 Time Series Models

3.1 Introduction

A stationary time series is a necessary component for a time series analysis. Stationarity implies that parameters such as mean and variance remain constant over time. Mathematically, a time series $\{Y_t\}$ is stationary if the joint distribution $Y_{t_1}, Y_{t_2}, ..., Y_{t_n}$ is the same as the joint distribution $Y_{t_1-k}, Y_{t_2-k}, ..., Y_{t_n-k}$ for all choices of time points $t_1, t_2, ..., t_n$ and all choices of time lag k. (Cryer and Chan 2008)

Some nonstationary time series can be transformed into stationary time series through differencing. The first difference of a time series is defined as $W_t = \Delta Y_t = Y_t - Y_{t-1}$, the second difference as $W_t = \Delta^2 Y_t = \Delta(Y_t - Y_{t-1}) = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$, and so on. An ARMA model applied to d differenced data is called an autoregressive integrated moving average process, abbreviated ARIMA(p,d,q). (Bisgaard and Kulahci 2011)

The covariance and correlation between two random variables X and Y are defined by, respectively, $\text{Cov}(X, Y) = E[(X - \mu_x)(Y - \mu_y)]$ and $\text{Corr}(X, Y) =$

$\frac{\text{Cov}(X,Y)}{\sqrt{\sigma_X^2}\sqrt{\sigma_Y^2}}$.

By extension, the covariance between Y_t and Y_{t-1} is $\text{Cov}(Y_t, Y_{t-1}) = E[(Y_t - \mu)(Y_{t-1} - \mu)]$, where $\mu = E(Y_{t-1}) = E(Y_t)$ due to stationarity. We can also find the covariance of observations k lags apart by $\text{Cov}(Y_{t+k}, Y_t) = E[(Y_{t+k} - \mu)(Y_t - \mu)]$. Then, the autocovariance function is defined as

$$\gamma(k) = E[(Y_{t+k} - \mu)(Y_t - \mu)],$$

and the variance of the time series is $\gamma(0)$. We can estimate the mean and variance, respectively, through the sample by $\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} Y_t$ and $\hat{\sigma}^2 = \frac{1}{T}\sum_{t=1}^{T}(Y_t - \bar{Y})^2$. (Bisgaard and Kulahci 2011)

Drawing from the correlation function of two random variables, we define the autocorrelation function (ACF) for a stationary time series as

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)}.$$

The ACF helps identify times series models as it reveals how correlated observations k lags apart are. In practice, we cannot know the true value of an ACF and, instead, must rely on an estimate from the data using

$$\hat{\gamma}(k) = \frac{1}{T}\sum_{t=1}^{T-k}(Y_{t+k} - \bar{Y})(Y_t - \bar{Y})$$

and

$$\hat{\rho}(k) = \frac{\hat{\gamma}(k)}{\hat{\gamma}(0)}.$$

Another tool to aid the identification of a time series model is the partial autocorrelation function (PACF). The PACF differs from the ACF in that it measures the autocorrelation between Y_t and $Y_t + k$, while controlling for the values at all shorter lags. Further explanation of the PACF is beyond the scope of this paper.

3.2 ARMA and ARIMA Models

The autoregressive moving average time series model - denoted ARMA(p,q) - is defined as

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + e_t - \theta_1 e_{t-1} - ... - \theta_q e_{t-q} \ (1),$$

where $\{Y_t\}$ is the observed time series, t is the index of time periods where $t = 0, 1, ..., T$, $\{\phi_p\}$ is the series of autoregressive (AR) coefficients, $\{e_t\}$ is the series of "white noise" errors (i.e., a sequence of identically distributed, zero-mean, independent random variables), and $\{\theta_q\}$ is the series of moving average (MA) coefficients. We say $\{Y_t\}$ is a mixed autoregressive moving average process of orders p and q, respectively. (Cryer and Chan 2008)

Finally, we must also consider potential seasonality in the data. Seasonality occurs when observations show dependence on past observations in a cyclical pattern. For instance, monthly sales in a retail store may be dependent on both the prior month and the same month one year ago. A seasonal ARMA model has the form

$$Y_t = \phi_1 Y_{t-s} + \phi_2 Y_{t-2s} + ... + \phi_p Y_{t-ps} + e_t - \theta_1 e_{t-s} - ... - \theta_q e_{t-qs} \ (2),$$

where s is the time interval between observations. For our retail store, we would use $s = 12$. (Bisgaard and Kulahci 2011)

In addition, just as we have for nonstationary nonseasonal models, we may need to add a seasonal difference $\Delta_s = Y_t - Y_{t-s}$. Then, we denote a differenced seasonal model as an ARIMA(p,d,q) x $(P,D,Q)_s$.

Once we have fit a model to the data, we turn to the Akaike information criterion (AIC). The AIC is defined as

$$\text{AIC} = 2k - 2\ln(\hat{L}),$$

where $\hat{L}$ is the maximized value of the likelihood function of the model and k is the number of estimated parameters. AIC values have no intrinsic meaning, but we can compare AIC values of different models to help determine model

quality. When comparing two models, we prefer the model with the lower AIC value.

We also consider how parsimonious a model is. All other things being equal, we prefer models with both fewer components and fewer differences to the data. Each additional component or difference to the model increases the likelihood of "overfitting."

4 Analysis and Results

Our goal is to model the red light camera time series data in order to forecast future observations, using a seasonal ARIMA modeling procedure as in equation (2). Our first step is to check for stationarity and difference the data if necessary. Figure 8 shows four plots of the total number of tickets issued by the system: 1) the original time series data, 2) the first difference of the data, 3) the seasonal difference ($s = 12$) of the data, and 4) the first difference and the seasonal difference ($s = 12$) of the data.

As we can see, plots (1) and (3) are non-stationary. Plots (2) and (4) both show stationarity. With parsimony in mind, we move forward with a single difference to the data.

Our next step is to find an appropriate ARIMA model using the autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the differenced data. As we can see in Figure 5, the shows significant spikes at lags 12, 24, and 36, and a moderate spike at lag 48. Table 4.1 from (Bisgaard, Kulahci) suggests an autoregressive (AR) component with $p = 4$, giving us an ARIMA(0,1,0) x $(4,0,0)_{12}$ model.

The model produces coefficients and AIC found in Table 1. To check our model, we plot both the residuals and the ACF and PACF of the residuals. As we can see in Figure 6, the residuals do not appear to show a pattern and the ACF and PACF do not show any significant spikes.

We can attempt to improve our model by iteratively changing our order parameters. In this manner, we do not find any better models.

Alternatively, we also use the *auto.arima* function from the R *forecast* package. The *auto.arima* function combs through combinations of order pa-

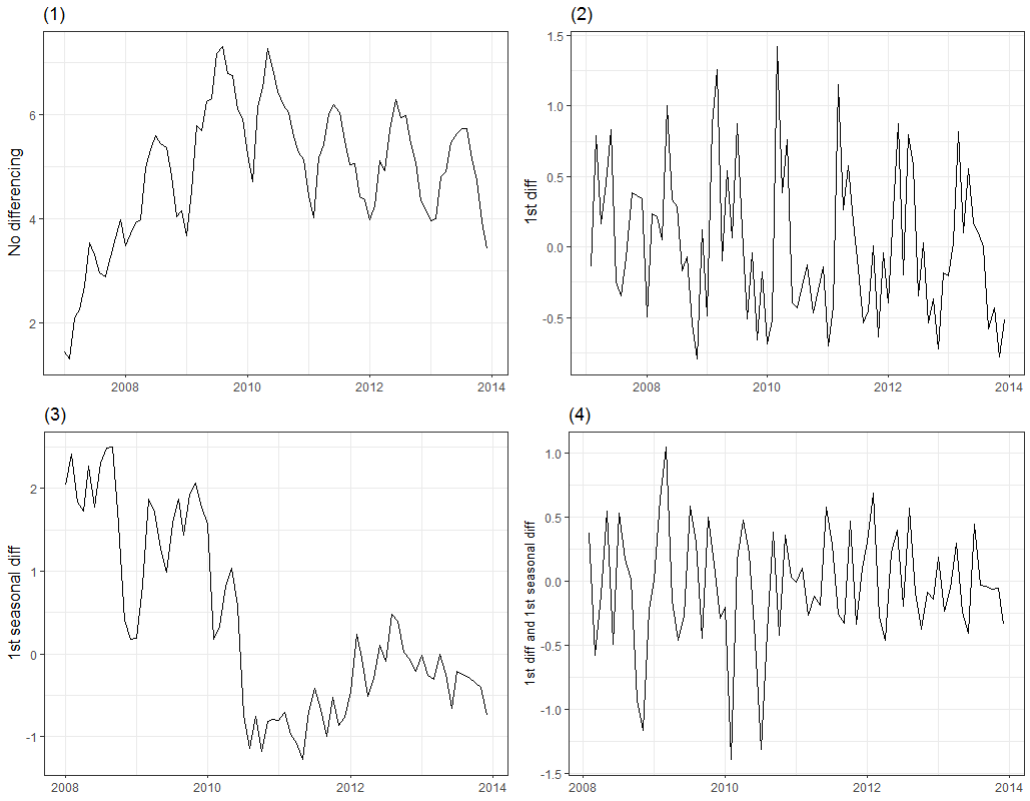

Figure 4: (1) original time series, (2) first difference, (3) seasonal difference, (4) first and seasonal difference

rameters to find the optimal model fit. Under this method, we have an ARIMA(1,1,0) x $(2,0,0)_{12}$, with coefficients and AIC values found in Table 1. This model does not provide for better AIC values, but does provide us with a more parsimonious model.

Figure 7 shows our two fitted models plotted on top of the original time series.

In the same manner, we found "manual" and *auto.arima* models for the mean number of tickets per camera. A comparison of those models can be seen in Table 2. Other support plots can be found in the Appendix.

For both the total and mean tickets, we choose the more parsimonious model in lieu of the model that minimizes the AIC, respectively, the *auto.arima* and

Manual: ARIMA(0,1,0)x(4,0,0)$_{12}$		
Term	**Coefficient**	**Std Error**
SAR1	0.2745	0.1203
SAR2	0.0237	0.0996
SAR3	0.1666	0.1323
SAR4	0.4205	0.1445
AIC: 86.26		
Auto Arima: ARIMA(1,1,0)x(2,0,0)$_{12}$		
Term	**Coefficient**	**Std Error**
AR1	0.1694	0.1099
SAR1	0.4723	0.1107
SAR2	0.1937	0.1205
AIC: 93.28		

Table 1: Total tickets model comparison

Manual: ARIMA(0,1,1)x(2,0,0)$_{12}$		
Term	**Coefficient**	**Std Error**
MA1	-0.2588	0.1193
SAR1	0.3132	0.1260
SAR2	0.4116	0.1405
AIC: 760.92		
Auto Arima: ARIMA(2,1,2)x(2,0,0)$_{12}$		
Term	**Coefficient**	**Std Error**
AR1	-1.3712	0.1622
AR2	-0.9002	0.0949
MA1	1.2404	0.2348
MA2	0.6031	0.1733
SAR1	0.2838	0.1167
SAR2	0.4762	0.1284
AIC: 751.39		

Table 2: Mean tickets model comparison

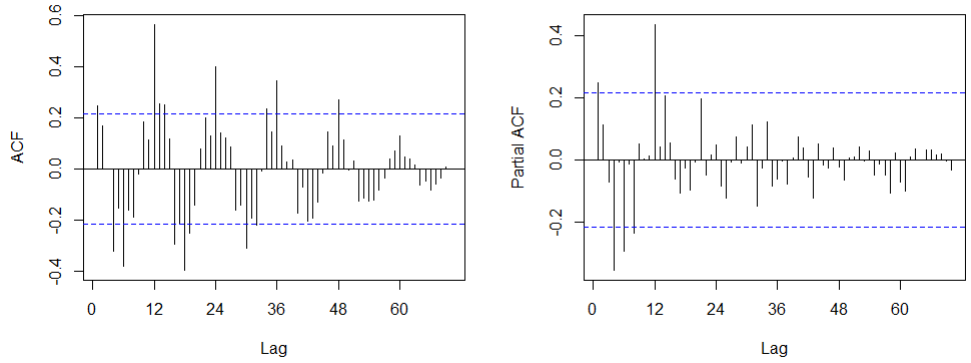

Figure 5: ACF and PACF of the differenced data

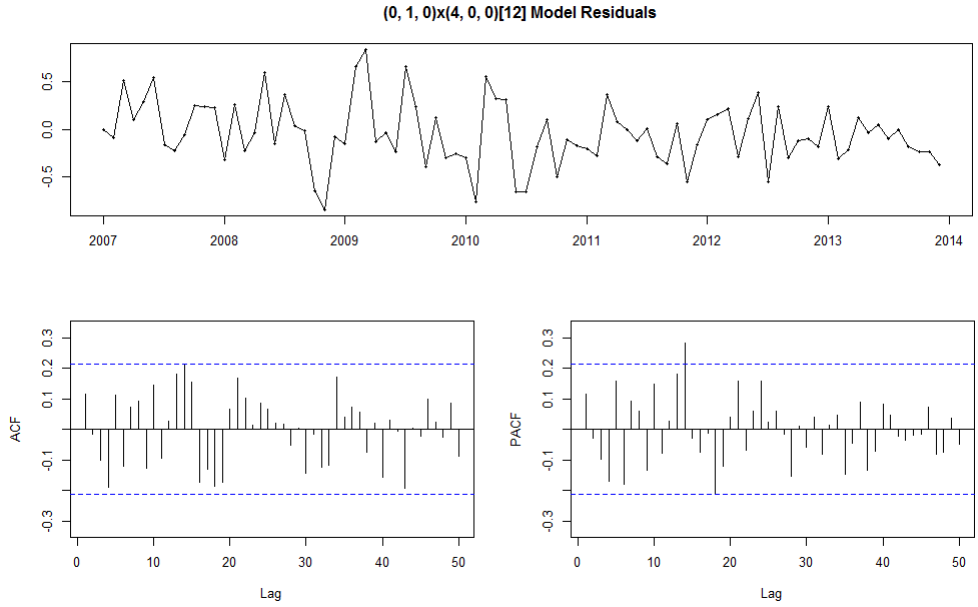

Figure 6: Residual checks

"manual" models. In the case of mean tickets, the "manual" model is far more parsimonious: three model components and one difference of the data versus six model components and one difference of the data. The decision is more difficult in the case of the total tickets; the only difference between the

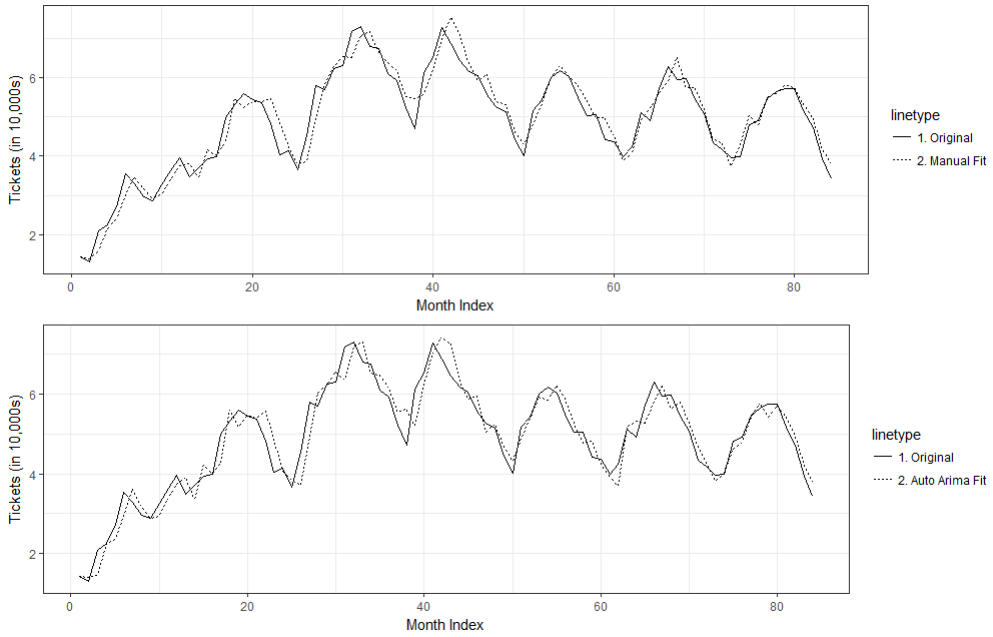

Figure 7: Fitted models against the original time series - "manual" (top) vs. *auto.arima*

two models is one additional model component in the "manual" model.

As seen in Figure 8, we plot preliminary forecasts of both the total and mean models using the *forecast* function in the R *forecast* package. Both forecasts suggest a continued slight downward trend. Given a consistent system (stable camera installation, no major changes in the auto industry, etc.) we intuitively expect a "leveling off" in both trends, though it's difficult to ascertain how long it would take for the system to reach that equilibrium.

5 Conclusion

We have found that we can reasonably model Chicago's red light camera system using time series analysis. Preliminary forecasts of both our models suggest a slight downward trend, though we intuitively would expect this trend to level off at some point, assuming the system remains relatively stable. If that's the case, the red light camera system would provide the City

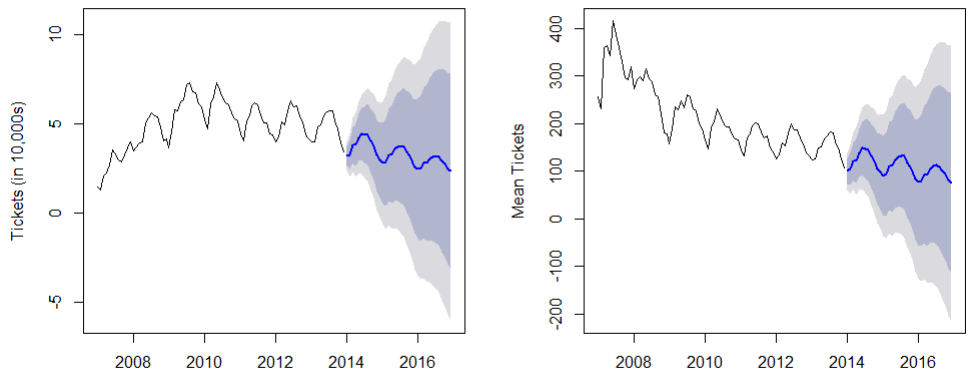

Figure 8: Preliminary forecasts of total (left) and mean tickets

of Chicago a reliable and steady source of income, outside of whatever traffic safety benefits the system may provide.

As reported by the *Chicago Tribune*, 74% of people responding to a January 2015 poll asking "Do you believe the red light program should be..." answered "Eliminated" or "Reduced." (Ruthhart 2015) Despite this widespread opposition, the red light camera system remains intact. This resistance is more understandable given Chicago's financial situation and the $60 million the red light system brings in annually.

6 Future Research

We consider this research incomplete. Directly, more work can be done to fine tune our models in order to improve forecasts. In this regard, models were created using Automatic Forecasting System's *Autobox* package for R. AFS's proprietary software proved to be too much of a "black box" for this paper, but their software is more sensitive to seasonal pulses and other times series phenomena. We've included details of these models in the Appendix.

We also looked at potential spatial relationships in the data. We divided the cameras into four "quadrants." Interstate 290 separates the north and

south sides of the city, while east and west is divided by the I-90 corridor in the north and Ashland Ave. in the south. We found that the mean tickets per camera was consistently higher in the southeast "quadrant" after the middle of 2008 while the northeast "quadrant" had the second highest mean for much of that same period of time. Further research could determine whether this is being caused by driver behavior, camera placement, and/or other factors. Preliminary plots can be found in the Appendix.

The percentage of tickets issued to vehicles registered out of state may be an indicator of the change in driver behavior of local residents. A preliminary plot (available in the Appendix) suggests the percentage of tickets issued to out-of-state vehicles has increased over the time frame of the data. We also plotted the locations of the top quartile of cameras in terms of percentage of tickets issued to out-of-state vehicles. The vast majority of these locations lay in the "northeast" quadrant referenced above. We also noted a camera in Hyde Park, home to the University of Chicago and a likely higher percentage of residents with vehicles registered in other states. Cross-referencing in-state license plates with the county the vehicle is registered in may also aid in research focusing on "tourist vs. local" driving behavior and the effect of local tourist attractions like the Taste of Chicago, the Air and Water Show, sporting events, and so on.

Finally, one factor that may explain the overall downward trend in tickets issued is a declining population in the City of Chicago. The 2010 Census saw a 6.8% decrease in population since 2010, and a March 23, 2017, *Chicago Tribune* article reports that the Chicago-Naperville-Elgin metropolitan area lead the United States in population loss in 2016, the second consecutive year in which the area lost residents. This might explain the downtrend, should the population loss also result in a decrease in vehicles on the road.

7 Acknowledgments

I owe a very special thanks to Dr. Jing Wang, my Honors College Fellow and Project Supervisor. Her input, guidance, and advice has been indispensable, and her patience and support over the course of this project has been very much appreciated.

Many thanks to the *Chicago Tribune* both for making their data publicly

available and their reporting on this issue.

Despite ultimately not using his software in the main writeup, I owe Tom Reilly and Automated Forecasting Systems a huge thanks for their generosity. Mr. Reilly provided the *Autobox* R package at minimal cost and was always readily available to answer questions, including while on vacation.

This project would have been more frustrating without Rob Hyndman's *forecast* package for R and the various R tools and packages provided by Hadley Wickham.

8 Appendix

8.1 Red Light Camera Details

By law, Chicago red light cameras must catch motorist as they enter an intersection under a red light. To do this, the camera system utilizes magnetic induction loop technology. Sensors embedded in the pavement[1] trigger their respective camera if a motorist approaches an intersection above a certain speed threshold (typically 13 miles per hour). The video camera then records the vehicle traveling through the intersection and includes the traffic light in frame. A separate still camera photographs the vehicles license plate.

Employees of the red light camera vendor – Redflex during the timeframe of our dataset; Xerox thereafter – review the video. If they determine a violation has been committed, they forward the video and information to a second city contractor for further review. If the second city contractor also determines a violation has been committed, the license plate number is sent to state authorities, who then retrieve vehicle ownership information and issue a ticket by mail.

According to the City of Chicago, the camera system does not photograph vehicles making left turns who have already entered the intersection before a light turned red, nor does it photograph vehicles making legal right turns on red. A vehicle must make a full stop in order to take a legal right turn

[1] An above-ground radar system may have been adopted after the time frame in our dataset.

on red. The system does photograph vehicles who make a rolling stop at an intersection prior to a right turn on red.

According to the City of Chicago, intersections chosen for red light camera installation are chosen as follows. Intersections with the highest incidents of total crashes are identified based on data from the Chicago Crash Database. Information on types of crashes (angle, rear-end, head-on, etc.) and traffic volume are compiled for these high-incident intersections. Crash Rate and Angle Crash Rate statistics are calculated for these intersections and then rank-ordered by Angle Crash Rate. This list is cross-referenced against intersections with existing red light cameras. Intersections with the highest Angle Crash Rates are given priority for red light camera installation. The city then conducts field experiments to determine the constructability of a red light camera at an intersection. Constructability factors include sight lines, pavement condition, power sources, proximity to other red light cameras, etc.

8.2 Data

Our data is available in two places. First, the *Chicago Tribune* has made the raw data available on its website, which can be found at `http://goo.gl/gGNrPz`. In addition, we have made the cleaned-up data available on Dropbox (login required) via `http://goo.gl/eKkG9x`.

8.3 R Code

```
library(dplyr)
library(forecast)
library(ggmap)
library(ggplot2)
library(zoo)

### Reading tickets.csv ###
tickets <- read.csv("tickets.csv")

### Plotting cameras ###
camera.list <- unique(tickets[, c('latitude', '
   longitude')])
```

```
map3 <- get_map(location = c(lon = -87.64, lat =
   41.8540375), source = "google", maptype = "
   roadmap", zoom = 11)

ggmap(map3) +
  geom_point(data = camera.list, mapping = aes(x =
     longitude, y = latitude), size = 2) +
  theme(axis.title = element_blank()) +
  guides(color = guide_legend("Legend"))

### Aggregating tickets by month ###
tickets.month <- tickets %>%
  group_by(month.index) %>%
  summarize(
    count = n()
  )
tickets.month$count <- tickets.month$count / 10000
tickets.month <- filter(tickets.month, month.index <
    85)

### Plotting total tickets by month ###
ggplot(data = tickets.month) + geom_line(mapping =
   aes(x = month.index, y = count)) +
  labs(x = "Month␣Index", y = "Tickets␣(in␣10,000s)"
     ) +
  theme_bw() +
  theme(axis.text = element_text(size = 12), axis.
     title = element_text(size = 14))

### Converting total data to time series object ###
monthly.ts <- ts(tickets.month$count, start = c
   (2007, 1), frequency = 12)

### Differencing ###
diff.ts <-diff(monthly.ts, differences = 1)
                  # First difference
```

```
diff2.ts <- diff(monthly.ts, lag = 12, differences =
    1)          # First seasonal difference
diff3.ts <- diff(diff(monthly.ts, lag = 12))
                    # First difference plus first
   seasonal difference

### Multiplot function (via cookbook-r.com) ###

# Multiple plot function
#
# ggplot objects can be passed in ..., or to
   plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If
   present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3),
    nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go
   in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols
   =1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine
     layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from #
        of cols
```

```
    layout <- matrix(seq(1, cols * ceiling(numPlots/
       cols)),
                     ncol = cols, nrow = ceiling(
                        numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(
       layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions
         that contain this subplot
      matchidx <- as.data.frame(which(layout == i,
         arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row
          = matchidx$row,
                                      layout.pos.col
                                          = matchidx
                                         $col))
    }
  }
}

### Plotting original and differenced data ###

p1 <- autoplot(as.zoo(monthly.ts), geom = "line") +
   theme_bw() + theme(axis.title = element_text(size
    = 12), axis.title.x = element_blank(), axis.text
    = element_blank()) + labs(y = "No␣differencing",
    title = "(1)")
```

```
p2 <- autoplot(as.zoo(diff.ts), geom = "line") +
   theme_bw() + theme(axis.title = element_text(size
    = 12), axis.title.x = element_blank(), axis.text
    = element_blank()) + labs(y = "1st␣diff", title
   = "(2)")
p3 <- autoplot(as.zoo(diff2.ts), geom = "line") +
   theme_bw() + theme(axis.title = element_text(size
    = 12), axis.title.x = element_blank(), axis.text
    = element_blank()) + labs(y = "1st␣seasonal␣diff
   ", title = "(3)")
p4 <- autoplot(as.zoo(diff3.ts), geom = "line") +
   theme_bw() + theme(axis.title = element_text(size
    = 10), axis.title.x = element_blank(), axis.text
    = element_blank()) + labs(y = "1st␣diff␣and␣1st␣
   seasonal␣diff", title = "(4)")

multiplot(p1, p3, p2, p4, cols = 2)

### ACF and PACF of first difference data ###

par(mfrow = c(1, 2))
Acf(diff.ts, lag.max = 70, main = "")
Pacf(diff.ts, lag.max = 70, main = "")
par(mfrow = c(1, 1))

### Fitting (0,1,0) x (4,0,0)[12] model ###
manual.fit <- Arima(monthly.ts, order = c(0, 1, 0),
   seasonal = c(4, 0, 0), method = "ML")
manual.fit
tsdisplay(residuals(manual.fit), lag.max = 50, main
   = "(0,␣1,␣0)x(4,␣0,␣0)[12]␣Model␣Residuals")

### Preliminary forecast of
plot(forecast(manual.fit, h = 36))

### Generating model via auto.arima ###
### Auto.arima forecast ###
aa.fit <- auto.arima(monthly.ts, seasonal = TRUE)
```

```
aa.fit
tsdisplay(residuals(aa.fit), lag.max = 50)

### Computing ticket averages ###
tickets.averages <- tickets %>%
  group_by(location, month.index) %>%
  summarize(
    count = n()
  ) %>%
  group_by(month.index) %>%
  summarize(
    mean = mean(count)
  )

### Plotting ticket averages ###
ggplot(data = tickets.averages) +
  geom_line(mapping = aes(x = month.index, y = mean)
     ) +
  labs(x = "Month␣Index", y = "Tickets") +
  theme_bw() +
  theme(axis.text = element_text(size = 12), axis.
     title = element_text(size = 14))

### Converting mean data to time series object ###
mean.ts <- ts(tickets.averages$mean, start = c(2007,
    1), frequency = 12)

### Differencing ###

### Differencing ###
meandiff.ts <-diff(mean.ts, differences = 1)
                    # First difference
meandiff2.ts <- diff(mean.ts, lag = 12, differences
   = 1)         # First seasonal difference
meandiff3.ts <- diff(diff(mean.ts, lag = 12))
                    # First difference plus first
   seasonal difference
```

```
### Plotting original and differenced data ###

q1 <- autoplot(as.zoo(mean.ts), geom = "line") +
   theme_bw() + theme(axis.title = element_text(size
    = 12), axis.title.x = element_blank(), axis.text
    = element_blank()) + labs(y = "No differencing",
    title = "(1)")
q2 <- autoplot(as.zoo(meandiff.ts), geom = "line") +
    theme_bw() + theme(axis.title = element_text(
   size = 12), axis.title.x = element_blank(), axis.
   text = element_blank()) + labs(y = "1st diff",
   title = "(2)")
q3 <- autoplot(as.zoo(meandiff2.ts), geom = "line")
   + theme_bw() + theme(axis.title = element_text(
   size = 12), axis.title.x = element_blank(), axis.
   text = element_blank()) + labs(y = "1st seasonal
   diff", title = "(3)")
q4 <- autoplot(as.zoo(meandiff3.ts), geom = "line")
   + theme_bw() + theme(axis.title = element_text(
   size = 10), axis.title.x = element_blank(), axis.
   text = element_blank()) + labs(y = "1st diff and
   1st seasonal diff", title = "(4)")

multiplot(q1, q3, q2, q4, cols = 2)

### ACF and PACF of first difference data ###

par(mfrow = c(1, 2))
Acf(meandiff.ts, lag.max = 70, main = "")
Pacf(meandiff.ts, lag.max = 70, main = "")
par(mfrow = c(1, 1))

### Fitting (0,1,1) x (2,0,0)[12] model ###
meanmanual.fit <- Arima(mean.ts, order = c(0, 1, 0),
    seasonal = c(2, 0, 0), method = "ML")
meanmanual.fit
tsdisplay(residuals(meanmanual.fit), lag.max = 50,
   main = "(0, 1, 1)x(2, 0, 0)[12] Model Residuals")
```

```
### Preliminary forecast of
plot(forecast(meanmanual.fit, h = 36))

### Generating model via auto.arima ###
### Auto.arima forecast ###
meanaa.fit <- auto.arima(mean.ts, seasonal = TRUE)
meanaa.fit
tsdisplay(residuals(meanaa.fit), lag.max = 50)
```

References

Bisgaard, Soren and Murat Kulahci (2011). *Time Series Analysis and Forecasting by Example*. Wiley.

Byrne, John (2010). "Ald. Burke: Red light cameras a 'money machine'". In: *Chicago Tribune*. Mar. 10, 2010.

Cryer, Jonathan and Kung-Sik Chan (2008). *Time Series Analysis With Applications in R*. Springer.

Erke, Alena (2009). "Red light for red-light cameras?: A meta-analysis of the effects of red-light cameras on crashes". In: *Accident Analysis & Prevention* 41.5, pp. 897–905.

Ruthhart, Bill (2015). "Most Chicagoans want to eliminate or cut back Emanuel red light program". In: *Chicago Tribune*. Jan 31, 2015.