Forecasting Monthly Product Usage

Tonnage Group 010 at Branch 1

Michael Verwijst

Executive Summary

One major challenge that procurement teams across all industries face is determining how much and when to buy products to set up the organization for success. Profitability starts with the purchasing team being able to cost-effectively acquire the necessary products and materials needed to fill customer orders without buying too much and inflating inventory costs. The main challenge that the procurement team at this organization faces is the lack of tools available for planning future buys. The current method is backwards-looking and relies heavily on viewing usage history and making a best guess at what will be needed in the coming months. With the industry shift toward more advanced technology and utilization of data tools, this will put the organization at a disadvantage in the market.

The aim of this project is to address the above problem by identifying a 6-month forecasting tool for the purchasing team to utilize. Rather than relying on "tribal knowledge" and educated guessing techniques, the purchasing agents will be provided with statistically-backed models to predict the next six months of usage at the product level. Armed with the necessary tools to make more efficient buys, the purchasing team will be able to acquire more accurate quantities of material which will result in more fulfilled orders and a leaner inventory.

The data set used consists of 36 months of usage history, January 2016 through December 2018, for each part in product group 010. Usage data from January 2019 through June 2019 was held out as test data, and each model's forecast accuracy was compared to this test data set. During this analysis, nine different predictive techniques were evaluated to determine which model type was most accurate compared to actual usage. After thorough evaluation, it was discovered that all nine models were more accurate than the incumbent method of educated guessing what the organization will need. Of the models evaluated, the recommendation is to move forward with the Monthly Average Forecasting model because of its prediction accuracy as well as its simplicity and ease of deployment into the production ERP system.

The project team, co-led by the VP of Purchasing and the VP of IT, defined a successful project as a tool that could produce more accurate product usage quantity forecasts than the incumbent method currently used by the procurement team. After thorough analysis of multiple techniques, this project has delivered those results and has been deemed a success. With the positive results and feedback from the procurement team, the recommendation is to expand the scope of this project to include additional product groups and additional branches. Doing so will allow the organization to acquire more accurate quantities of material across the board, which will set the organization up for success and improve profitability.

Table of Contents

Contents

List of Tables and Figures

Preface

The major challenge that procurement teams face is determining how much of a product to buy and when to buy it. Purchasing agents are tasked with acquiring enough quantity to fill orders without buying too much inventory that inflates warehousing costs or doesn't turn over. Corporate teams that purchase for multiple locations have the additional challenge of buying the right quantities for each specific site, which generally vary tremendously from one to the next. Other factors involved in this process include assessing how much of a product will be used in a specific time period, determining lead times to get the material delivered, and evaluating internal transfers between sites to optimize order sizes. Although very difficult, this process of acquiring product, cost effectively, is crucial and has a direct impact on an organization's profitability.

This project will use predictive modeling techniques to forecast usage at the product level to mitigate the problems outlined above. The project will utilize the Cross-Industry Standard Process for Data Mining (CRISP-DM) to organize and deliver results. This is a project methodology that helps ensure thorough analysis and high quality output of data science projects, and is used as a common and standard outline for analytics groups to use for all projects. CRISP-DM is defined by six major phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. Each phase flows in a logical manner, but the process is designed to be iterative, as many data projects tend to naturally flow. This project will utilize CRISP-DM by defining the business problem to be addressed, detailing the data used and how it was acquired, explaining the steps taken to prepare the data, experimenting with different modeling techniques to optimize prediction accuracy, evaluating each model explored and determining the accuracy on test data, and finally outlining the deployment strategy into a production system. The R script will be shared in the Appendix for detailed steps taken, and will be referenced throughout the project.

Introduction

The specific problem faced by the purchasing team at this organization is that there are currently no tools available to signal future needs of material. The current method used by purchasing agents, and the only method available, is to look at a table of historic usage for each part over the past 36 months and make a best guest at what the organization will need in the coming months. This method has been used for many years and is the standard, but the team continually struggles with anticipating end customer needs, lengthy time of analysis, and decisions on how much quantity to buy.

The aim of this project is to analyze historic usage and build a reliable predictive model with the end goal of providing the purchasing team a tool that will forecast monthly usage of each product for the next 6 months. Additional objectives include identifying a seasonality index to signal up and down months as well as providing forecasts for monthly transfers between branch locations. Monthly usage forecasts, monthly transfer forecasts, and seasonality indexes for each product together will help the purchasing agents acquire material much more efficiently and move material throughout the organization at a lower cost. The benefit will be more accurate quantities of material purchased throughout the year, which will result in more orders fulfilled and less warehousing costs accumulated.

This project has been approved and is sponsored by both the VP of Purchasing and the VP of IT, with the major stakeholders and beneficiaries of a successful outcome being the VP of Purchasing and his team of Purchasing Agents. The business and data teams have defined the scope of the project in this initial stage as one product group (tonnage group 010) for one branch (Branch 1) of the organization. If, after adequate testing by the purchasing team, the results are positive, the scope will expand to additional product groups and additional branches.

The business has defined a successful project as a tool that shows a seasonality index for each specific product and can forecast usage for the next 6 months. The forecasts must produce more accurate quantities than the incumbent method in testing, as compared to actual usage; and the tool must be deployed in a way that is easily accessible by each member of the team. If these criteria are met, the project will be deemed a success and will be approved to progress to the next phase.

CRISP – DM Methodology Phases

Business Understanding

The purchasing team is not currently equipped with the tools to adequately anticipate future customer needs. The incumbent method of determining quantities of material to purchase include looking at a table of historic usage, for each part at each branch, over the past 36 months, and making a best guess. This often results in too little quantities acquired to fulfill orders, which results in loss of revenue; or too much quantity purchased, which results in material sitting in the warehouse and accumulating inventory costs. Both of these scenarios negatively impact the bottom line of the organization and are focus areas for improvement for the purchasing team.

The aim of this project is to provide a tool that will mitigate the problems defined above. By providing seasonality indexes, 6-month usage forecasts, and 6-month transfer forecasts for each part, the purchasing agents will spend significantly less time analyzing historic data and will be equipped with tools to acquire more accurate quantities of product to fulfill future customer needs. This project will be successful if the output is more accurate than using the incumbent method, supported by analysis of actual usage data held-out for testing. Additionally, the final tool must be deployed in a way that is easily accessible to the team in the production ERP environment.

Data Understanding

The data set used by this project is time series data and is fairly straightforward. The data consists of monthly quantity used for 1,306 parts. All the parts are from one product group (tonnage group 010) for one branch in the organization (Branch 1). Each part includes 42 months of history, starting January 2016 through June 2019. Data was acquired from the production data warehouse and has already been cleaned and verified as accurate.

The data is sufficient to accomplish what the Business Understanding section above defines. In fact, the data set is the same historic data that the purchasing team is currently using with the incumbent method. Until recently, the usage data was a rolling 36 months, so January 2016 is as far back as the data goes. However, as testing and time goes on, the data captured will continue to grow and additional patterns may come to light with more history to analyze.

Data Preparation

Preparation and cleansing steps for this monthly time series data were minimal. Data was cleansed by eliminating all “NA” and “<NULL>” values and replaced with values of 0. Parts that aren’t stocked in the warehouse, parts that didn’t have history in at least 1 of the last 12 months, and parts with total usage of less than 10 were removed. These parts were deemed as one-off buys with no need to be forecasted. Column names were re-organized in a way that allows the data to be analyzed using a hierarchical time series forecasting technique (see Appendix A). Finally, data for the year 2019 was held out as the Test data set, which is used in the Modeling and Evaluation phases for testing the accuracy of the models.

Modeling

Initial modeling consisted of a hierarchical time series forecasting function in R Studio (See Appendix A). The hts() function allows data to be assessed in aggregate across the organization; but for the purposes of analyzing the data for only one branch in this first phase, the data was analyzed at the base level. This function allowed for all parts in the time series data set to be analyzed using four different techniques very easily: Optimal Reconciliation, Random Walk, ETS, and ARIMA forecasting methods. Each of the four models was designed to produce 6 month usage forecasts starting January through June 2019.

Using "For Loops" further modeling was performed on the time series data using more traditional forecast functions in R Studio (See Appendix A). The additional modeling techniques consisted of STLF(), TBATS(), ARIMA(), ETS(), and a simple Monthly Avg method. This last method is the Average Monthly Usage for each product added to the Seasonality value identified by the stl() function. Since the Random Walk method in the initial modeling phase above produced competitive forecasts as it relates to RMSE (shown in the Evaluation Phase below), this Monthly Average method was created as a middle ground between a simple naive method and the more sophisticated methods. Each of these five models was also designed to produce 6 months of usage forecasts starting January through June 2019.

In addition to being a component to a new potential forecasting method, the Seasonality Index was a requirement of this project. Using the stl() function, each product's usage was analyzed for a seasonality component. The Seasonality Index identifies the percentage that each month contributes to the year's total usage, on average. Essentially, the monthly average usage for each part over the 36 month training period was identified, and the Seasonality Index is the multiplier for each month. The output from the Seasonality Analysis is shown in Figure 1:

Figure 1: Seasonality Index - by Part (Only six parts shown)

	AA0000_B1_US	AA0000_B1_TR	AA0001_B1_US	AA0001_B1_TR	AA0009_B1_US	AA0009_B1_TR
Jan	109.9%	30.2%	68.8%	291.7%	36.1%	137.6%
Feb	85.7%	17.7%	79.3%	71.6%	69.5%	-11.4%
Mar	197.6%	194.5%	140.7%	118.2%	47.2%	182.5%
Apr	74.0%	359.1%	114.9%	12.1%	9.0%	183.2%
May	83.8%	154.8%	85.1%	153.6%	193.4%	164.9%
Jun	128.8%	8.6%	134.7%	-2.7%	38.4%	-5.6%
Jul	131.9%	-2.5%	98.0%	-6.7%	296.5%	14.6%
Aug	58.2%	7.1%	92.5%	293.8%	44.1%	-0.1%
Sep	30.9%	5.0%	181.1%	308.6%	61.9%	404.3%
Oct	123.5%	163.3%	170.3%	-20.6%	7.4%	11.4%
Nov	86.1%	-37.0%	22.3%	-6.9%	215.2%	94.7%
Dec	89.9%	299.1%	12.2%	-13.0%	181.2%	24.0%

As can be seen above, each part has a Seasonality index for each month. Assessing at the part level was and will continue to be crucial, as each part has spikes and troughs at different times of the year. It is also likely that the seasonality is different for the same part, at different branches. This is an important concept for the purchasing team to understand during the buying process, and will affect the forecasts provided by the Monthly Average Forecast method below.

The output from the forecasting methods described previously is a monthly forecast for each part. The results from each forecasting method were exported from R to separate tabs in the same Excel Spreadsheet. Each technique provided forecasts for each part over the 6 month test period in 2019. An example of predictions from the Monthly Avg Forecast method is shown in Figure 2 (note the other eight forecast methods shown at the bottom of Figure 2). The structure of this output in Figure 2 is slightly different than the Seasonality Index in Figure 1, but both are designed in a way to be ingested by the production ERP system in later phases of this project.

Figure 2: Monthly Avg Forecasts - by Part (Only six parts shown)

	Jan_Forecast	Feb_Forecast	Mar_Forecast	Apr_Forecast	May_Forecast
AA0000_B1_US	986.55	769.07	1773.93	664.07	752.54
AA0000_B1_TR	41.11	24.09	264.74	488.94	210.8
AA0001_B1_US	140.37	161.72	287.07	234.37	173.66
AA0001_B1_TR	122.53	30.09	49.66	5.09	64.52
AA0009_B1_US	36.36	69.93	47.51	9.1	194.7
AA0009_B1_TR	57.79	-4.79	76.63	76.94	69.24

OptimalReconciliation_hts / RandomWalk_hts / ETS_hts / ARIMA_hts / STLF_Pred / TBATS_Pred / ARIMA_Pred / ETS_Pred / **MonthlyAvg**

From the predictions shown in Figure 2 for the Monthly Average Forecast method, you can see that they are directly influenced by the Seasonality Index identified in Figure 1. Using Part AA0000_B1_US as an example; the Avg Monthly usage for that part over the 36 month Train period was 897 units. January received a multiplier of 109.9%, which results in a January prediction of 986 units (using rounded numbers). The other methods use more complex calculations and show different predictions. The benefit of exporting all predictions is that, depending on the results from the RMSE comparison in the Evaluation phase, any forecast method is ready to be used. It’s also helpful to see the predictions, as viewing this much data is a challenge in R.

Evaluation

The accuracy for each forecast method was assessed against the 2019 Test data that was held-out in the Data Preparation phase above using Root Mean Squared Error (RMSE). This accuracy measure allows for models of different types to be compared; a lower RMSE indicates less error and a more accurate model. Figure 3 shows the results of the accuracy() function on the initial 4 hts() modeling techniques:

Figure 3: hts() Function – RMSE results

```
> accuracy(FC_AGG$OptRec,UsageBTS_Test)
                 ME     RMSE      MAE MPE MAPE      ACF1 Theil's U
Test set -6.930543 485.6407 150.2037 NaN  Inf 0.3740568       NaN
> accuracy(FC_AGG$RandomWalk,UsageBTS_Test)
               ME     RMSE     MAE MPE MAPE      ACF1 Theil's U
Test set 17.53786 481.7267 135.386 NaN  Inf 0.4026573       NaN
> accuracy(FC_AGG$ETS,UsageBTS_Test)
                 ME    RMSE      MAE MPE MAPE     ACF1 Theil's U
Test set -17.44267 490.306 154.1693 NaN  Inf 0.379344       NaN
> accuracy(FC_AGG$ARIMA,UsageBTS_Test)
                 ME    RMSE      MAE MPE MAPE      ACF1 Theil's U
Test set -15.38686 510.469 156.9954 NaN  Inf 0.3890409       NaN
```

The results above are interesting, as all models have a fairly similar RMSE value. In fact, the Random Walk method, which simply uses the most recent value as the forecast for the next 6 months, outperformed the more sophisticated techniques by a small margin with an RMSE of 481. The ETS model performed worst with an RMSE of 510.

The more traditional forecast methods were also assessed using RMSE. The RMSE of these techniques were all lower (even for the same forecasting methods used by hts() at the base level of aggregation), which was interesting. Figure 4 below shows the results of the accuracy() function on the 5 traditional modeling techniques:

Figure 4: Tradition Methods – RMSE results

```
> #traditional methods
> accuracy(STLF_Pred, UsageBTS_Test)
                ME     RMSE      MAE MPE MAPE       ACF1 Theil's U
Test set -26.36608 378.3544 125.6582 NaN  Inf 0.03724333       NaN
> accuracy(TBATS_pred, UsageBTS_Test)
                ME     RMSE      MAE MPE MAPE       ACF1 Theil's U
Test set -15.06669 358.4429 109.5317 NaN  Inf 0.08504961       NaN
> accuracy(ARIMA_pred, UsageBTS_Test)
                ME     RMSE      MAE MPE MAPE    ACF1 Theil's U
Test set -15.38685 370.3564 111.3318 NaN  Inf 0.13397       NaN
> accuracy(ETS_pred, UsageBTS_Test)
                ME     RMSE    MAE MPE MAPE      ACF1 Theil's U
Test set -39.61136 390.0232 135.71 NaN  Inf 0.1126653       NaN
> accuracy(MonthlyAvg_Pred, UsageBTS_Test)
                ME     RMSE      MAE MPE MAPE      ACF1 Theil's U
Test set -22.91224 380.4156 122.4983 NaN  Inf 0.0330228       NaN
```

All of these techniques performed better than the hts() function analyzed above. The most accurate model was TBATS with an RMSE of 358, while the ETS model performed worst, again, with an RMSE of 390. The simple Monthly Average forecasting method performed competitively with an RMSE of 380. It's interesting that this simple/naïve model performed just as good, if not better, than many of the other more sophisticated forecasting techniques explored. A case could be made for any of the model types; but for the sake of simplicity in model understanding and deployment, the project team decided to move forward with the Monthly Average Forecasting method during this first phase of the project.

After comparing the performance of each forecasting model and deciding on the forecast method to move forward with, the Monthly Average Forecasting method was compared to the incumbent method to assess which was more accurate. The incumbent method of purchasing relies heavily on looking at the table of previous month's quantity used and making a best guess on how much quantity to buy in the coming months. In order to assess the incumbent method, actual Quantity Ordered data from the production system was brought in to R (See Appendix A). The data was aggregated by month and formatted into Time Series data, similar to the monthly quantity data used in generating the forecasts throughout this project. The Monthly Average Forecast method predictions and the Incumbent method using actual quantity ordered were then assessed for accuracy using RMSE, and the results are shown in Figure 5:

Figure 5: RMSE Compared to Actual Usage

```
> accuracy(MA_Pred_Aggregate, Usage_Aggregate)
                ME     RMSE      MAE MPE MAPE       ACF1 Theil's U
Test set -28.81679 380.4542 113.6226 NaN  Inf 0.02065736       NaN
> accuracy(QtyOrd_Aggregate, Usage_Aggregate)
               ME     RMSE      MAE  MPE MAPE       ACF1 Theil's U
Test set 3.227451 609.2389 149.8291 -Inf  Inf -0.1187904         0
>
```

The results in Figure 5 are promising, as the accuracy of the predictions showed significantly higher accuracy than the incumbent method. The RMSE value for the forecasts was 380, while the RMSE of the actual quantity ordered data set was 609. In fact, all forecasting techniques performed notably better than the incumbent method in terms of RMSE, which indicates discernable benefit to using forecasts going forward.

An additional step was taken in the Evaluation phase; this consisted of bringing in data for an additional Branch (Branch 3, Product group 010) to validate the results found thus far. This second round of testing support the above findings; the traditional methods perform better than the hts() methods, the Monthly Average Prediction method performs competitively/best compared to the more sophisticated models, and the accuracy of the forecasts outperform the incumbent method by a significant margin.

From the business perspective, the results of this analysis are encouraging. The project has identified a Seasonality Index, and more importantly, a predictive model that forecasts monthly usage more accurately than the incumbent method of ordering products. The output has been exported and deployed in a way that can be consumed by the Purchasing team and provides an opportunity for more efficient buying.

Deployment

The output of this project is monthly usage predictions for all parts in product group 010 for Branch 1. After eliminating some of the products that forecasts were not needed for, like products with very sparse history and products with no history in the last year, the total number of 6-month forecasts was 899. Each forecast suggests how much of each product will be used, and thus suggests to the purchasing team how much of a quantity to buy. The forecasts are monthly, even though the purchasing agents sometimes are required to buy 3-months' worth at a time. So, in order to best use this tool, the monthly forecasts may need to be added up and purchased in bulk. If used correctly, the tool will help the purchasing team acquire more accurate quantities of product to balance being able to fill orders and managing inventory costs.

The first phase of deployment will consist of exporting the prediction results to the Monthly Average Forecasting model as well as the Seasonality Index to Excel. The Excel spreadsheet will then be distributed to the Purchasing agents in charge of Product Group 010 for Branch 1, each month. This will provide the team with an easily accessible tool to reference during the buying process while the tool gets tested and validated.

Once the model goes through initial testing and validation, the seasonality index and forecast data will be imported into a User-Defined table in the production ERP system. The UD Table will consist of the Branch, Part, Seasonality Index for each month, and the Monthly Avg Model's forecast for a rolling 6 month period. In terms of data governance, the data will be updated monthly. This is a good balance of up-to-date forecasts and effort required to update. By doing so, the prediction model and seasonality index will continue to adjust to changing market conditions and will reflect those changes in the forecasts provided.

The data will be deployed into the Purchasing Teams Usage History Dashboard. This is the tool the Purchasing agents use to identify historic quantities used. This data will be deployed into a table within this dashboard to provide forward looking forecasts as well as upcoming identified seasonality indexes. This will allow the purchasing team to, not only see what was used in the past, but to see forecasts of what will likely be used in their planning of future buys.

The team will need to be trained on how to put this data to good use, and not to rely on the predictions as truth. Since there is not really any forecasting being done currently, this could be the biggest hurdle. The teams needs to understand that this is just one tool, and should be used alongside the other tools used to make effective purchases. The tool won't know company initiatives or current market conditions, so expert knowledge will be a key input that only the purchase agent can provide. Even so, the results are promising and provide significant opportunity to take advantage of and potential to improve the success of the organization.

Appendix – R Code

```
library(forecast)
library(hts)
library(fpp2)
library(readxl)
library(xlsx)
library(data.table)
library(tsintermittent)

###Update Label for File Naming
Label <- "Branch1_010"

###Get Data into R
ImportFile <-  paste("C:/Users/mverwijst/Documents/4. Elmhurst - Data Science/6. Summer2019/ProjectData_CrossTab_"
                   , Label,".xlsx", sep="")
ProjectData <-read_excel(ImportFile, sheet="Sheet1")
UsageData <- ProjectData

###Remove Period Column
x <- ncol(UsageData)
UsageData <- UsageData[,2:x]

###Only select Products with History in at least 1 of last 12 months and total usage of > 10 units
keep <- (colMeans(UsageData[25:36,]>0) >= 0.0833)
UsageData_trimmed <- UsageData[, keep, drop = FALSE]
UsageData <- UsageData_trimmed
UsageData <- UsageData[, colSums(UsageData) > 10]

###Re-organize column headers for Correct Aggregation needed by hts()
h1 <- substring(names(UsageData),7)
h2 <- substring(names(UsageData),0,5)
names(UsageData) <- paste(h1,h2, sep="_")

###Split data into Train and Test Data Sets
train <- UsageData[1:36,]
test <- UsageData[37:41,]
```

```
###Create the Aggregation Structure using hts()
UsageBTS <- ts(train, frequency = 12, start=c(2016,1), end = c(2018,12))
usageHTS <- hts(UsageBTS, characters = c(6,6))
UsageBTS_Test <- ts(test, frequency = 12, start=c(2019,1), end = c(2019,5))
usageHTS_Test <- hts(UsageBTS, characters = c(6,6))

###Optimal Reconciliation Forecast Method using hts()
FC_OptRec <- forecast(usageHTS, h=5, algorithms="cg", keep.fitted=TRUE, keep.resid=TRUE,
parallel=TRUE, num.cores=4)
FC_AGG_OptRec <- aggts(FC_OptRec, levels=2:2)

###Random Walk Forecast Method using hts()
FC_RandomWalk <- forecast(usageHTS, method="bu", fmethod="rw", h=5, keep.fitted=TRUE,
keep.resid=TRUE, parallel=TRUE, num.cores=4)
FC_AGG_RandomWalk <- aggts(FC_RandomWalk, levels=2:2)

##ETS Forecast Method using hts()
FC_ETS <- forecast(usageHTS, method="bu", fmethod="ets", h=5, keep.fitted=TRUE, keep.resid=TRUE,
parallel=TRUE, num.cores=4)
FC_AGG_ETS <- aggts(FC_ETS, levels=2:2)

#ARIMA Forecast Method using hts()
FC_ARIMA <- forecast(usageHTS, method="bu", fmethod="arima", h=5, keep.fitted=TRUE,
keep.resid=TRUE, parallel=TRUE, num.cores=4)
FC_AGG_ARIMA <- aggts(FC_ARIMA, levels=2:2)

###Convert Bottom Level hts aggregation to Data Frames
DF_OptRec_BL <- as.data.frame(aggts(FC_OptRec, levels = 2:2))
DF_RandomWalk_BL <- as.data.frame(aggts(FC_RandomWalk, levels = 2:2))
DF_ETS_BL <- as.data.frame(aggts(FC_ETS, levels = 2:2))
DF_ARIMA_BL <- as.data.frame(aggts(FC_ARIMA, levels = 2:2))

###Convert Data Frames to Tables for proper Exporting to Excel
DF_OptRec_BL = t(DF_OptRec_BL)
DF_RandomWalk_BL = t(DF_RandomWalk_BL)
DF_ETS_BL = t(DF_ETS_BL)
DF_ARIMA_BL = t(DF_ARIMA_BL)
```

```
###Change Column Names for Tables
colnames(DF_OptRec_BL) <-
c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(DF_RandomWalk_BL) <-
c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(DF_ETS_BL) <- c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(DF_ARIMA_BL) <-
c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')

###Round Forecast Results to 2 digits
DF_OptRec_BL <- round(DF_OptRec_BL, digits = 2)
DF_RandomWalk_BL <- round(DF_RandomWalk_BL, digits = 2)
DF_ETS_BL <- round(DF_ETS_BL, digits = 2)
DF_ARIMA_BL <- round(DF_ARIMA_BL, digits = 2)

###Calculate Accuracy of each Forecast Method using hts() on TEST data
###Determine best performing method as it relates to RMSE (MAPE=Inf because of zero values)
accuracy(FC_AGG_OptRec,UsageBTS_Test)
accuracy(FC_AGG_RandomWalk,UsageBTS_Test)
accuracy(FC_AGG_ETS,UsageBTS_Test)
accuracy(FC_AGG_ARIMA,UsageBTS_Test)


###EXPLORE TRADITIONAL FORECAST METHODS

###Create List of Monthly Seasonality using STL() for all columns
SeasonalList <- list()
for(i in 1:ncol(train)){
  STLdecomp <- stl(UsageBTS[ ,i],s.window="periodic")
  SeasonalList[[i]] <- STLdecomp$time.series[1:12,1]
}
###Create List of Forecasts using STLF() for all columns
STLF_List <- list()
  for(i in 1:ncol(train)){
      STLForecast <- stlf(UsageBTS[ ,i],s.window="periodic", h=5)
      STLF_List[[i]] <- STLForecast$mean
  }
###Create List of Forecasts using TBATS() for all columns
TBATS_List <- list()
for(i in 1:ncol(train)){
  TBATSForecast <- forecast(tbats(UsageBTS[ ,i], parallel=TRUE, num.cores=4), h=5)
  TBATS_List[[i]] <- TBATSForecast$mean}
```

```r
###Create List of Forecasts using auto.arima() for all columns
ARIMA_List <- list()
for(i in 1:ncol(train)){
  ArimaForecast <- forecast(auto.arima(UsageBTS[ ,i]), h=5)
  ARIMA_List[[i]] <- ArimaForecast$mean
}

###Create List of Forecasts using ETS for all columns
ETS_List <- list()
for(i in 1:ncol(train)){
  ETSForecast <- stlf(UsageBTS[ ,i], etsmodel="ZZZ", damped=TRUE, h=5)
  ETS_List[[i]] <- ETSForecast$mean
}

###Convert Lists to Dataframes and round values
SeasonalData <- data.frame(t(sapply(SeasonalList,c)))
SeasonalData <- round(t(SeasonalData), digits = 2)
STLF_pred <- data.frame(t(sapply(STLF_List,c)))
STLF_pred <- round(t(STLF_pred), digits = 2)
TBATS_pred <- data.frame(t(sapply(TBATS_List,c)))
TBATS_pred <- round(t(TBATS_pred), digits = 2)
ARIMA_pred <- data.frame(t(sapply(ARIMA_List,c)))
ARIMA_pred <- round(t(ARIMA_pred), digits = 2)
ETS_pred <- data.frame(t(sapply(ETS_List,c)))
ETS_pred <- round(t(ETS_pred), digits = 2)

###Rename columns and rows
colnames(SeasonalData) <- colnames(train)
rownames(SeasonalData) <- c('Jan', 'Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec')
colnames(STLF_pred) <- colnames(train)
rownames(STLF_pred) <- c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(TBATS_pred) <- colnames(train)
rownames(TBATS_pred) <-
c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(ARIMA_pred) <- colnames(train)
rownames(ARIMA_pred) <-
c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(ETS_pred) <- colnames(train)
rownames(ETS_pred) <- c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
```

```
###SEASONALITY ANALYSIS

###Avg Monthly Usage
MonthlyAvg <- apply(train,2,function(x){(mean(x))})
MonthlyAvg <- t(MonthlyAvg)

###Bind Avg Monthly Usage to bottom of SeasonalData
pred <- rbind(SeasonalData,MonthlyAvg)

###Loop through each column, add each row to the Monthly Average row (13)
for(i in 1:ncol(pred)){
  pred[1,i] <- pred[1,i]+ pred[13,i]
  pred[2,i] <- pred[2,i]+ pred[13,i]
  pred[3,i] <- pred[3,i]+ pred[13,i]
  pred[4,i] <- pred[4,i]+ pred[13,i]
  pred[5,i] <- pred[5,i]+ pred[13,i]
  pred[6,i] <- pred[6,i]+ pred[13,i]
  pred[7,i] <- pred[7,i]+ pred[13,i]
  pred[8,i] <- pred[8,i]+ pred[13,i]
  pred[9,i] <- pred[9,i]+ pred[13,i]
  pred[10,i] <- pred[10,i]+ pred[13,i]
  pred[11,i] <- pred[11,i]+ pred[13,i]
  pred[12,i] <- pred[12,i]+ pred[13,i]
  }
MonthlyAvg_Pred <- round(pred, digits = 2)

Seasonality_Index <- MonthlyAvg_Pred
for(i in 1:ncol(Seasonality_Index)){
  Seasonality_Index[1,i] <- Seasonality_Index[1,i] / Seasonality_Index[13,i]
  Seasonality_Index[2,i] <- Seasonality_Index[2,i] / Seasonality_Index[13,i]
  Seasonality_Index[3,i] <- Seasonality_Index[3,i] / Seasonality_Index[13,i]
  Seasonality_Index[4,i] <- Seasonality_Index[4,i] / Seasonality_Index[13,i]
  Seasonality_Index[5,i] <- Seasonality_Index[5,i] / Seasonality_Index[13,i]
  Seasonality_Index[6,i] <- Seasonality_Index[6,i] / Seasonality_Index[13,i]
  Seasonality_Index[7,i] <- Seasonality_Index[7,i] / Seasonality_Index[13,i]
  Seasonality_Index[8,i] <- Seasonality_Index[8,i] / Seasonality_Index[13,i]
  Seasonality_Index[9,i] <- Seasonality_Index[9,i] / Seasonality_Index[13,i]
  Seasonality_Index[10,i] <- Seasonality_Index[10,i] / Seasonality_Index[13,i]
  Seasonality_Index[11,i] <- Seasonality_Index[11,i] / Seasonality_Index[13,i]
  Seasonality_Index[12,i] <-Seasonality_Index[12,i] / Seasonality_Index[13,i]
}
Seasonality_Index <- round(Seasonality_Index, digits = 4)
```

```
###Calculate Accuracy of each Forecast Method on TEST data
###Determine best performing method as it relates to RMSE (MAPE=Inf because of zero values)
MonthlyAvg_Pred <- ts(MonthlyAvg_Pred[1:12,], frequency = 12, start=c(2019,1), end = c(2019,12))
STLF_Pred <- ts(STLF_pred, frequency = 12, start=c(2019,1), end = c(2019,12))
TBATS_pred <- ts(TBATS_pred, frequency = 12, start=c(2019,1), end = c(2019,12))
ARIMA_pred <- ts(ARIMA_pred, frequency = 12, start=c(2019,1), end = c(2019,12))
ETS_pred <- ts(ETS_pred, frequency = 12, start=c(2019,1), end = c(2019,12))

#hts() methods
accuracy(FC_AGG_OptRec,UsageBTS_Test)
accuracy(FC_AGG_RandomWalk,UsageBTS_Test)
accuracy(FC_AGG_ETS,UsageBTS_Test)
accuracy(FC_AGG_ARIMA,UsageBTS_Test)

#traditional methods
accuracy(STLF_Pred, UsageBTS_Test)
accuracy(TBATS_pred, UsageBTS_Test)
accuracy(ARIMA_pred, UsageBTS_Test)
accuracy(ETS_pred, UsageBTS_Test)
accuracy(MonthlyAvg_Pred, UsageBTS_Test)

###Convert Data Frames to Tables for proper Exporting to Excel
MonthlyAvg_Pred1 <- t(MonthlyAvg_Pred)
STLF_pred1 <- t(STLF_Pred)
TBATS_pred1 <- t(TBATS_pred)
ARIMA_pred1 <- t(ARIMA_pred)
ETS_pred1 <- t(ETS_pred)

MonthlyAvg_Pred1 <- MonthlyAvg_Pred1[,1:5]
STLF_pred1 <- STLF_pred1[,1:5]
TBATS_pred1 <- TBATS_pred1[,1:5]
ARIMA_pred1 <- ARIMA_pred1[,1:5]
ETS_pred1 <- ETS_pred1[,1:5]

colnames(MonthlyAvg_Pred1) <-
c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(STLF_pred1) <- c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(TBATS_pred1) <-
c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(ARIMA_pred1) <-
c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
colnames(ETS_pred1) <- c('Jan_Forecast','Feb_Forecast','Mar_Forecast','Apr_Forecast','May_Forecast')
```

```
###Define File Paths
Fc_Path <- paste("C:/Users/mverwijst/Documents/4. Elmhurst - Data Science/6. Summer2019/PredictionResults_", Label,".xlsx", sep="")
Seas_Path <- paste("C:/Users/mverwijst/Documents/4. Elmhurst - Data Science/6. Summer2019/Seasonality_", Label,".xlsx", sep="")

###Export All Forecasts to Separate Sheets in same Excel Workbook
write.xlsx(DF_OptRec_BL, Fc_Path, sheetName="OptimalReconciliation_hts", append=FALSE)
write.xlsx(DF_RandomWalk_BL, Fc_Path, sheetName="RandomWalk_hts", append=TRUE)
write.xlsx(DF_ETS_BL, file=Fc_Path, row.names=TRUE,
        sheetName="ETS_hts", append=TRUE)
write.xlsx(DF_ARIMA_BL, file=Fc_Path, row.names=TRUE,
        sheetName="ARIMA_hts", append=TRUE)
write.xlsx(MonthlyAvg_Pred1, file=Fc_Path, row.names=TRUE,
        sheetName="MonthlyAvg_Pred", append=TRUE)
write.xlsx(STLF_pred1, file=Fc_Path, row.names=TRUE,
        sheetName="STLF_Pred", append=TRUE)
write.xlsx(TBATS_pred1, file=Fc_Path, row.names=TRUE,
        sheetName="TBATS_Pred", append=TRUE)
write.xlsx(ARIMA_pred1, file=Fc_Path, row.names=TRUE,
        sheetName="ARIMA_Pred", append=TRUE)
write.xlsx(ETS_pred1, file=Fc_Path, row.names=TRUE,
        sheetName="ETS_Pred", append=TRUE)

###Write the Seasonality Index, actual seasonal values, monthly avg, and prediction to Excel
write.xlsx(Seasonality_Index, file=Seas_Path, row.names=TRUE,
        sheetName="Seasonality_Percentage")
write.xlsx(SeasonalData, file=Seas_Path, row.names=TRUE,
        sheetName="Seasonality_diffMean", append=TRUE)
write.xlsx(MonthlyAvg, file=Seas_Path, row.names=TRUE,
        sheetName="MonthlyAvg", append=TRUE)
write.xlsx(MonthlyAvg_Pred, file=Seas_Path, row.names=TRUE,
        sheetName="MonthlyAvg_Pred", append=TRUE)
```

```
###Compare Simple Seasonal Method to Incumbent Method

###Get Aggregated QtyOrd Data into R
ImportFile1 <-  paste("C:/Users/mverwijst/Documents/4. Elmhurst - Data Science/6. Summer2019/ProjectData_CrossTab_"
                  , Label,".xlsx", sep="")
QtyOrd_Aggregate <-read_excel(ImportFile1, sheet="Sheet2")
y <- ncol(QtyOrd_Aggregate)
QtyOrd_Aggregate <- QtyOrd_Aggregate[,2:y]
h3 <- substring(names(QtyOrd_Aggregate),8)
names(QtyOrd_Aggregate) <- paste(h3)
#View(QtyOrd_Aggregate)

###Aggregate Actual Usage Data at the Part Level
UsageTest <- test
U_BTS <- ts(UsageTest, frequency = 12, start=c(2019,1), end = c(2019,5))
U_HTS <- hts(U_BTS, characters = c(6,6))
Usage_Aggregate <- aggts(U_HTS, levels=1:1)
#View(Usage_Aggregate)

###Aggregate Monthly Avg Prediction at the Part Level
MA_Pred_Aggregate <- MonthlyAvg_Pred
MA_BTS <- ts(MA_Pred_Aggregate, frequency = 12, start=c(2019,1), end = c(2019,5))
MA_HTS <- hts(MA_BTS, characters = c(6,6))
MA_Pred_Aggregate <- aggts(MA_HTS, levels=1:1)
#View(MA_Pred_Aggregate)

###Get only matching columns... exclude new parts
ncol(Usage_Aggregate)
ncol(QtyOrd_Aggregate)
ncol(MA_Pred_Aggregate)

column_matches <- intersect(colnames(Usage_Aggregate),colnames(QtyOrd_Aggregate))
column_matches <- intersect(column_matches,colnames(MA_Pred_Aggregate))

Usage_Aggregate <- Usage_Aggregate[ , column_matches]
QtyOrd_Aggregate <- QtyOrd_Aggregate[ , column_matches]
MA_Pred_Aggregate <- MA_Pred_Aggregate[ , column_matches]
ncol(Usage_Aggregate)
ncol(QtyOrd_Aggregate)
ncol(MA_Pred_Aggregate)
```

```
###Compute RMSE of MonthAvg Prediction and Incumbent method using accuracy()
MA_Pred_Aggregate <- ts(MA_Pred_Aggregate[1:5,], frequency = 12, start=c(2019,1), end = c(2019,5))
QtyOrd_Aggregate <- ts(QtyOrd_Aggregate[1:5,], frequency = 12, start=c(2019,1), end = c(2019,5))
Usage_Aggregate <- ts(Usage_Aggregate[1:5,], frequency = 12, start=c(2019,1), end = c(2019,5))

accuracy(MA_Pred_Aggregate, Usage_Aggregate)
accuracy(QtyOrd_Aggregate, Usage_Aggregate)

###Sum Columns to get totals by part
MA_Pred_Aggregate_colsum <- colSums(MA_Pred_Aggregate)
QtyOrd_Aggregate_colsum <- colSums(QtyOrd_Aggregate)
Usage_Aggregate_colsum <- colSums(Usage_Aggregate)

###Calculate the difference of Prediction-Usage and Incumbent method-Usage
MA_Pred_Aggregate_DIFF <- MA_Pred_Aggregate_colsum - Usage_Aggregate_colsum
QtyOrd_Aggregate_DIFF <- QtyOrd_Aggregate_colsum - Usage_Aggregate_colsum

###Calculate the total Inventory Impact of Prediction and Incumbent methods
MA_Pred_Aggregate_InvImpact <- rowSums(t(MA_Pred_Aggregate_DIFF))
QtyOrd_Aggregate_InvImpact <- rowSums(t(QtyOrd_Aggregate_DIFF))
MA_Pred_Aggregate_InvImpact
QtyOrd_Aggregate_InvImpact
```