UNIVERSITY OF WISCONSIN-LA CROSSE

Graduate Studies

APPLICATIONS OF TIME SERIES ANALYSIS FOR FORECASTING FUEL SALES AND CHANGE POINT DETECTION

A Chapter Style Thesis Submitted in Partial Fulfillment of Requirements for the Degree of Master of Science

Calvin J. Corey

College of Science and Health
Applied Statistics

August, 2020

APPLICATIONS OF TIME SERIES ANALYSIS FOR FORECASTING FUEL SALES AND CHANGE POINT DETECTION

By Calvin Corey

We recommend acceptance of this thesis in partial fulfillment of candidate's requirements for the degree of Master of Science in Applied Statistics.

The candidate has completed the oral defense of the thesis.

David Reineke, Ph.D. Date
Thesis Committee Chairperson
On behalf of the Committee:

Song Chen, Ph.D.
Thesis Committee Member

Sherwin Toribio, Ph.D.
Thesis Committee Member

Thesis accepted

Meredith Thomsen, Ph.D. Date
Director of Graduate Studies

ABSTRACT

Corey, C.J. Applications of time series analysis for forecasting fuel sales and change point detection. MS in Applied Statistics, August 2020, 61pp. (D. Reineke)

This paper shows how time series models can be used to forecast motor vehicle fuel sales, and how to use those models to detect changes in the time series signals. The model used is a least squared regression model that considers seasonal trends, serial autocorrelation, day of week, holidays, and days since open. With these covariates, the models proved to be highly predictive with forecasts on gasoline and diesel fuel, as the maximum cumulative accuracy was at most 2.74% for gasoline and 1.33% for diesel fuel. It can also be shown that mean absolute error rates can be represented by a gamma distribution, and this can be used to detect changes in the time series signal.

ACKNOWLEDGEMENTS

This page is dedicated to all of the people that have helped me through this process. I am extremely happy to have had the opportunity to pursue this project, and to begin to see the applications that statistical theory can have in the real world.

Firstly, I would like to the members of my thesis committee, Dr. David Reineke, Dr. Sherwin Toribio, and Dr. Song Chen. Without their instruction at both the classroom and individual level this project would not have been possible. I would also like to thank Dr. Reineke especially as my Thesis Chair and helping with all of the finer details of the thesis.

I would like to thank Dr. Melissa Bingham and Dr. Robert Allen, heads of the Statistics and Mathematics departments here at UW-La Crosse for all of the advice and support along the way.

A big thank you goes out the Sean Callister, and the rest of the business analyst team at Kwik Trip. It was their industry specific knowledge that helped drive the focus of the study, and their guidance has made the output of the project have real world impact that I am extremely happy to have been a part of.

A final thank you to my friends and family, who have been supportive and encouraging throughout this entire process. Thank you all so much!

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURES PAGE

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

Businesses are constantly trying new strategies in order to maximize the revenue and profitability of their business. These strategies can encompass a wide range of departments, from manufacturing to marketing, and it is critical that an organization is able to measure the performance of these changes in strategy in order to make sure the changes are beneficial. Therefore, various statistical tools have been developed to measure this performance. External market forces can also create changes in the performance of a business, and these are beyond the control of the firm itself, such as new regulations, world events, and competing businesses.

The traditional statistical approach to this sort of analysis is a standard experimental design. A subset of the business continues to use the original strategy while another subset uses a new strategy. The performance of the subsets is then analyzed with a procedure such as analysis of variance or linear regression, and conclusions can be drawn from those results. However, this might not be possible given other factors within the business. For example, retail locations separated by large geographical distances may not react the same to similar changes in strategy, or seasonal fluctuations in the market may make direct comparison impossible.

When these sorts of direct comparisons are impossible, alternative methods are used. Modern business data warehousing software allows for precise records of transactions to be maintained in real time. Since timestamps for transactions are kept, time series methods can be used in order to determine how the organization is expected to perform and draw comparisons to actual performance after a change has taken place.

Kwik Trip is a convenience store company based in La Crosse, WI, with over 600 locations across Wisconsin, Minnesota, and Iowa. Fuel sales are always in a state of constant fluctuation. Factors ranging from weather to international politics are assumed to impact the demand for oil-based products. Therefore, accurate analysis for changes in fuel sales strategies are key in order to minimize the impact of these many external factors and maximize sales and profitability.

Purpose of the Study

The primary purpose of this study is to create a system of time series models that can be used to accurately forecast fuel sales volumes at individual Kwik Trip store locations. The secondary purpose is to detect changes in the sales performance of the retail locations, and select specific locations for further analysis.

Need for the Study

As corporations like Kwik Trip build and maintain large amounts of historical data, there is an ever-increasing desire to leverage this data in new and meaningful ways. At Kwik Trip, like many companies, historical data is only used in a system of summarization, looking at various key points of interest like year over year change in sales volume, total sales volume, and companywide averages, among others. Using new methods that are predictive in nature allow for an organization to be proactive rather than reactive, as is the case with summarization type reporting.

Analysis of time series models are largely limited to either forecasting techniques or post-hoc change point detection techniques. In a study measuring the impact on oil prices after major global geopolitical events (Jeng-Bau & Wei, 2019), the authors were attempting to analyze the price of oil after major events in the early 2000's. In these types of analysis, there emphasis is to measure the impact after the events have taken place. On the other hand, time series models that are employed in other economic disciplines and the sciences are more concerned with forecasted values. For example, in a study of electric vehicle sales in China (Zhong et. al, 2017), the authors were looking at the total volume of vehicle sales in the country, with a focus on prediction of total sales.

The goal is to build a system of forecasting models that considers data in real time to predict how these locations will perform in terms of sales volume. The secondary goal is to use these analyses determine to when significant changes to the performance of a location have occurred.

CHAPTER II

REVIEW OF LITERATURE

Introduction

This section is a review of time series analysis and the methods therein, such as regression, auto regression, and cosine trending, as well as examples of their application.

Time Series Basics

A time series signal is data obtained sequentially over time (Cryer, Kung-Sik, 2010, p. 1). A large number of fields utilize time series data, from the daily prices of stocks on the New York Stock Exchange, monthly number of patients admitted to a hospital, or annual population estimates of an animal species are all examples of time series signals. Given the very broad number of applications for time series signals, a large number of techniques have been developed to analyze this sort of data.

Before any modeling can be done, it must be determined beforehand whether the models will be used for a descriptive analysis or predictive forecasting. An analytical approach is meant to distill out historic information about the signal and determine things such as trends and seasonal fluctuation. In contrast, a predictive forecast is only to give the best prediction about events that have yet to happen. Therefore, interpretability of these models are not emphasized in this context, and a large number of methods may not be interpretable at all (Shmueli, et. al, 2018, p. 389).

Elements of a time series signal structure need to then be assumed prior to analysis as to whether they are a deterministic or a stochastic process. A stochastic process is simply observations of random variables collected over time (Cryer, Kung-Sik, 2010, p.11). In contrast, a process that has a particular, predictable structure is a deterministic process (Cryer, Kung-Sik, 2010, p.27). Specific domain knowledge is critical in order to determine what elements are stochastic and which are deterministic, as stochastic processes can manifest themselves in a way that makes them appear to be deterministic (Cryer, Kung-Sik, 2010, p.31).

Analysis Techniques

The goal of any time series analysis technique is to summarize trends in a time series signal and give a description of what has happened within the signal. They can also be used for change point detection, a process by which it is determined if the structure of a time series signal has changed after an event has taken place.

The most basic form of trend is the constant mean. This model is simply that values move stochastically around a mean value μ, and the values vary with the error terms e following a distribution with mean zero (Cryer, Kung-Sik, 2010, p. 28). Therefore, we can describe the value for Y at time t as:

$$Y_t = \mu + e_t \tag{2.1}$$

If circumstances allow, this model can be extended to allow for periodicity, where the mean value at time t is equal to the value at $t - n$. For example, if the data is collected monthly, then the following model may be appropriate:

$$Y_t = \mu_{t-12} + e_t \tag{2.2}$$

where the mean value in the current month is equal to the mean of the month 12 months prior. This model is occasionally called seasonal means (Cryer, Kung-Sik, 2010, p. 32).

If it is assumed that values at time t are similar to values nearby, $t \pm 1, t \pm 2$, etc., and that values are assumed follow a sinusoidal trend, then cosine trending can be employed to describe the signal. Cosine trending can be described with the following formula:

$$(2.3) \qquad Y_t = \mu + P\cos\left(\frac{2\pi t}{L} + \Phi\right) + e_t = \mu + P_1\cos\left(\frac{2\pi t}{L}\right) + P_2\sin\left(\frac{2\pi t}{L}\right) + e_t$$

Where L is the length of the period and Φ is the phase shift. Alternatively, least squares regression can be used to solve for Φ with to separate coefficients P_1 and P_2 (Cryer, Kung-Sik, 2010, p. 34). All of these models can be combined to give more accurate descriptions of data if this structure is assumed to be true.

In a large number of cases, values for Y_t are not consistent around some value μ. This type of relationship is sometimes considered a growth or declining trend. These sorts of trends can be analyzed using a number of different linear models, be it linear, or exponential change (Shmueli, et. al, 2018, p. 401).

The most basic of these linear models is simple linear regression. This model has a number of advantages in that it is simple to fit and to describe. The model is as follows (Shmueli, et. al, 2018, p. 403):

$$(2.4) \qquad Y_t = \beta_0 + \beta_1 t + e_t$$

where β_0 can be described as the value at Y_0 and β_1 is the change Y over 1-unit change in time. However, due to a large number of data sets experiencing growth that is not linear, a common extension of this model is the exponential growth model. This model uses the natural log of the values, rather than the values directly (Shmueli, et. al, 2018, p. 405).

$$\ln(Y_t) = \beta_0 + \beta_1 t + e_t \tag{2.5}$$

While linear trends have the ability to describe events that have already occurred, there are several limitations on their practicality for forecasting (Cryer, Kung-Sik, 2010, p. 30).

Another way that changes in Y_t over time can be tracked is serial autocorrelation (Sheather, 2009, p.305). Serial autocorrelation, also called autoregression, relates the value Y_t a series of "lag" terms for $t-1, t-2,$ etc. (Shmueli, et. al, 2018, p. 416).

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} \dots + e_t \tag{2.6}$$

The number of autocorrelation terms can be found using the autocorrelation function, that measures the correlation at time t with the any number of previous lags (Sheather, 2009, p.308).

Applications of Time Series Analysis: Change Point Detection

Change point detection is often a critical area of application for the analysis of time series signals. Put simply, change point detection is when the fundamental structure of a time series signal changes (Basseville, 2019). In a business setting, events such as world political events (Lin & Tsai, 2019) or marketing campaigns (Zhang et. al, 2019) can change the way time series signals manifest over time.

In a study conducted by Lin and Tsai, they were looking to see if the prices of fuel sales were strictly driven by the market fundamentals of supply and demand, or if major political events on the world stage had an impact on the price of crude oil. For a ten-year period from 2007 to 2017, they looked at major events such as the economic collapse of

2008. Even with a very simple model, it was very straight forward to demonstrate the impacts of these world events on the price of oil and measure of volatility indexes.

In the study by Zhang et. al, the effect that online marketing campaigns have on purchases made to an online marketplace was tested. The goal was to maximize the return on investment for online marketing spending. They were able to accurately quantify the impact that different marketing schemes had and used that to determine the most profitable campaign methods.

Forecasting Methods

Forecasting techniques are similar to analysis techniques in that the majority of the methods that can be used to describe the data can also be carried forward to infer future values of Y_t. However, there are a number of considerations that make them slightly different.

To make predictions on a mean model, the calculation is straight forward in that all future expected values are simply the mean value:

$$E(Y_k) = \mu \tag{2.7}$$

where the μ can be estimated as the average of all previous observations (Cryer, Kung-Sik, 2010, p. 28).

$$\hat{\mu} = \bar{Y} \tag{2.8}$$

By extension if there is periodicity assumed in the model, a set of estimates for μ are generated for all of the different means from 1 to n. Again, if the case is monthly data, the model is built with 12 different means

$$(2.9) \qquad \hat{\mu} = \begin{cases} \widehat{\mu_1} = \left(\frac{1}{k}\right)(Y_1 + Y_{13} + \cdots) \\ \widehat{\mu_2} = \left(\frac{1}{k}\right)(Y_2 + Y_{14} + \cdots) \\ \vdots \\ \widehat{\mu_{12}} = \left(\frac{1}{k}\right)(Y_{12} + Y_{24} + \cdots) \end{cases}$$

where k is the number of observations in the subset of data that corresponds to that particular month. (Cryer, Kung-Sik, 2010, p. 32) While these models may not be very powerful predictors, they do generally serve as a benchmark for other model's predictive accuracy, and are sometimes called a naïve and seasonal naïve forecasts, respectively. (Shmueli, et. al, 2018, p. 396)

When a more refined model is needed and the data is suspected to follow a consistent seasonal trend, a cosine curve can be employed. (Cryer, Kung-Sik, 2010, p. 34)

$$(2.10) \qquad Y_t = \mu + P\cos\left(\frac{2\pi t}{L} + \Phi\right) = \mu + P_1\cos\left(\frac{2\pi t}{L}\right) + P_2\sin\left(\frac{2\pi t}{L}\right)$$

The linear combination of sine and cosine function allows for least squares regression to be used in order to determine their coefficients, as the value Φ is more difficult to determine via nonlinear techniques. Once the coefficients P_1 and P_2 are estimated, the forecast for time $t + k$ can be determined by:

$$\hat{Y}_{t+k} = \mu + P_1 \cos\left(\frac{2\pi(t+k)}{L}\right) + P_2 \sin\left(\frac{2\pi(t+k)}{L}\right) \tag{2.11}$$

Autoregression allows for analysis of local trends because it is assumed that earlier observations in time are related to later observations. (Sheather, 2009, p.305) Therefore, the coefficients of the autoregressive coefficients can be estimated using standard least squares regression, but a special procedure is required to make forecasts. (Sheather, 2009, p.310) Firstly, the standard regression model is fit

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} \dots + e_t \tag{2.12}$$

To make a forecast for $t + 1$, the following estimate is made

$$\hat{Y}_{t+1} = \beta_0 + \beta_1 Y_t + \beta_2 Y_{t-1} \dots \tag{2.13}$$

Then, to forecast $t + 2$, the estimate for $t + 1$ is used in the right-hand side of the autoregressive model

$$\hat{Y}_{t+2} = \beta_0 + \beta_1 \hat{Y}_{t+1} + \beta_2 Y_t \dots \tag{2.14}$$

This process is repeated until the desired number of forecasts are reached, such that

$$\hat{Y}_{t+k} = \beta_0 + \beta_1 \hat{Y}_{t+k-1} + \beta_2 \hat{Y}_{t+k-2} \dots \tag{2.15}$$

where k is the number of intervals needed to be forecast.

All of these methods can be combined in a mixed effects model. Mixed models are the combination of fixed effects as well as time series effects. Since there is an assumption of serial autocorrelation, these models do not behave the same as standard regression models since the condition of independent observations is violated, as is the case with all time series models. Therefore, the standard tests of model fit do not necessarily apply (Sheather, 2009, p.310).

A standard approach to assessing model fit in a machine learning setting is through the use of a training and a validation set. In a general sense, a training set is a subset of all of the data that is used for the purposes of model building. This set contains the majority of the data, and the remaining data are placed in the validation set. The validation set is then used to assess predictions made by the model generated by the training set. Since the actual outcome is known, the model predictions can be compared to the actual outcomes, and the various error metrics that will be discussed can be used to gauge a model's ability to make predictions (Shmueli, et. al, 2018, p. 36).

In most cases, observations for the training and testing sets are selected at random. By selecting at random, this prevents bias in the sets and provides a clear sense of a models' ability to make predictions (Shmueli, et. al, 2018, p. 36). However, one of the model requirements for a majority of time series techniques is that time lags must be equally spaced (Shmueli, et. al, 2018, p. 395). Therefore, selecting at random creates unequal gaps in the data and violates this assumption. Alternatively, the training set and testing set is defined by the observations that come before a point in time, and the validation set becomes all observations after this point (Shmueli, et. al, 2018, p. 395).

K-folds cross validation is another, more robust method for analyzing a model's predictive accuracy. It is similar to the standard testing and validation procedure described above, except instead of one large training set and one small validation set, the data is split into k random subsets. One of these subsets are selected to become the validation set, and the remaining $k-1$ sets are combined to create the training set. This procedure is repeated until all k sets have been used as validation sets, and all of the predictive accuracy is compared (Shmueli, et. al, 2018, p. 37).

As with a standard training validation procedure, a random division of data is not possible with a time series signal. Instead the data is divided at time t, and the remaining data is split into k intervals of length l. Therefore, the test sets are from t to $(t+l)$, $(t+l)$ to $(t+2l)$, … , $(t+(k-1)l)$ to $(t+kl)$. This gives a similar level of robustness as k folds cross validation.

There are a number of different metrics for measuring a models' ability to make predictions. Each metric has its advantages and disadvantages, but the two of the most common are mean percent error and mean absolute percent error. Mean percent error is represented by the following equation:

$$MPE = \frac{100}{n}\sum_{i=1}^{n}\frac{e_i}{y_i} \tag{2.16}$$

The advantage of this metric is that it can show overall bias in model predictions. For example, large values in mean percent error indicate that the model consistently under predicts the actual values. However, if errors fluctuate between positive and negative values, then errors can be artificially deflated. For example, if we have two observations with predictive errors (y_i, e_i) and (y_{i+1}, e_{i+1}), where $\frac{e_i}{y_i} = -\frac{e_{i+1}}{y_{i+1}}$ and $\frac{e_i}{y_i}$ is very large, the net error will sum to zero even though the model was not very useful for making predictions.

Mean absolute percent error is very similar to mean percent error in that:

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{e_i}{y_i}\right| \tag{2.17}$$

An advantage in mean absolute percent error is that it standardizes all errors to be positive, meaning it allows for an accurate representation of the models' ability to make individual predictions. However, it removes the ability to see any bias in the predictions. Therefore, it is necessary to look at both error metrics in aggregate to get an idea of model performance (Shmueli, et. al, 2018, p. 119).

Applications of Forecasting Techniques

In general, applications of forecasting techniques are used to assist organizations in planning and to anticipate what the future brings. One example is a study of the sales of various types of electric vehicles in China (Zhang et. al., 2019). The researchers are trying to predict the future sales of electric vehicles as the Chinese government attempts to reduce greenhouse gas emissions. In another study, the future demand of electricity in Europe is studied (Sun et. al. 2019). As more and more demand on the electrical power grid is needed, researchers look to predict this demand in order to grow electrical capacity most efficiently.

China is one of the largest producers of greenhouse gas emotions, given its immense size. China is responsible for 27.1% of global greenhouse gas emissions and is dependent on oil imports for 59.6% of the nation's needs. Therefore, the Chinese government believes it is important to reduce the number of fuel driven vehicles and replace them with electric powered models.

In the study by Zhang et. al., they looked to the sale of electric powered vehicles as a means to track the transition from fuel oils to electric power. There are two types of electrical vehicles studied, completely electric and hybrid vehicles, and their system of autoregression as a means of prediction future demand two years into the future with reasonable accuracy of 29.1% and 16.1% MAPE (Zhang et. al., 2019).

In the example by Sun et. al., they are looking at the demand for electricity in Europe. As more and more technology is used in every day life, this places an ever increasing demand on the electrical power grid. In order for European nations to plan for future demand of electricity, it is necessary to make predictions and estimations. In this example, they us a system of autoregression and monthly smoothers, and they were able to predict the demand across a 12-month period to an average mean absolute percent error of 4.00% (Sun et. al. 2019).

CHAPTER III

METHODS AND PROCEDURES

Introduction

This section outlines the overall statistical model that will be used to predict the total volumes of fuel sales. The secondary purpose is to use these models to determine if the structure of the time series signal has changed. These models use the aggregate accuracy of multiple models in a standardized fashion in order to build accurate forecast prediction intervals. As the structure of the organization and marketplace changes these forecasts will be used as a measure of the effect of these changes, based on the confidence intervals discussed previously.

Modeling Procedure

In order to make accurate analyses on a very fine time scale, there are a number of different factors to consider. Larger macro trends must be balanced with the micro trends that create large amounts of noise within the time series signal. In the particular case of daily sales, represented as y_t, trends that manifest over the course of the fiscal year must be balanced with the trends on a day to day basis. Therefore, different components were included in the model to account for the larger seasonal trends and the finer day to day trends.

$$(3.1)\quad y_t = C + \mathrm{P}_1 \cos\left(\frac{2\pi t}{L}\right) + \mathrm{P}_2 \sin\left(\frac{2\pi t}{L}\right) + \beta_T + \sum_{i=1}^{n} \alpha_i Y_{t-i} + \sum_{i=1}^{n} \alpha_i \beta_{T-i} Y_{t-i} + \gamma_T + \lambda_T$$

The model above can be broken into three major components. The first part controls for the seasonal trend:

$$(3.2)\qquad \mathrm{P}_1 \cos\left(\frac{2\pi t}{L}\right) + \mathrm{P}_2 \sin\left(\frac{2\pi t}{L}\right)$$

The coefficients P_1 and P_2 control the amplitude as well the phase shift of the seasonal trend through signal interference. The value L is the number of points within a complete cycle. Typically, a year is used and therefore $L = 365$. However, Kwik Trip operates on a 13-period fiscal calendar and in this case $L = 13$. This is to done reduce model complexity by combining the cosine trend and seasonal naïve trend into a single value.

The second set of components accounts for local daily trends. The factor β_T is the control for day of week. In practical terms, this value is actually the difference between a baseline and the factor that is being analyzed. The baseline value is a factor that gets absorbed by the intercept term C. In the context of a large number of coding languages, the factor that gets absorbed by the intercept term is the factor level that comes first alphabetically. In this case, the B_T term is actually comparing the day of the week that is being predicted to Friday, and when the day is Friday, $B_T = 0$.

The following part controls the autoregression portion:

$$\sum_{i=1}^{n} \alpha_i Y_{t-i} + \sum_{i=1}^{n} \alpha_i \beta_{T-i} Y_{t-i} \tag{3.3}$$

There are 2 components within the autoregression portion of the model. The first component is a standard autoregression model, with n lagging terms. The second portion is the interaction between the lagging terms and the day of the week. For example, this allows for the differentiation if Monday was yesterday or four days ago. For this reason, n is chosen to be seven for the model to work most efficiently.

The final section deals with local known events. When a new store is opened, there is a unique yet consistent pattern in the way sales behave. Typically, this is manifested as a short-term boost in sales, denoted by γ_T. This is caused by a combination of local public interest and a strong marketing campaign. During this initial window, the sales bubble generally follows the kernel of a gamma distribution such that:

$$\gamma_T = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} t^{\alpha-1} e^{\frac{t}{\beta}} \tag{3.4}$$

where t is the number of days since the store opened. The alpha and beta terms are best estimated to be around 2 and 30 respectively. The last term λ_T is a categorical variable correcting for deviations from holidays. The holidays considered are a subset of federally recognized holidays: New Year's Day, Memorial Day, Independence Day, Labor Day, Thanksgiving Day, and Christmas Day. The Birthday of Martin Luther King Jr., George Washington's Birthday, Columbus Day, and Veterans Days are not considered.

Confidence Intervals

The fundamental problem of comparing multiple retail locations is that they can be vastly different in terms of sales volume. One location, for example, located on a major highway, may attract a large number of sales from both commercial and retail customers. Whereas a location located in a small town may largely draw from a small, local consumer base. Therefore, a standardized system for gauging model performance across a wide range of retail locations is critical. Fortunately, mean absolute percent error allows for a standardized measure for gauging model performance, and the error rates across multiple models is expected to follow a gamma distribution.

A set of time series data containing $t + k$ observations, and a split point at time t is selected. This data is split into two sections, the training set which contains the data from observation 1 to observation $t,$ and the testing set which contains the observations from $t + 1$ to $t + k$. The training set is then used to build a model as per the procedure as described in the previous section. The prediction is then compared to the actual values from $t + 1$ to $t + k$ and reported in terms of mean absolute percent error. This complete procedure is repeated across all locations. The mean and standard deviation of the mean absolute percent error is recorded, which are used to estimate the shape and scale parameters of the gamma distribution. Since the values for mean absolute percent error lose their directionality, if a $100(1 - \alpha)\%$ confidence interval is desired on the original forecast, 2α is used as the upper-tail probability in the gamma distribution. This resulting percentage is used to create the following confidence interval:

$$\hat{Y}_{t+k} \pm \rho_{2\alpha}\hat{Y}_{t+k} \tag{3.5}$$

Where ρ follows and gamma distribution with a shape of $(\bar{y}_e/s_e)^2$ and a rate of $\bar{y}_e/(s_e)^2$.

Cumulative Model Performance

Another point of consideration is cumulative performance. In a number of cases, the actual daily performance is not a practical level of detail to analyze data in order to make sound strategy decisions. Therefore, cumulative performance over longer periods of time, such as weekly or monthly, are more useful in the broad decision-making process. Models will be based on their predictive daily values, as well as their ability to make cumulative forecasts.

Change Point Detection

Once the forecasts and confidence intervals are built, data is continuously monitored as it becomes available. A significant change is determined when the signal leaves the confidence interval as described in the previous section.

Data Available

The data is a set of total daily fuel sales from 288 stores, of which 204 sell diesel fuel. The data ranges from September 30, 2016 to September 26, 2019. Other data included is the number of days since open, day of week, as well as the fiscal period. A fiscal period is defined as 28 days, starting on Friday and ending on a Thursday. The first fiscal period is defined as the first 28 days of the fiscal year, and so on up to 13 periods, meaning a fiscal calendar contains 364 days total.

Data Masking

Even though Kwik Trip is generous enough to supply data, they still desire that their proprietary information is protected from competitors. Therefore, steps have been taken to hide the true values of the data without undermining the fundamental structure. Firstly, the data supplied is a random subset of 288 stores that were open before the start of fiscal 2017 (September 28, 2016), of which 204 stores sell diesel fuel. The stores are then assigned a random 5-digit number to act as its identifier. Finally, two random numbers are generated between 0.5 to 2. The first number is multiplied by gasoline gallons and the second number is multiplied by the diesel gallons if the store happens to sell that product.

Initial testing has shown that these constants do not fundamentally change the predicative accuracy of the models given the measures of accuracy described previously.

Simulation Procedure

Starting on September 30, 2018, forecasts will be cast into the future at intervals of 28, 56, 112, 224, and 336 days, and the cumulative performance is measured against actual values. These forecasts are conducted on both the 288 gasoline stores as well as the

204 diesel sites. Then starting at September 30, 2018, December 30, 2018, and May 31, 2019, daily models will be conducted from 2 to 91 days. These models will be used to show daily as well as cumulative model performance and measure stability over time. Finally, a selection of stores will be showcased to give a better understanding of a models ability to make daily predictions.

CHAPTER IV

RESULTS AND DISCUSSION

Results

Tables 1 and 2 represent the long-term forecasts from 28 to 336 days. These days were selected as they were all multiples of 28 and would contain the same number weekdays and weekend days. These tables represent the model's ability to make predictions overall. These model's had percent errors of less than 3% in gasoline and 1.5% in terms of diesel. Again, there are 288 locations that sell gasoline and 204 that sell diesel fuel.

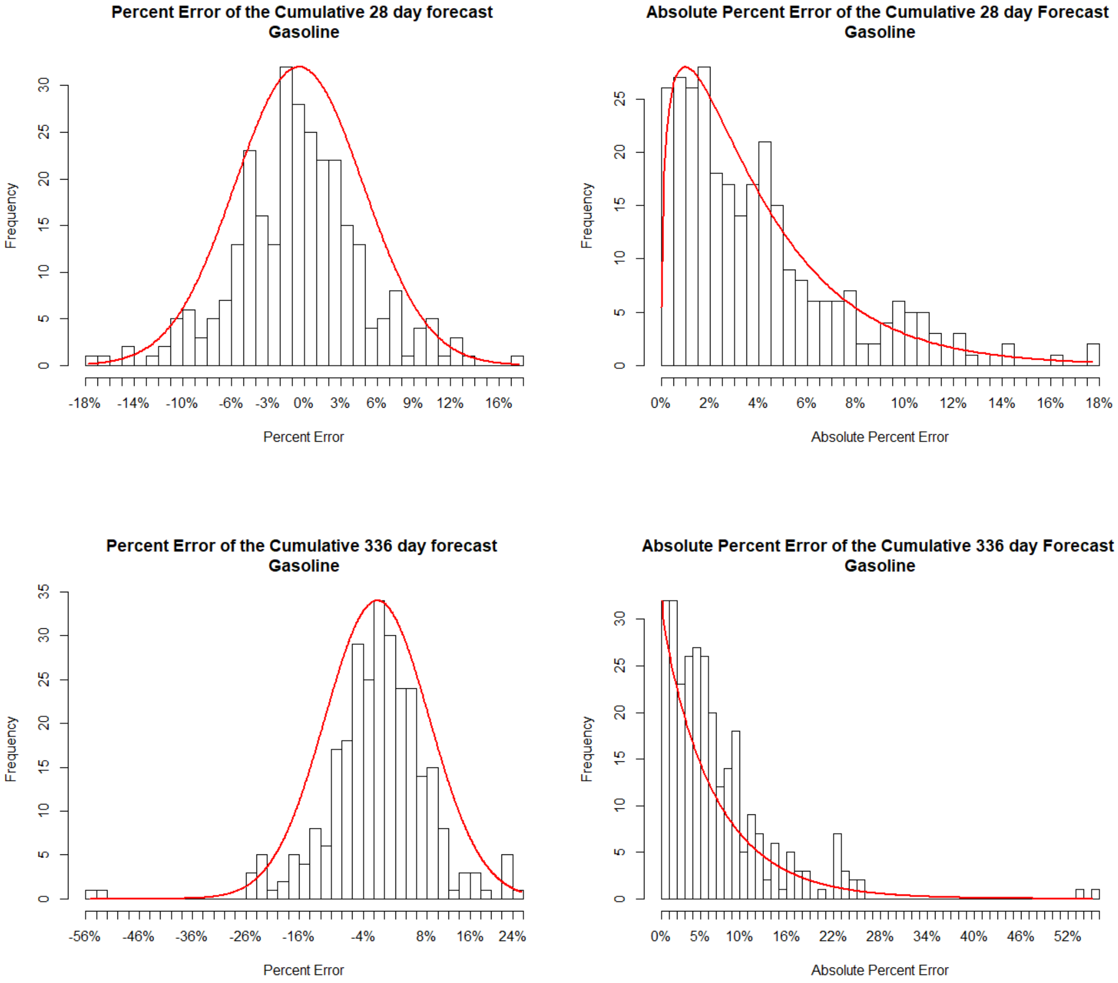
Table 1: Cumulative Gasoline Forecasts starting from 9/30/2018

Days Ahead	Forecast Total Gallons	Actual Total Gallons	Percent Error
28	37,816,136	38,094,669	-0.73%
56	74,285,110	76,060,619	-2.33%
112	144,339,791	148,412,206	-2.74%
224	288,606,102	289,967,470	-0.47%
336	441,852,504	430,849,581	2.55%

Table 2: Cumulative Diesel Forecasts starting from 9/30/2018

Days Ahead	Forecast Total Gallons	Actual Total Gallons	Percent Error
28	9,021,046	9,006,224	0.16%
56	17,502,170	17,272,598	1.33%
112	33,734,611	33,627,068	0.32%
224	66,627,241	66,514,683	-0.17%
336	100,802,926	99,731,461	1.07%

Histograms are used to represent the model’s ability to make predictions on an individual store basis. These histograms show the overall distribution of model predictions in terms of mean percent error and mean absolute percent error. Mean percent error is used to show bias in the model predictions, and mean absolute percent error is used to show the ability to make predictions on an individual basis. Shown will be 28 days and 336 days shown here in Figure 1 and 2, the remaining plots for 56, 112, and 224 days will be available in Appendix A.



Percent Error of the Cumulative 28 day forecast Gasoline

Absolute Percent Error of the Cumulative 28 day Forecast Gasoline

Percent Error of the Cumulative 336 day forecast Gasoline

Absolute Percent Error of the Cumulative 336 day Forecast Gasoline

Figure 1: Plots showing model accuracy for forecasted gasoline sales

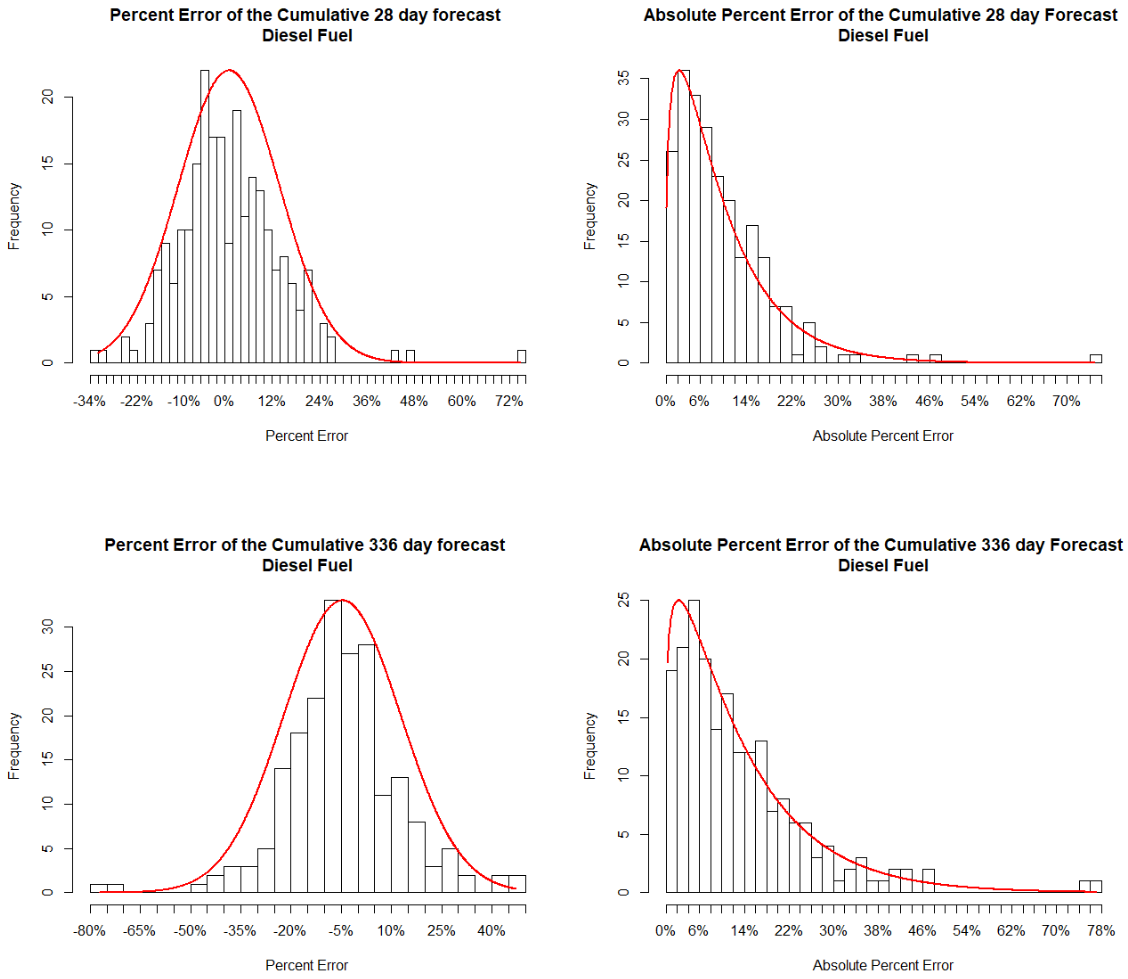


Figure 2: Plots showing model accuracy for forecasted diesel sale

In order to verify that the following distributions are in fact appropriate, the Shapiro-Wilk goodness of fit test is conducted. This is done using the gofTest function in the EnvStats package, with p-values being adjusted with the FDR correction using p.adjust function in base R. Tests were run using R version 3.5.0. These will be done to confirm that the mean percent error and mean absolute percent error follow a normal and gamma distribution, respectively.

Table 3: Testing the appropriateness of the Normal and Gamma distribution for Gasoline model error rates. Models use the Shapiro-Wilks Test of Model fit W statistic. The test statistic for MPE is testing for normality, whereas MAPE is testing for a gamma distribution. The p-values are adjusted with the FDR correction.

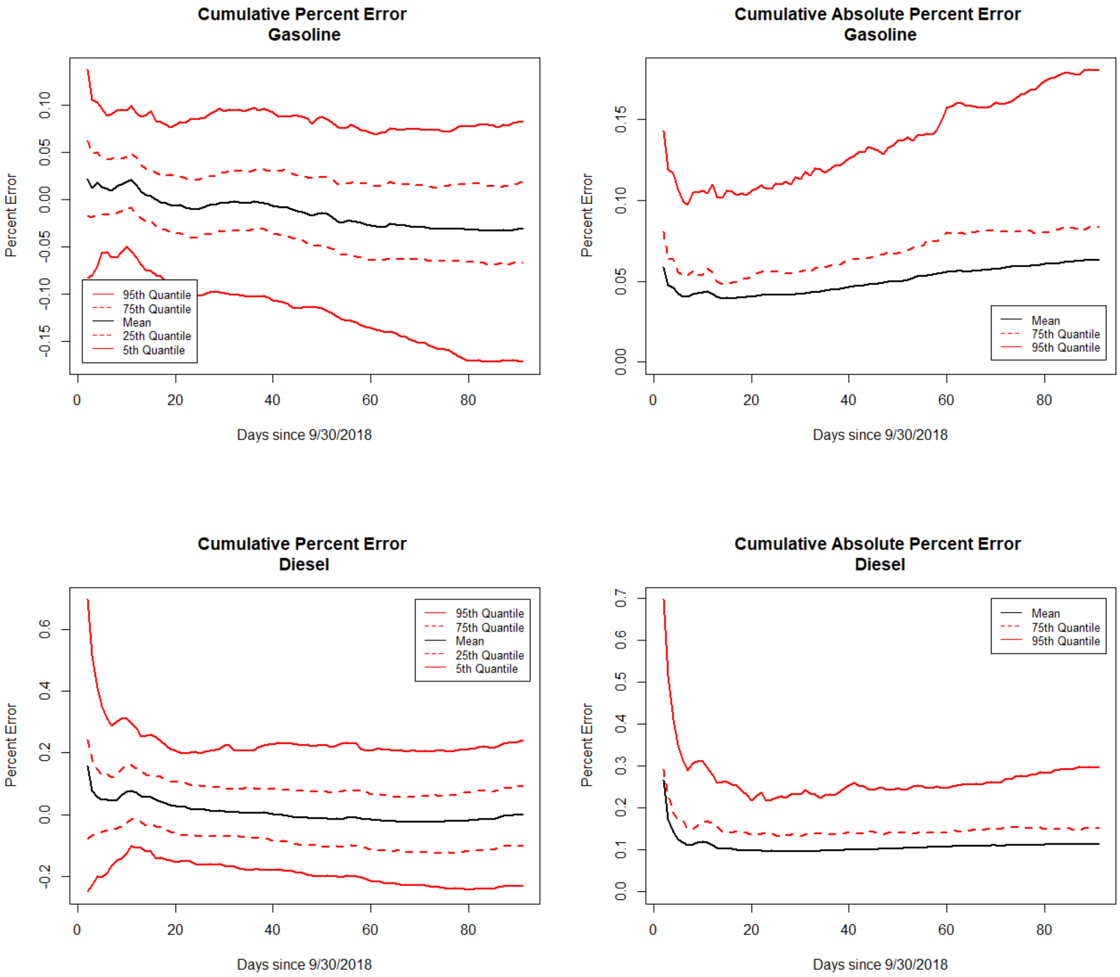
Days Ahead	MPE (W)	p-value	MAPE (W)	p-value
28	0.989	0.026	0.993	0.239
56	0.961	<0.001	0.995	0.549
112	0.969	<0.001	0.997	0.918
224	0.936	<0.001	0.993	0.160
336	0.929	<0.001	0.990	0.047

Table 4: Testing the appropriateness of the Normal and Gamma distribution for Diesel Fuel model error rates. Models use the Shapiro-Wilks Test of Model fit W statistic. The test statistic for MPE is testing for normality, whereas MAPE is testing for a gamma
distribution. The p-values are adjusted with the FDR correction.

Days Ahead	MPE (W)	p-value	MAPE (W)	p-value
28	0.939	<0.001	0.988	0.075
56	0.970	<0.001	0.992	0.344
112	0.979	0.005	0.987	0.058
224	0.961	<0.001	0.997	0.978
336	0.973	0.001	0.993	0.528

The following graphs show each model’s ability to make predictions over time. These plots are cumulative performance, similar to Tables 1 and 2, yet are daily forecasts for up to 91 days respectively. Like the previous models, the start dates for the models are 9/30/2018. Alternative 91-day forecast plots are available starting on 12/30/2018 and

3/31/2019 in Appendix B. Plots that show the model's ability to make predictions on a daily basis are also shown in Figure 4. These are similar in structure to the previous plots.



Cumulative Percent Error Gasoline

Cumulative Absolute Percent Error Gasoline

Cumulative Percent Error Diesel

Cumulative Absolute Percent Error Diesel

Figure 3: Daily cumulative forecast errors over time.

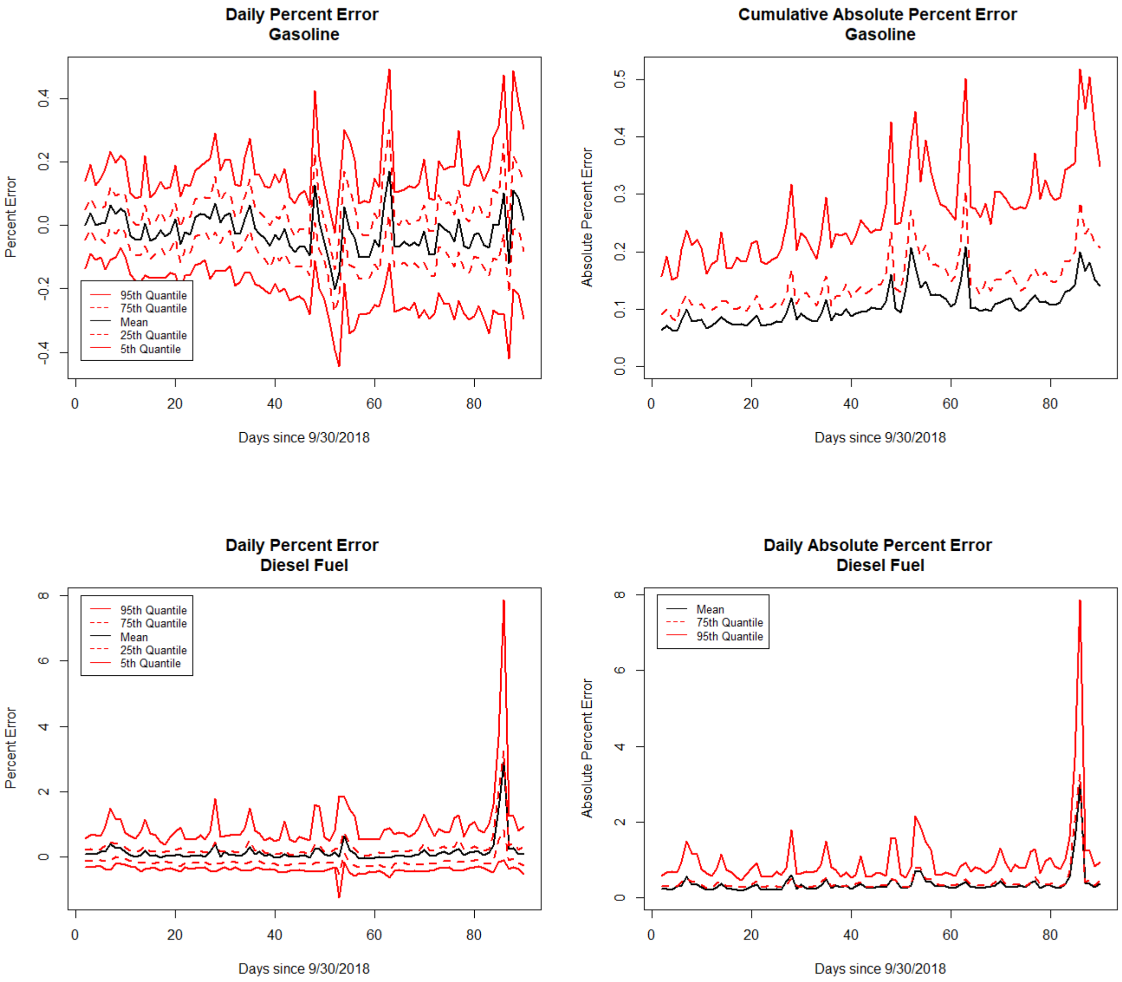


Figure 4: Daily forecast errors over time

Finally, a subset of stores is selected to be shown in Figure 5 on an individual bases to show the models ability to make individual forecasts. These stores were selected at random, and comprise three gasoline sales signals and three diesel fuel signals. Like the above plots, forecasts begin on September 30, 2018.

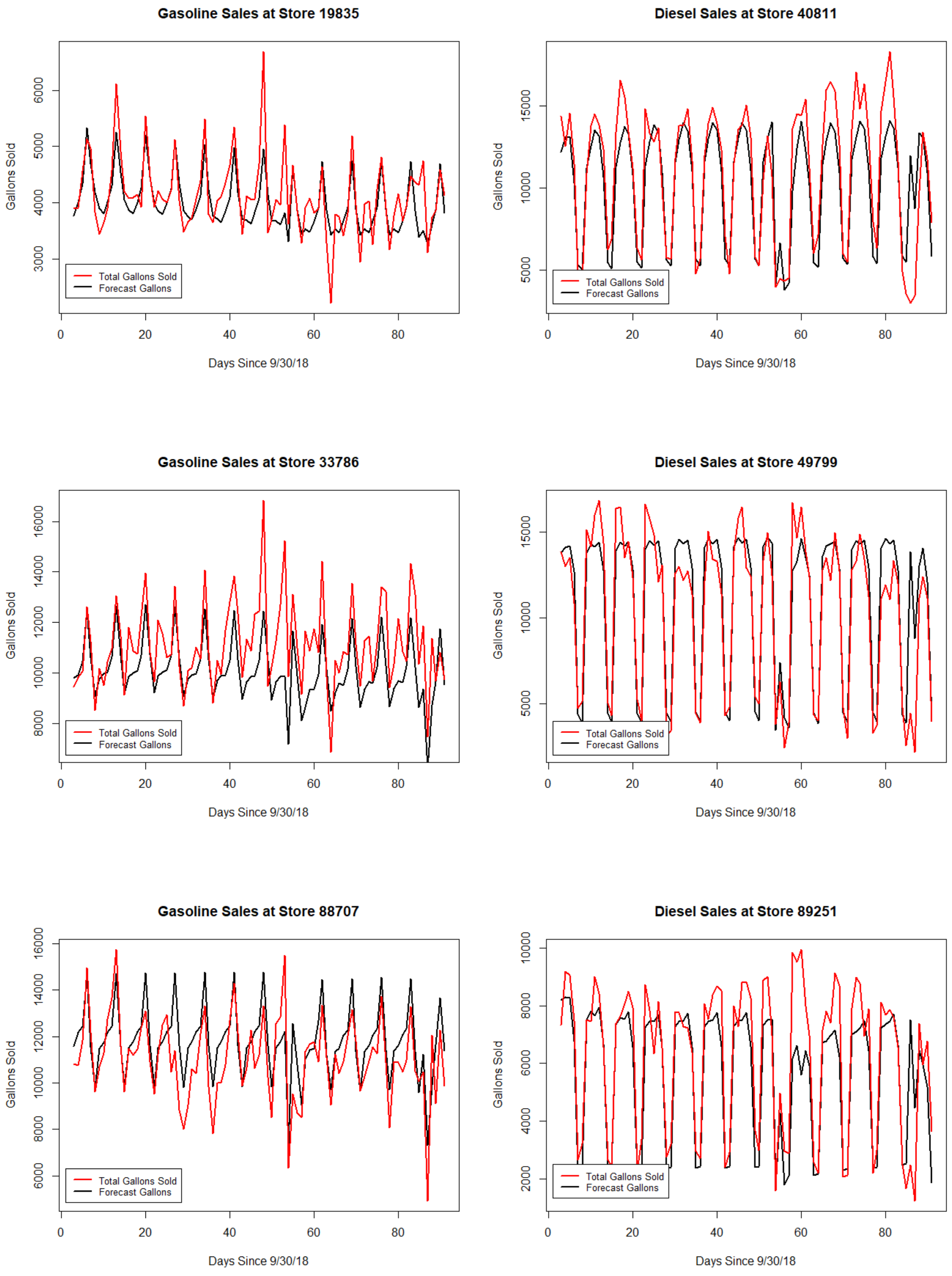


Figure 5: Selection of daily forecasts by location.

Discussion

The primary purpose of the project was to develop a set of time series models that have strong predictive power over long periods of time. The secondary purpose was to test the model’s ability in detecting changes in the structure of a store’s performance.

In terms of long-term performance, cumulative model performance was very strong with a maximum mean absolute percent error of 2.77% and 1.33% for gasoline and diesel respectively. On an individual store basis, plotting the distribution of mean percent error on cumulative performance was well represented by a normal distribution. Analysis of plots over the three 91-day time periods demonstrated that models are stable in terms of their predictive performance.

These models show their usefulness in terms of long-term forecasting and are useful for applications such as budgeting. The model shows strong predictive performance for either fuel type. However, on a daily basis, diesel fuel was much more highly predictive. Diesel customers, by and large, are purchasing their fuel for commercial purposes. Commercial buyers are much more predictable in their habits and are largely making their purchases on specific weekdays. Commercial customers are in industries such as construction, shipping, parcel services, landscaping, etc. Gasoline, in contrast, is largely purchased by everyday consumers and is much less predictable.

In terms of mean absolute percent error, the models are clearly shown to follow a gamma distribution, both graphically and running the goodness of fit test. This can be leveraged to demonstrate when a store’s performance has fundamentally changed in structure. Similar to a traditional confidence test, stores that have error rates in the higher percentiles tend to have some fundamental shift in their signal structure that is not expected.

It can be noted that the all of the test statistics testing for normality came back as significant, meaning that the data is not normally distributed. However, upon inspection of the histogram showing mean percent error, the data appears approximately normal. The Shapiro-Wilks Test of Normality is known to be too powerful, and the graphical representation should be trusted over the test statistic.

Some of these signal changes can be external, such as a competitor lowering prices at a nearby location. This will affect consumer behavior and shift them to purchase their fuel at a different location. In a similar vein, over the road truck drivers, especially drivers for larger companies, will use a system of electronic optimizers for their fuel purchasing. These electronic optimizers consider a driver’s potential route and selects locations to fuel where the total cost will be the least. On diesel purchases, this means that competing locations that might influence the sales of a modeled location may be separated by vast distances but follow the same interstate corridors.

Another thing that was not considered but worthy of future consideration is weather. In terms of gasoline, people are easily influenced by inclement weather conditions. Winter weather has a significant effect on a persons average daily driving habits, as people tend to avoid dangerous driving conditions. In terms of diesel fuel, diesel jelling is another consideration. At temperatures below approximately 30 degrees, standard diesel fuel will begin to solidify in a waxy jell like substance, and this can clog up and damage engines. To combat this, additives are added to the fuel, and drivers will idle their engines continuously to prevent the temperature from dipping and prevent jelling. These fuel additives unfortunately reduce engine efficiency, meaning drivers must burn more fuel over a given distance traveled. This causes a general increase in fuel sales during periods of cold weather. Weather, while having a noticeable impact on fuel sales, was not considered for modeling as access to reliable forecasts was not available.

Certain company policies also have an impact on sales. Undercutting and restoration pricing are strategies that are used by Kwik Trip meant to influence the total sales in a market. Undercutting is simply the lowering of a price below that of the competition. This means that while there is less profit margin per gallon of fuel sold, undercutting should drive more sales as customers are pulled away from the competition. The differences between a models expected performance and actual performance after a change is made is useful in evaluating the effectiveness of these changes and whether a given set of changes motivated an increase or decrease in profits.

Alternatively, restoration pricing is a strategy to increase profit margins while sacrificing sales. A store may price itself higher than its competition, meaning the store increases the profit margin on each gallon of fuel sold. Customers tend to choose the lowest price option, and stores typically see less total sales of fuel. However, the primary goal of restoration pricing is to influence other competitors to raise their prices in response. If competing locations raise their prices, an individual store will see less loss in terms of sales and an increase in profitability as a result. In production, these models have shown to be useful for measuring the impact of these sales when combined with margin information.

One thing that is noticeable in these models is the significant number of covariates. In a traditional modeling approach, model complexity is generally kept to a minimum to make the models as interpretable as possible. The focus of this study is to build a system of models focused on predictive power, and shares more in common with a machine learning approach to analysis. Therefore, interpretation of the models largely wasn’t considered.

CHAPTER V

SUMMARY CONCLUSIONS AND RECOMMENDATIONS

Summary

The primary purpose of the project was to develop a set of time series models that have strong predictive power over long periods of time. The secondary purpose was to test the model's ability in detecting changes in the structure of a store's performance. The model used a selection of variables to consider seasonal trends, the most recent week, weekdays, holidays, and its time since opening. The models were applied to 288 stores that sold gasoline and 204 stores that sold diesel fuel with data from September 30, 2016 to September 26, 2019.

The data was split on September 30, 2018 and forecasts were sent out for multiples of 28 days up to 336-day forecasts. These forecasts were analyzed for their cumulative performance, error rates and error distributions. In a global sense, the models made strong predictions of less than 2.74% error for gasoline and 1.33% diesel fuel. Analysis of the error plots showed that they're largely stable over time, and can be relied upon for accurate long-term predictions. Analysis of the mean absolute percent error rates suggests that a gamma distribution represents the distribution of errors well. Stores that cluster near the extreme edges of the distribution can be selected for further investigation, similarly to a standard significance test, as there is likely a fundamental change in the structure of the sales signal that isn't accounted for by the model.

Conclusions

Based on the results of the study, the following conclusions were reached:

1. The models were able to make accurate long-term predictions despite being over defined.
2. The models were stable over long periods of time.
3. Error rates were consistent with a normal and gamma distributions, and this can be used for a standardized procedure to select locations for further investigation.

Recommendations

Based on the results and conclusions, recommendations for future investigations are as follows:

1. If accurate weather forecasts are available, including them as a set of covariates in the model could prove beneficial.
2. Similar to weather, implement changes to include changes in company policy to measure their impact on store performance, such as changing in pricing, marketing strategy, etc.
3. Analyze the parameters of the gamma kernel to better understand how new stores mature over time.
4. Consider developing a system of multivariate models to consider retail locations that related by geographic location or located along highway corridors.

REFERENCES

Basseville M. (2019) *40Irisa.* Irisa.fr. people.irisa.fr/Michele.Basseville/

Cryer J. & Chan K. (2010). *Time Series Analysis with Applications in R.* New York: Springer

Farzaneh A. (2009). Change point detection with multivariate control charts by artificial Neural network. *International Journal of Advanced Manufacturing Technology,* 97:3179–3190

Hossein H., Mouhamad M., Pantelis C., & Adedapo O. (2017). A statistical study of the short- and long-term drivers of crude oil prices. *OPEC Energy Review*, 93-114

Kwon H., & Godman B., (2016). Do newly marketed generic medicines expand markets using descriptive time series analysis and mixed logit models? Korea as an exemplar and its implications. *BMC Health Services Research,* 16:130

Lie T & Wei Z., (1982). Least Squares Estimates in Stochastic Regression Models With Applications to Identification and Control of Dynamic Systems. *The Annals of Statics,* 10(2), 154-166

Millard SP (2013). *EnvStats: An R Package for Environmental Statistics*. Springer, New York URL http://www.springer.com.

National Institute of Standards and Technology. *Introduction to Time Series Analysis*. nist.gov. Itl.nist.gov/div898/handbook/pmc/section4/pmc4.html

UK Center for the Measurement of Government Activity. (2008). From Holt-Winters to ARIMA Modeling: Measuring the Impact on Forecasting Errors for Components of Quarterly Estimates of Public Service Output. *Office for National Statistics.*

R Core Team (2018). R: A language and environment for statistical computing. *R Founaion for Statistical Computing*, Vienna, Austria. https://www.R-project.org/.

Ravinder H., (2013). Determining The Optimal Values Of Exponential Smoothing Constants – Does Solver Really Work? *American Journal of Business Education.* 6(2), 347 – 360

Sheather S. (2009). *A Modern Approach to Regression with R*. New York: Springer

Shmeuli G., Bruce P., Yahav I., Patel N., Lichtendahl K., (2018). Data Mining for Business Analytics. New Jersey: Wiley Publishing

Sun T., Zhang T., Teng Y., Chen Z. & Fang J. (2019). Monthly Electricity Consumption Forecasting Method Based on X12 and STL Decomposition Model in an Integrated Energy System. *Mathematical Problems in Engineering,* 9012543, 16

Tsai W., & Lin J., (2019). The Relations of Oil Price Change with Fear Gauges in Global Political and Economic Environment. *Energies 2019, 12, 2982*

Zhang J., Wei Z., Yan Z., Zhou M., & Pani A., (2019). Online Change-Point Detection in Sparse Time Series with Application to Online Advertising. *Transactions on Systems, Man, and Cybernetics: Systems*, 49(6), 1141-1151

Zhang Y., Zhong M., Geng N., & Jiang Y. (2017). Forecasting electric vehicles sales with univariate and multivariate time series models: The case of China. *PLoS ONE* 12(5):e0176729'

APPENDIX A

SUPPLEMENTAL GRAPHS FOR CUMULATIVE PREDICTIVE ACCURACY

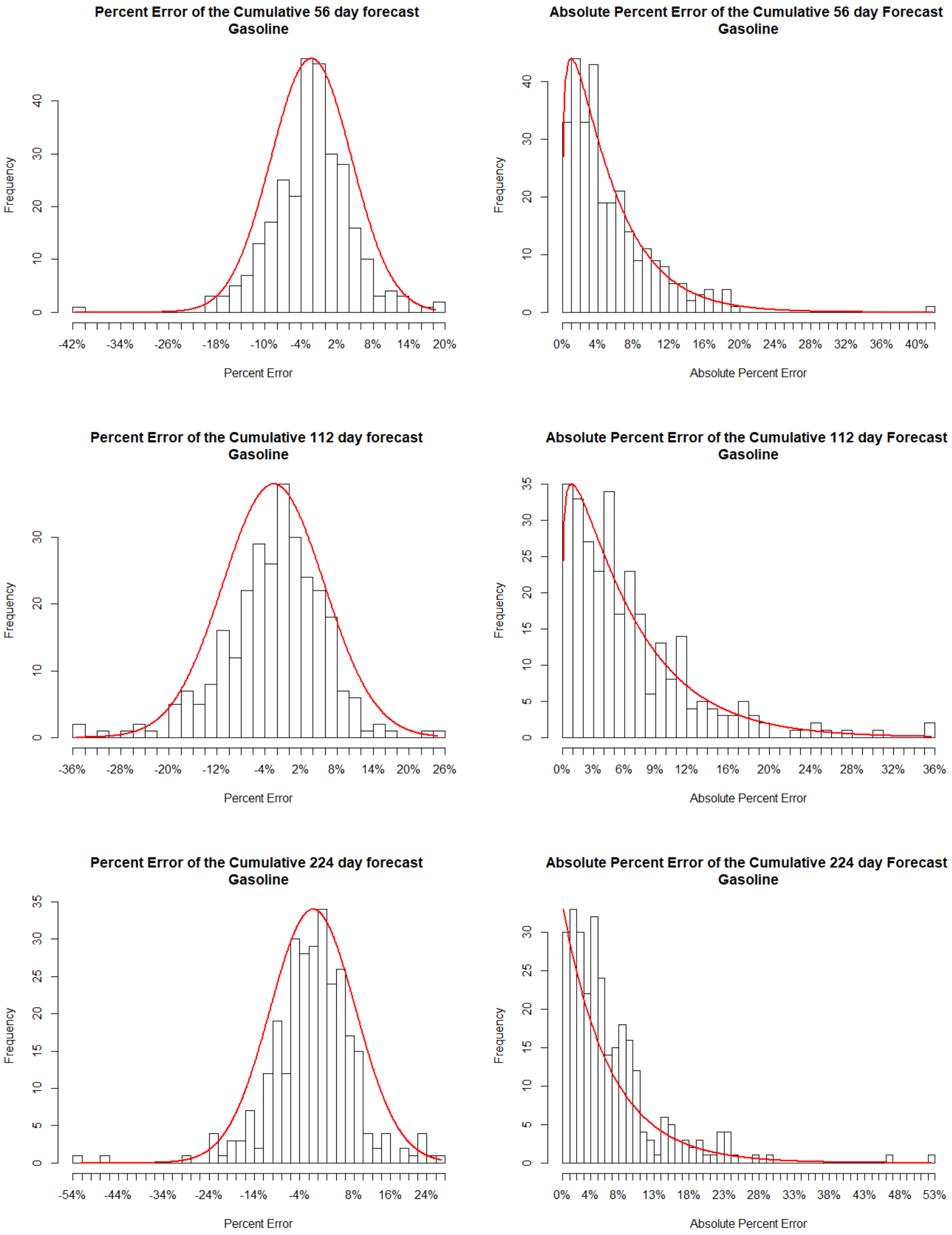


Figure 6: Supplementary figures showing model accuracy for forecasted gasoline sales

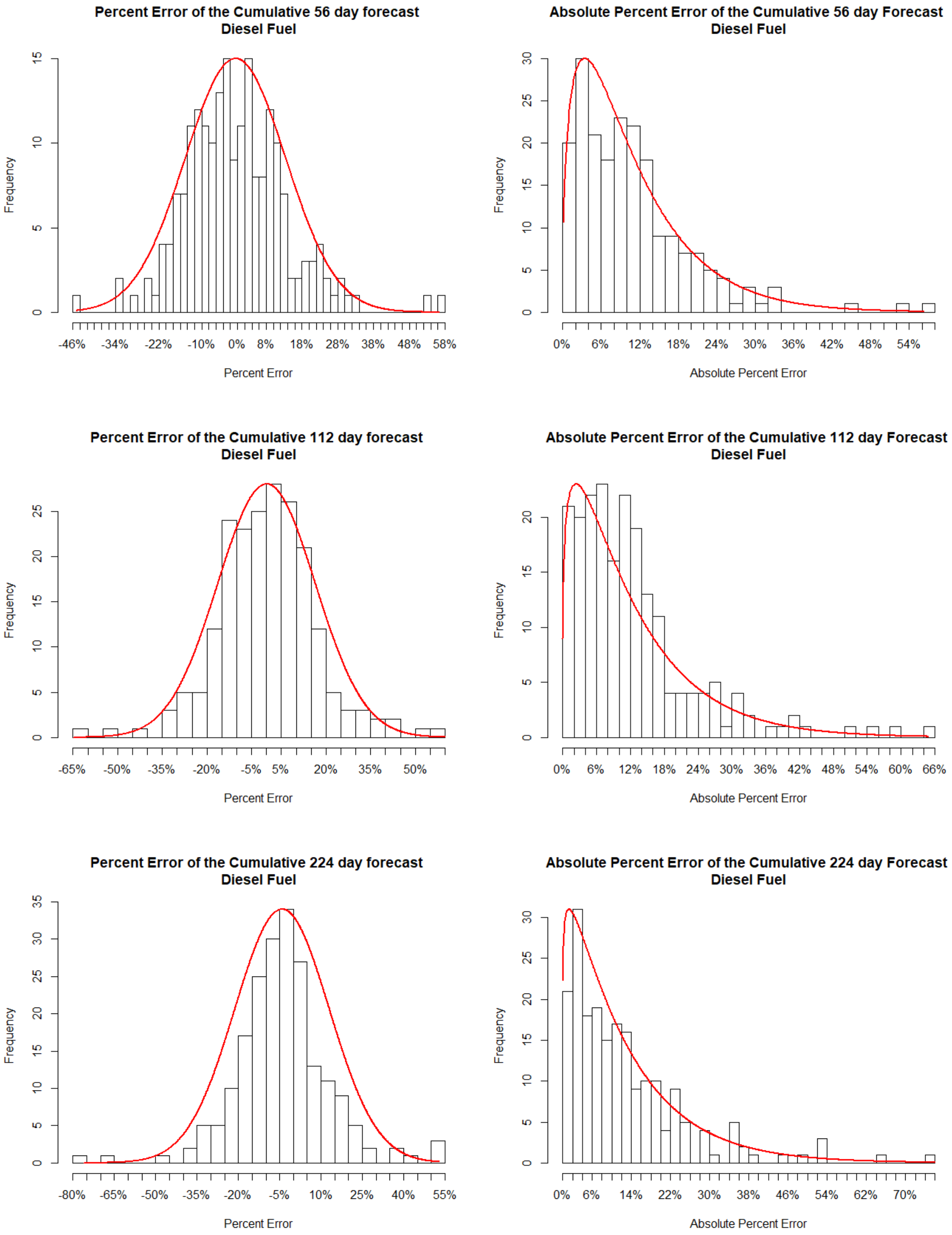


Figure 7: Supplementary figures showing model accuracy for forecasted diesel fuel sales

APPENDIX B

SUPPLEMENTAL GRAPHS FOR DAILY PREDICTIVE ACCURACY

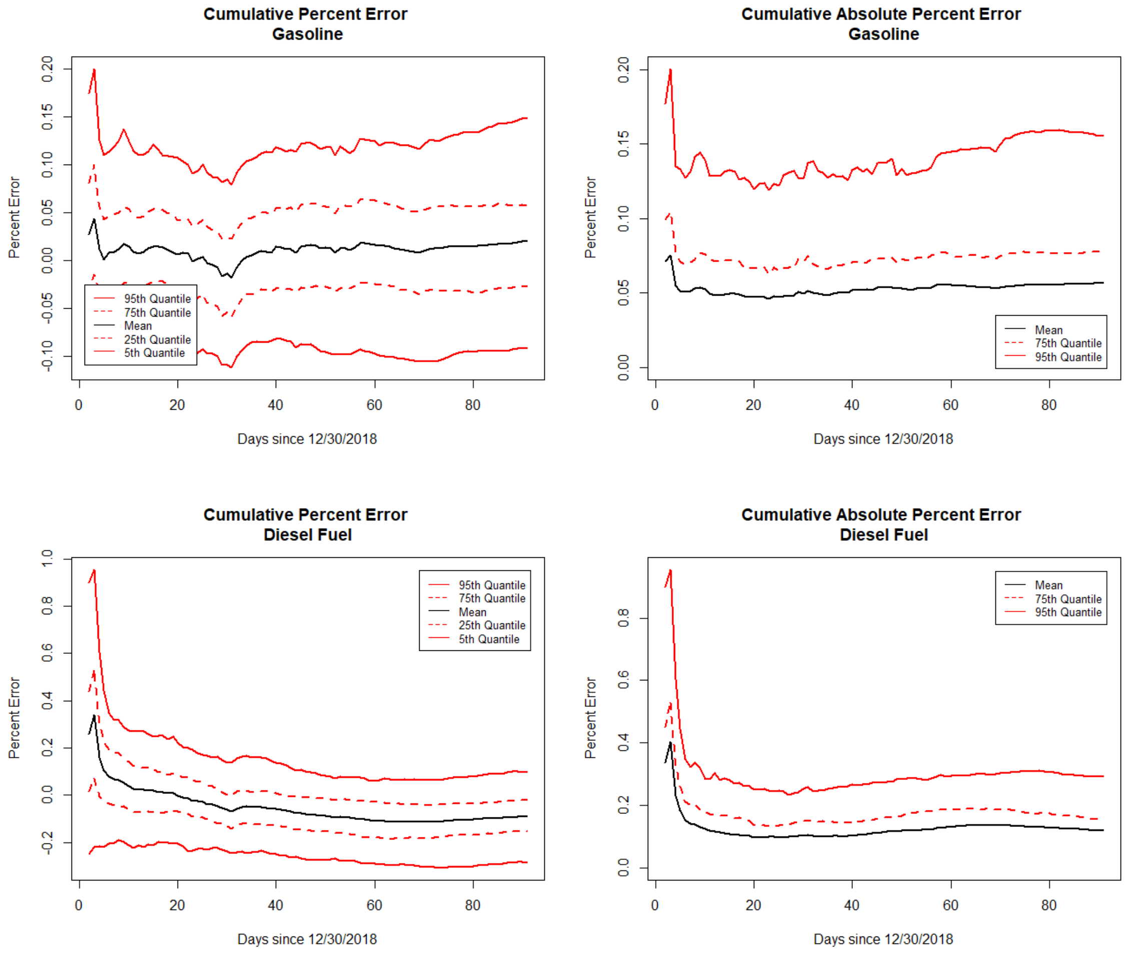


Figure 8: Supplemental figures showing the daily cumulative forecast accuracy, December

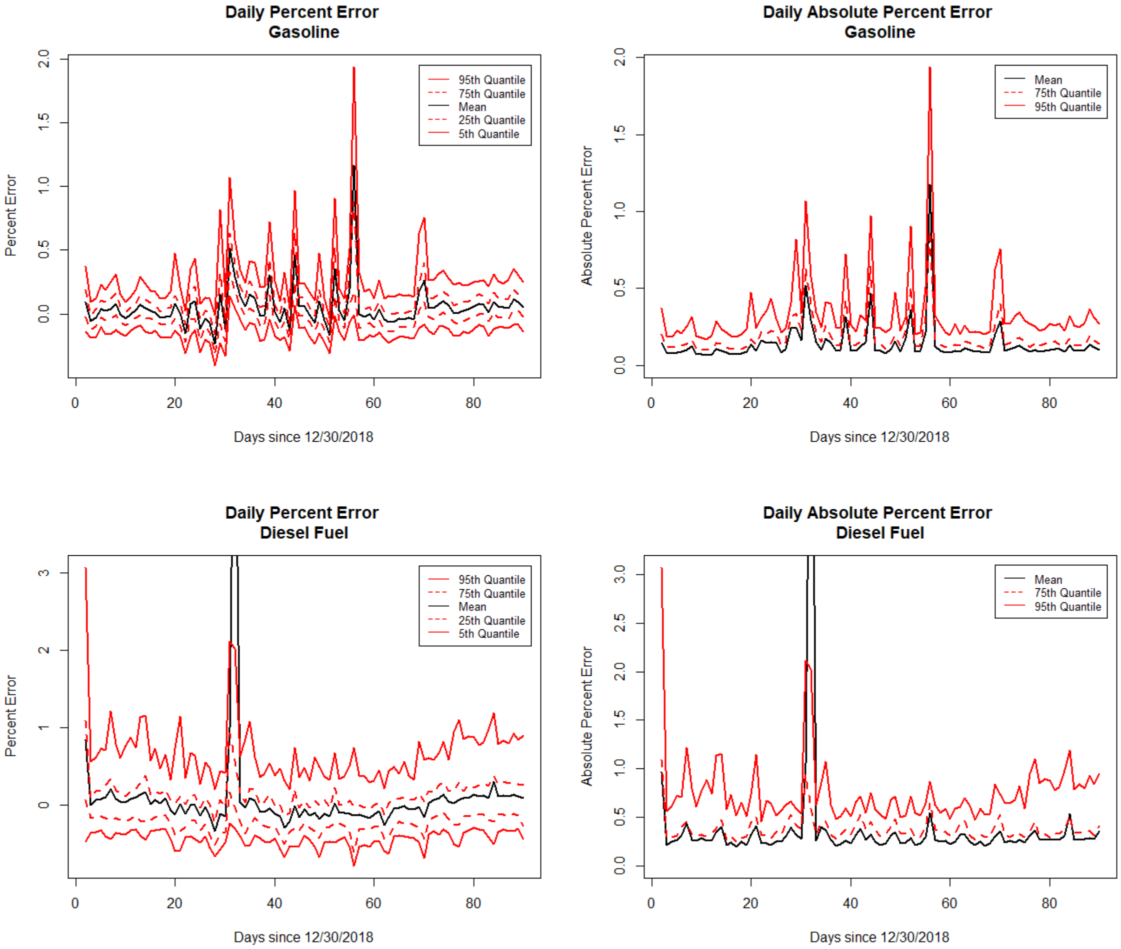


Figure 9: Supplemental figures showing the daily forecast accuracy, December

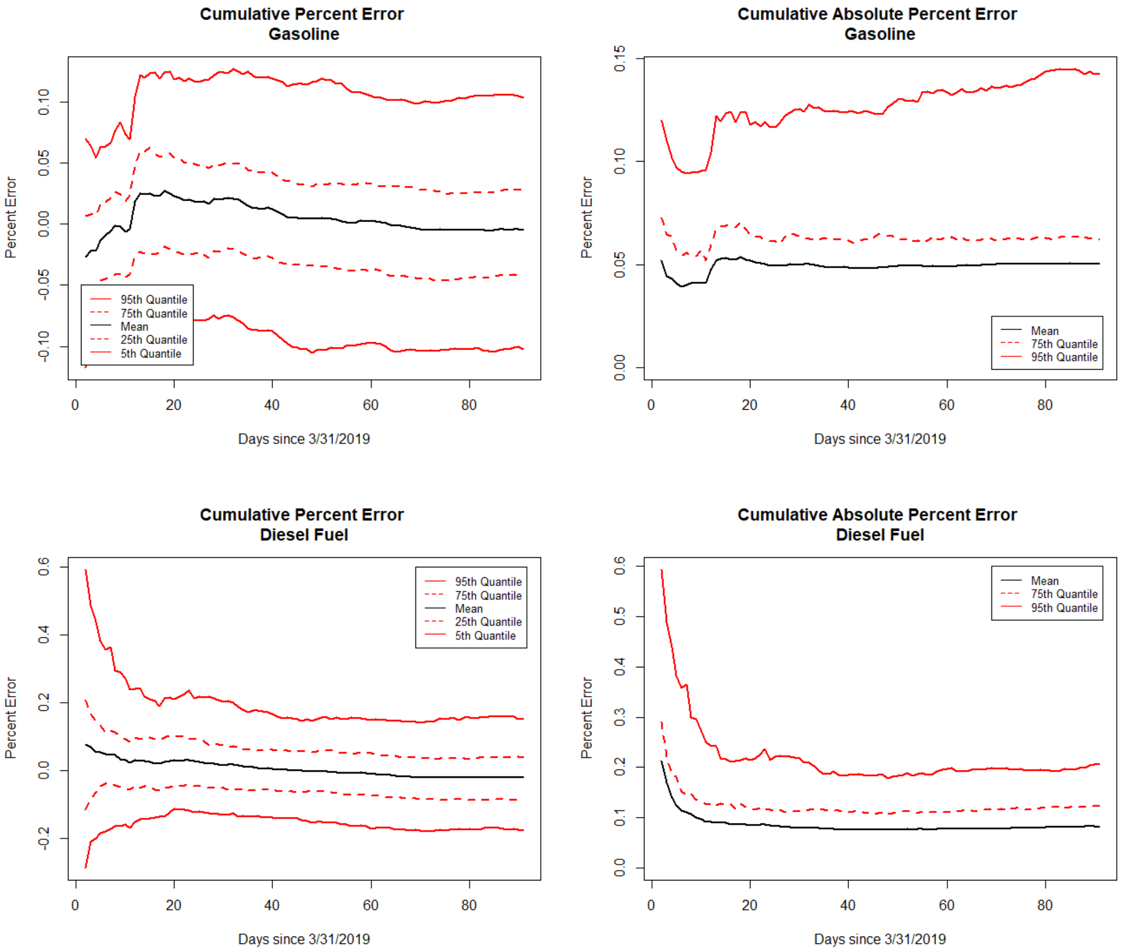


Figure 10: Supplemental figures showing the daily cumulative forecast accuracy, March

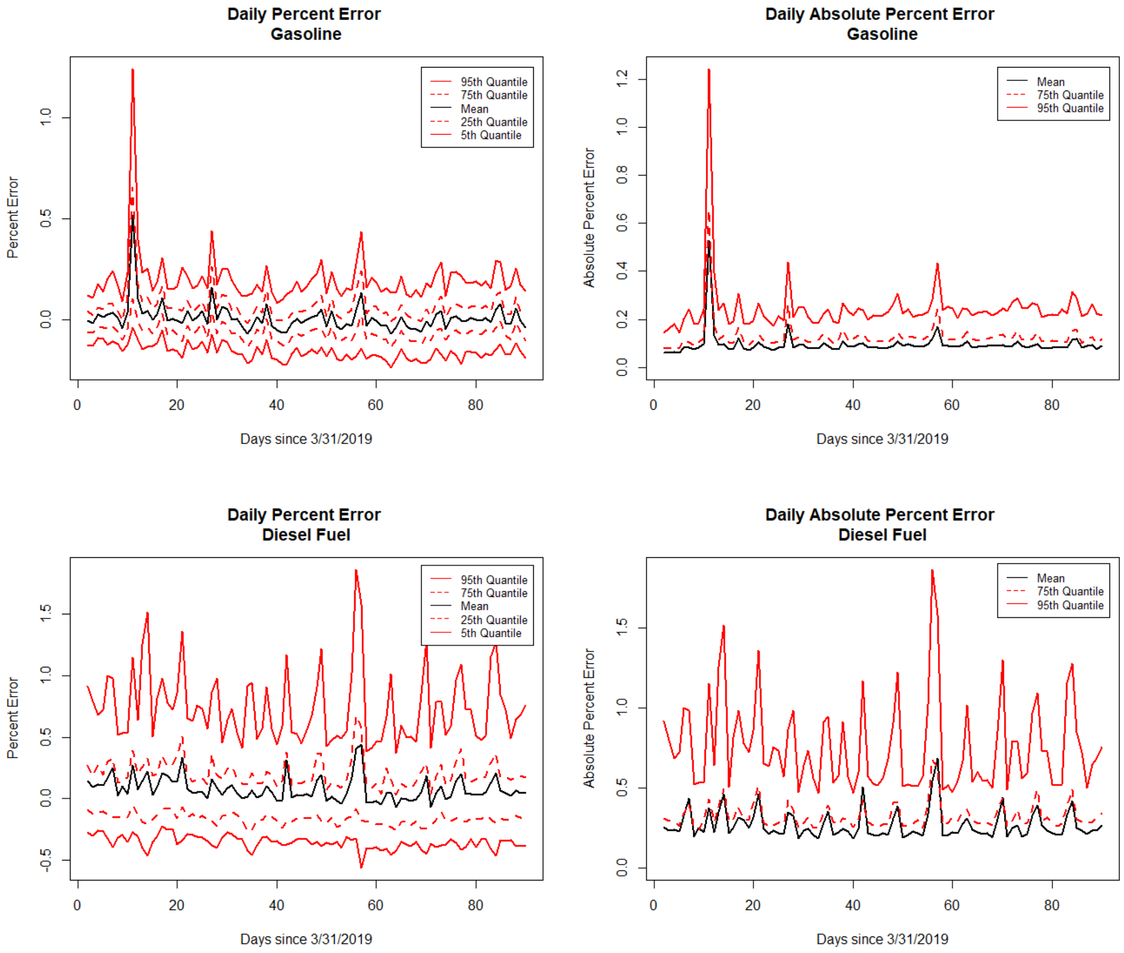


Figure 11: Supplemental figures showing the daily forecast accuracy, March

APPENDIX C

EXAMPLE CODE

```r
store.puller <- function(data, store.for.analysis, cut.date = Sys.Date
()){
  # Pulls out Gallons and Budget for later use, this will be used for m
odel
  # comparison and plotting
  store.data = data[data$Store.No == store.for.analysis,]
  Gallons.and.Budget = select(store.data,Date,Fiscal.Year,
                              Fiscal.Period,Gallons,Budget)

  store.data = store.data[store.data$Date < cut.date &
                          store.data$Days.Since.Open >= 0,]
  store.data$Budget = NULL

  store.data$Gallons = ifelse(is.na(store.data$Gallons) == T,
                              0, store.data$Gallons)

  # Adds in lag information
  store.data$lag.1 = store.data$lag.2 = store.data$lag.3 = store.data$l
ag.4 = store.data$lag.5 = store.data$lag.6 = store.data$lag.7 = NA

  for(i in nrow(store.data):8){
    store.data$lag.1[i] = store.data$Gallons[i-1]
    store.data$lag.2[i] = store.data$Gallons[i-2]
    store.data$lag.3[i] = store.data$Gallons[i-3]
    store.data$lag.4[i] = store.data$Gallons[i-4]
    store.data$lag.5[i] = store.data$Gallons[i-5]
    store.data$lag.6[i] = store.data$Gallons[i-6]
    store.data$lag.7[i] = store.data$Gallons[i-7]
  }

  store.data = na.omit(store.data); store.data$Store.No = NULL

  out = list(store.data = store.data,Gallons.and.Budget = Gallons.and.B
udget)
  return(out)
}
```

```r
forecast.builder <- function(data,store.for.analysis,model,days.ahead,
                             start.new.period = 1){

  if(days.ahead == "End of Period"){
    last.28.days = data$Fiscal.Period[nrow(data):(nrow(data)-27)]
    days.in.current.period = sum(last.28.days == data$Fiscal.Period[nrow(data)])
    days.ahead = 28 - days.in.current.period
    if(days.ahead < start.new.period){
      days.ahead = days.ahead + 28
    }
  } else {
    last.28.days = data$Fiscal.Period[nrow(data):(nrow(data)-27)]
    days.in.current.period = sum(last.28.days == data$Fiscal.Period[nrow(data)])
  }

  # Builds new dates for forecasting
  new.dates = as.Date((max(data$Date)+1):(max(data$Date)+days.ahead), origin = "1970-01-01")

  # Builds new Days of Week for forecasting
  days.of.the.week = c("Monday","Tuesday","Wednesday",
                       "Thursday","Friday","Saturday","Sunday")
  new.day.of.week = NULL
  if(data$Day.Of.Week[data$Date == max(data$Date)] == "Monday"){
    x = (1+1):(days.ahead+1); x = x%%7; x = ifelse(x==0,7,x); new.day.of.week = days.of.the.week[x]
  } else if(data$Day.Of.Week[data$Date == max(data$Date)] == "Tuesday"){
    x = (1+2):(days.ahead+2); x = x%%7; x = ifelse(x==0,7,x); new.day.of.week = days.of.the.week[x]
  } else if(data$Day.Of.Week[data$Date == max(data$Date)] == "Wednesday"){
    x = (1+3):(days.ahead+3); x = x%%7; x = ifelse(x==0,7,x); new.day.of.week = days.of.the.week[x]
  } else if(data$Day.Of.Week[data$Date == max(data$Date)] == "Thursday"){
    x = (1+4):(days.ahead+4); x = x%%7; x = ifelse(x==0,7,x); new.day.of.week = days.of.the.week[x]
  } else if(data$Day.Of.Week[data$Date == max(data$Date)] == "Friday"){
    x = (1+5):(days.ahead+5); x = x%%7; x = ifelse(x==0,7,x); new.day.of.week = days.of.the.week[x]
  } else if(data$Day.Of.Week[data$Date == max(data$Date)] == "Saturday"){
    x = (1+6):(days.ahead+6); x = x%%7; x = ifelse(x==0,7,x); new.day.of.week = days.of.the.week[x]
  } else if(data$Day.Of.Week[data$Date == max(data$Date)] == "Sunday"){
    x = (1+7):(days.ahead+7); x = x%%7; x = ifelse(x==0,7,x); new.day.o
f.week = days.of.the.week[x]
  }

  # Builds periods and transforms data

  Christmas = as.Date(c("2016-12-25", "2017-12-25", "2018-12-25",
                        "2019-12-25", "2020-12-25"))
  Thanksgiving = as.Date(c("2016-11-24", "2017-11-23", "2018-11-22",
                           "2019-11-28", "2020-11-26"))
  New.Years = as.Date(c("2017-01-01", "2018-01-01", "2019-01-01",
                        "2020-01-01"))
  The.4th = as.Date(c("2016-07-04", "2017-07-04", "2018-07-04",
                      "2019-07-04", "2020-07-04"))
  Memorial.Day = as.Date(c("2016-05-30", "2017-05-29",
                           "2018-05-28", "2019-05-27", "2020-05-25"))
  Labor.Day = as.Date(c("2016-09-05","2017-09-04","2018-09-03",
                        "2019-09-02", "2020-09-07"))

  holiday = rep("Normal.Day",length(new.dates))
  for(i in 1:length(new.dates)){
    if(sum(new.dates[i] == Christmas)>0){
      holiday[i] = "Christmas"
    } else if(sum(new.dates[i] == Thanksgiving)>0){
      holiday[i] = "Thanksgiving"
    } else if(sum(new.dates[i] == New.Years)>0){
      holiday[i] = "New.Years"
    } else if(sum(new.dates[i] == The.4th)>0){
      holiday[i] = "The.4th.of.July"
    } else if(sum(new.dates[i] == Memorial.Day)>0){
      holiday[i] = "Memorial.Day"
    } else if(sum(new.dates[i] == Labor.Day)>0){
      holiday[i] = "Labor.Day"
    }
  }

  periods.by.day = c(rep(1,28),rep(2,28),rep(3,28),rep(4,28),rep(5,28),
                     rep(6,28),rep(7,28),rep(8,28),rep(9,28),rep(10,2
8),
                     rep(11,28),rep(12,28),rep(13,28))
  previous.finished.period = (data$Fiscal.Period[nrow(data)]-1)

  day.of.year = 28*previous.finished.period+days.in.current.period

  new.period.days = periods.by.day[(day.of.year+1):(day.of.year+days.ah
ead)]
  new.period.days = ifelse(is.na(new.period.days) == T,
                           1, new.period.days)

  sin.transform = sin(2*pi*new.period.days/13);
  cos.transform = cos(2*pi*new.period.days/13)

  # Builds influence from new store drive
  new.days.since.open = (max(data$Days.Since.Open)+1):(max(data$Days.Si
nce.Open)+days.ahead)
  new.new.store.drive = dgamma(new.days.since.open, shape = 1, scale =
30)

  # Complies data into one source
  new.data = data.frame(Date = new.dates,
                        Day.Of.Week = new.day.of.week,
                        sin.transform = sin.transform,
                        cos.transform = cos.transform,
                        new.store.drive = new.new.store.drive,
                        holiday = holiday)

  new.data$lag.1 = new.data$lag.2 = new.data$lag.3 = new.data$lag.4 =
    new.data$lag.5 = new.data$lag.6 = new.data$lag.7 = NA

  new.data$lag.1[1] = data$Gallons[nrow(data)-0]
  new.data$lag.2[1] = data$Gallons[nrow(data)-1]
  new.data$lag.3[1] = data$Gallons[nrow(data)-2]
  new.data$lag.4[1] = data$Gallons[nrow(data)-3]
  new.data$lag.5[1] = data$Gallons[nrow(data)-4]
  new.data$lag.6[1] = data$Gallons[nrow(data)-5]
  new.data$lag.7[1] = data$Gallons[nrow(data)-6]


  # Forecasts are made
  pred = data.frame(predict(model, newdata = new.data[1,],
                            interval = "predict"))
  for(i in 2:days.ahead){
    new.data$lag.1[i] = pred$fit[[i-1]]
    new.data$lag.2[i] = new.data$lag.1[[i-1]]
    new.data$lag.3[i] = new.data$lag.2[[i-1]]
    new.data$lag.4[i] = new.data$lag.3[[i-1]]
    new.data$lag.5[i] = new.data$lag.4[[i-1]]
    new.data$lag.6[i] = new.data$lag.5[[i-1]]
    new.data$lag.7[i] = new.data$lag.6[[i-1]]

    for(j in 1:3){
      pred[i,j] = predict(model, newdata = new.data[i,],interval = "pre
dict")[j]
    }
  }
  pred$Date = new.dates
  return(pred)
}
```

```
full.store.deficit <- function(gallons.clean,stores.to.exclude,
                               ahead = "End of Period",
                               behind = "Start of Period",
                               print = FALSE,
                               cut.date = Sys.Date(),
                               print.diagnostic = FALSE,
                               factor.for.analysis = NA,
                               min.days.open = 367){

  k = 0
  store.deficit = data.frame()

  for(i in 1:length(levels(gallons.clean$Store.No))){ #
    if(print == TRUE){
      print(i)
      print(levels(gallons.clean$Store.No)[i])
    }
    if(length(stores.to.exclude[stores.to.exclude == levels(gallons.cle
an$Store.No)[i]]) == 0){

      store.data.both = store.puller(gallons.clean,levels(gallons.clean
$Store.No)[i],
                                     cut.date = cut.date)

      if(max(store.data.both$store.data$Days.Since.Open)<min.days.open)
{
        data.for.forecasting = select(store.data.both$store.data,
                                      Gallons,Date,
                                      Day.Of.Week,new.store.drive,
                                      sin.transform,cos.transform,
                                      lag.1,lag.2,lag.3,lag.4,
                                      lag.5,lag.6,lag.7)
      } else {
        data.for.forecasting = select(store.data.both$store.data,
                                      Gallons,Date,
                                      Day.Of.Week,new.store.drive,
                                      sin.transform,cos.transform,holid
ay,
                                      lag.1,lag.2,lag.3,lag.4,
                                      lag.5,lag.6,lag.7)
      }
      if(nrow(data.for.forecasting) != 0 &
         sum(data.for.forecasting$Gallons) > 1000){
        model = lm(Gallons~.+
                     Day.Of.Week*(lag.1+lag.2+lag.3+lag.4+lag.5+lag.6+l
ag.7),
                   data = data.for.forecasting)

        pred = forecast.builder(store.data.both$store.data,
                                levels(gallons.clean$Store.No)[i],mode
l,ahead)

        current.year = store.data.both$Gallons.and.Budget$Fiscal.Year[s
tore.data.both$Gallons.and.Budget$Date == cut.date]

        current.period = store.data.both$Gallons.and.Budget$Fiscal.Peri
od[store.data.both$Gallons.and.Budget$Date == cut.date]

        if(behind == "Start of Period"){
          historic.gallons =
            store.data.both$Gallons.and.Budget[store.data.both$Gallons.
and.Budget$Date < cut.date&
                                                store.data.both$Gallons.
and.Budget$Fiscal.Year == current.year&
                                                store.data.both$Gallons.
and.Budget$Fiscal.Period == current.period,]

        } else {

          historic.gallons =
            store.data.both$Gallons.and.Budget[store.data.both$Gallons.
and.Budget$Date < cut.date &
                                                store.data.both$Gallons.
and.Budget$Date > cut.date - behind,]
        }


        full.fit = data.frame(Date = c(historic.gallons$Date, pred$Dat
e),
                              Gallons = c(historic.gallons$Gallons,pred
$fit[1],rep(NA,nrow(pred)-1)),
                              Forecast = c(rep(NA,nrow(historic.gallon
s)),pred$fit),
                              lower.bound = c(rep(NA,nrow(historic.gall
ons)),pred$lwr),
                              upper.bound = c(rep(NA,nrow(historic.gall
ons)),pred$upr))

        full.fit = left_join(full.fit,store.data.both$Gallons.and.Budge
t[,c(1,5)],"Date")
        full.fit = full.fit[,c(1,2,3,6,4,5)]
        if(ahead == "End of Period"){
          if(nrow(full.fit) > 28){
            full.fit = full.fit[29:nrow(full.fit),]

            temp.Gallons = full.fit$Gallons; temp.Gallons  = temp.Gallo
ns[!is.na(temp.Gallons)]
            temp.Forecast = full.fit$Forecast
            temp.Forecast = temp.Forecast[!is.na(temp.Forecast)]
            temp.Forecast = temp.Forecast[1:length(temp.Forecast)]
            cumulative.fit = sum(temp.Gallons) + sum(temp.Forecast)

          } else {

            temp.Gallons = full.fit$Gallons
            temp.Gallons  = temp.Gallons[!is.na(temp.Gallons)]
            temp.Forecast = full.fit$Forecast
            temp.Forecast = temp.Forecast[!is.na(temp.Forecast)]
            temp.Forecast = temp.Forecast[2:length(temp.Forecast)]
            cumulative.fit = sum(temp.Gallons) + sum(temp.Forecast)
          }
        } else {
          temp.Gallons = full.fit$Gallons; temp.Gallons  = temp.Gallons
[!is.na(temp.Gallons)]
          temp.Forecast = full.fit$Forecast
          temp.Forecast = temp.Forecast[!is.na(temp.Forecast)]
          temp.Forecast = temp.Forecast[2:length(temp.Forecast)]
          cumulative.fit = sum(temp.Gallons) + sum(temp.Forecast)
        }

        budget = sum(full.fit$Budget)

        store.deficit[i-k,1] = na.omit(levels(gallons.clean$Store.No)
[i])
        store.deficit[i-k,2] = NA
        store.deficit[i-k,3] = cumulative.fit
        store.deficit[i-k,4] = budget
        store.deficit[i-k,5] = cumulative.fit - budget
        store.deficit[i-k,6] = i
        store.deficit[i-k,7] = nrow(full.fit)
        if(is.na(factor.for.analysis) == FALSE){
          store.deficit[i-k,8] = model$coefficients[factor.for.analysi
s]
        }
      } else {
        k = k + 1

        print("The following store has been excluded for lack of data:
")
        print(levels(gallons.clean$Store.No)[i])
      }
    } else {
      k = k + 1

      print("The following store has been excluded manually:")
      print(levels(gallons.clean$Store.No)[i])
    }
  }

  if(ncol(store.deficit) == 7){
    names(store.deficit) = c("Store.No","City","Expected.Total",
                             "Budget","Deficit","Level.Location","nrow
")
  } else {
    names(store.deficit) = c("Store.No","City","Expected.Total","Budget
",
                             "Deficit","Level.Location","nrow","Factor.
Impact")
  }

  store.deficit = store.deficit[order(store.deficit$Deficit),]
  return(store.deficit)
}
```