ISEN 615 Project Report

Deep Patel	130004781
Saurabh Jain	527002462
Smith Shah	227008337

Texas A&M University

ISEN 615 Project Report

The report is based on forecasting for Arconic Fastening Systems and Rings and Automation including optimization of truck loads for VTI Industries. For Arconic Fastening Systems and Rings, we have done data exploration, data cleaning and data analysis, to create two long term forecast models and two short term forecast models. These models also incorporate the open orders to be fulfilled based on Arconic history data and open order data. In case of VTI Industries, we did data cleaning and tried to automate the existing file to automate the safety stock and reorder point values. These values are used to optimize the truck load.

Arconic Fastening Systems and Rings

Assumptions

Some of the underlying assumptions used are:

1. Item and Item group demand is directly related to the truck market rate.
2. Open orders must be fulfilled.
3. Unfulfilled demands are backordered.
4. No penalty cost is levied.

Idea Flow

To achieve most accurate forecasting we did data exploration, made an excel sheet to understand data. For the correlation we used pivot tables which can help you to identify the correlation between different customer and the products. Then we tried to use linear regression but due to lack of seasonality and huge amount of residuals we dropped the idea. This dropping of linear regression was concreted when we tried the plotting of regression line on all item groups. Once the models failed we thought to look for other models. To incorporate these models we used R programming. The different models we thought about were Exponential Time Smoothing, ARIMA, MAPA, TBATS, Holts Winter Method, Theta and linear regression. As the time series is irregular and there was no seasonality in some of the models, we dropped MAPA and Holts Winter method. Using the remaining method, we forecasted our models for a set of training data to check for the error in forecasting. Using AIC, BIC and forecast values we tried to find best fit of the model.

Data Exploration using Excel.

For data exploration, we used pivot chart to understand correlation between customer and their orders. These tables helped us to fit a linear regression line to the trend and understand the type of forecast too. More details can be found in appendix 1A

Data Exploration using R.

The linear regression model did not predict well and to validate this output, we used R programming to fit a linear regression line to a set of specific items and item groups. We got similar results and following those results we took a decision to drop linear regression method for forecasting. Theoretically the forecast do not have any trend or seasonality for each and every item or item group and there is huge amount of variation in data points. Following this it is not advisable to use linear regression to such a model. More information could be found in Appendix 1B

Data Cleaning using Excel and R.

The first step was to clean the irregular time series which had huge amount of randomness. We cleaned the Excel file to remove all Zeroes and Negative orders. To address this we thought to use TS clean but due to irregularity in time series, we need to use the 2 Standard Deviation as our control limit. Once the data was under the control limits, it was to be sorted according to order date. Then we plotted the data to ensure that the output matches our requirement. More information and R code in Appendix 1C.

Data Analysis using R.

The second step was to analyze different models for forecasting. For this analysis we used AIC, BIC, Forecasted value, ME (Mean), MAPE% (Mean Absolute Percentage Error), MAE (Mean Absolute Error) as the parameters to compare best fitting model. To get the above parameters, we first chose a random Item / Item Group, made a training data set having forecast till 2017 and implemented models on it to forecast for 2018. We compared the training set to test set having values of 2018. This gave us the needed parameters to get the best fit.

For Long Term Model

For long term forecasting we used data set till end of 2017. For Example:

For Item: "MGPB-E6-10AS "							
Model	Actual Demand	Forecasted Demand	AIC	BIC	ME	MAE	MAPE
ETS	6000	8173	861.83	866.97	-1.32	4384.8	81.19
TBATS	6000	5529	843.79				
ARIMA	6000	6895	822.15	827.29	60.07	3991.63	73.79
Theta	6000	6951					
MA (1)	6000	8173			8173		
LM	6000	4989					

Taking average values of Multiple Items and item groups we reached to the conclusion of using ARIMA and TBATS for Long term forecast.

Once we selected the method, we ran the forecast package for all dates to forecast in the future. For more information and R code see Appendix 1D

For Short Term Model.

For short term model we used data last 1 year data. For Example

For Item: "MBCP-R8-M7 "							
Model	Actual Demand	Forecasted Demand	AIC	BIC	ME	MAE	MAPE
ETS	168000	13377	755	759	9.8	15580	308
TBATS	168000	6327	678				
ARIMA	168000	13378	732	735	-1.5	15585	308
Theta	168000	13178					
MA (1)	168000	13377			13388		
LM	168000	11378					

Taking average values of Multiple Items and item groups we reached to the conclusion of using ETS and THETA for Short term forecast.

Once we selected the method, we ran the forecast package for all the dates to forecast in the future. For more information and R code see Appendix 1E

For Model Till 2023.

For this model we convert the data in monthly data and convert it to time series. Then using long term forecast models we forecast the time series to get the output. Refer to Appendix 1F.

For Open Orders.

For open orders we have provided a table to summarize the open orders. If the forecast is of January 2019, we can check amount of open orders for January 2019 and estimate number of parts to be made. Refer to Appendix 1G.

SOP

1. For Long Term Models.
    a. Open the Arconic Long Term ().RMD and load the given Arconic Data 1 01 .CSV file.
    b. Input the cutoff date. (The date till which you want data)
    c. Input period for which you want forecast. (h)
    d. Select the h*h Matrix to get the forecast values.

2. For Short Term Models.
   a. Open the Arconic Short Term ().RMD and load the given Arconic Data 1 01 .CSV file.
   b. Input the cutoff date. (The date till which you want data)
   c. Input period for which you want forecast.
   d. Select the h*h Matrix to get the forecast values.
3. For Long Term till 2023Model.
   a. For Item model open LongtermR and Longterm.csv
   b. Run the Model
   c. For Item Group model open LongtermItemGroupR and LongtermItemGroup.csv
   d. Run the Model
4. For Open Orders
   a. Open Open Order.xls
   b. Filter using Item/Item Group.
   c. Filter the required Date.
   d. This will give you the month wise open order to be incorporated in forecast.

Conclusion

- The short term models would help the company manage their inventory and forecast the demands reducing the cost of inventory and help in the smooth and efficient operations. They would ensure short term benefits for all of their portfolio of products and help in better short-term planning.
- The long term models would help company restructure its long term policies based on the inventory storage requirements by predicting the future demands for several consecutive years.
- These models also incorporate the open orders to be fulfilled based on Arconic history data and open order data, thus helping the company fulfill these orders efficiently.
- Successfully provided the company with these models which would reduce the inventory storage costs based on the predicted forecasted demands for the next day, the next month and the next year.

VTI Industries

The aim of this project is:

1) To develop a tool to determine safety stock, reorder point and reorder quantity, which automatically updates itself based on the usage.

2) Maximize the truckload capacity for each order along with an optimum product mix.

Assumptions

1) The product mix in the truck would be optimum and would be such that every product is included in the product mix.
2) The model assumes a periodic review system.
3) The demand is random and stationary, hence the expected (i.e. the average) value of the demand does not change over a period of time.
4) Lead times are assumed to positive.
5) The values for the fixed ordering cost (k) = 50 and inventory holding cost (h) = 2.

Idea Flow

With the basic understanding of the given data-sheet, we started by determining the policy we could use for calculating the safety stock, reorder point and the reorder quantity. There were two possibilities a continuous review system or a periodic review system. We decided to go forward with a periodic review system, implement the (s, S) policy as it would be easier to implement and would reduce the cost for the company. We then calculated the maximum truckload and the optimum product mix for the pool of the products given to us in the data. Using excel we calculated the required values and prepared a Linear Programming model to obtain the optimal truckload.

Policy for determining the Safety Stock, Reorder Point and the Reorder Quantity.

There are two possibilities here

1. **Continuous Review System, Type 1 service level.**

The optimal policy for a type 1 service level is

Number	Steps
1	To determine R that satisfies probability (no stock-outs during Lead Time) = Alpha.
2	Set Q = Economic Order Quantity.

The Given data is

Fixed ordering cost (k) = 50
Inventory holding cost(h) = 2
Alpha = 0.95 = 95%

(Refer to the Appendix – 2(a)) (Part of the Excel sheet shown here)

	40202	40201	40200	40211	40210
Demand	1834.667	2111	1652	61.75	21.35
DDLT	2201.6	2533.2	1982.4	123.5	42.7
STDEV Weekly use	385.4677	328	255	10	10
k	100	100	100	100	100
lamda					
h	2	2	2	2	2
Alpha	0.95	0.95	0.95	0.95	0.95
Z	1.644854	1.644854	1.644854	1.644854	1.644854
R(s)	2836	3073	2402	140	60
Q	429	460	407	79	47
Order upto level(S)	3265	3533	2809	219	107
Safety stock	634	539	419	16	17

A

- We started by calculating Q.
- Q = sqrt ((2*K*lambda)/h)

B

- We then moved on to finding the R
- Probability (no stock-out occurring during lead time) = F(R) = Alpha = 0.95
- The Z-value obtained from the alpha = norm.s.in(alpha) = 1.644
- The Re-order Quantity (R) = mean demand during lead time + Z*σ
- Z = Z-Value
- σ = Standard Deviation

C

- To find the average safety stock
- The average safety stock (s) = R – μ
- Hence here we obtained the (Q, R) values from the (Q, R) model.

2. Periodic Review System, we can use a (s,S) model

Here the values of s and S are given by

$$s = R$$

$$S = Q + R$$

Here's is the inventory level below which if the inventory level for a particular product falls we replenish it up to S.

S here can also be called as the order up to level for a particular product.

Determination of the maximum truck load capacity and the optimum product mix.

To maximize the truckload capacity from each supplier. We formulated a Linear Programming problem. We could explain this LP with an example from the problem itself.

For Example:

Let us assume a supplier Georgia Pacific

Let this be the product numbers along with the max number of pieces in each truck

40202	40201	40200		40211	40210	40209
301/4"X97"	301/4"X121"	301/4"X145"		491/4"x97"	491/4"x121"	491/4"x145"
8'PB	10'PB	12'PB		4x8'PB	4x10'PB	4x12'PB
Georgia-Pacific	Georgia-Pacific	Georgia-Pacific		Georgia-Pacific	Georgia-Pacific	Georgia-Pacific
6	6	6		10	10	10
34	34	34		40	40	40
714	578	476		480	380	320

Let $X1, X2, X3, X4, X5, X6$ be the variables

Therefore, the LP is given by

$Maximize\ Z =$

$$\left(\frac{1}{714}\right) * X1 + \left(\frac{1}{578}\right) * X2 + \left(\frac{1}{476}\right) * X3 + \left(\frac{1}{480}\right) * X4 + \left(\frac{1}{380}\right) * X5 + \left(\frac{1}{320}\right) * X6$$

Subject To

$$X1 >= 1834.667$$

$$X2 >= 2111$$

$$X3 >= 1652$$

$$X4 >= 61.75$$

$$X5 >= 21.35$$

$$X6 >= 22$$

$X1, X2, X3, X4, X5, X6 > 0$

The Results obtained are as follows:

Company	Truckload required
Georgia-Pacific	10 trucks every week
Arauco	1 truck every 6 weeks
Pacific Wood	1 truck every 10 weeks
Roseburg	1 truck every 3 weeks

Conclusion

- Developed a user form in excel to find the safety stock, reorder point, and reorder quantity which automatically updates itself based on the input values and usage.
- Maximized the total truckload from each supplier for each product and optimized the number of trucks from each supplier. Found out the optimum product mix in the truck from each supplier based on the truck capacities and the product weights.

References

[1] Virtocommerce. (2018, February 22). How to calculate safety stock? Safety stock formula and calculation. Retrieved from https://virtocommerce.com/glossary/how-to-calculate-safety-stock

[2] Using tapply() with strptime() formatted date. (n.d.). Retrieved from https://stackoverflow.com/questions/24367759/using-tapply-with-strptime-formatted-date

[3]TradeGecko. (n.d.). What is safety stock and how do you calculate it? Retrieved from https://www.tradegecko.com/learning-center/determining-your-safety-stock-level

[4] Thayer, A. (n.d.). SkuVault Blog. Retrieved from https://www.skuvault.com/blog/reorder-point-formula

[5] Using tapply for the subset group of data. (n.d.). Retrieved from https://stackoverflow.com/questions/21162260/using-tapply-for-the-subset-group-of-data

[6]Reorder Point Formula: This Is What You Need to Avoid Stockouts. (2017, August 23). Retrieved from https://dearsystems.com/inventory-software/blog/reorder-point-formula/

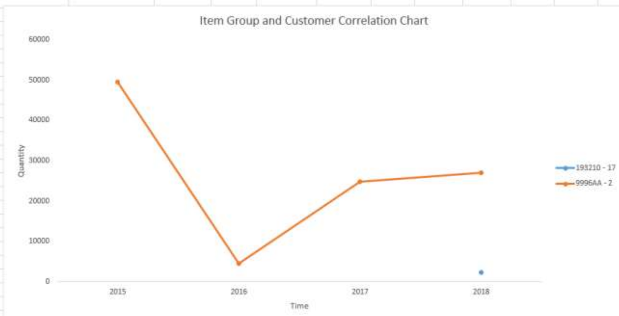
[7] Reorder Point Formula: This Is What You Need to Avoid Stockouts. (2017, August 23). Retrieved from https://dearsystems.com/inventory-software/blog/reorder-point-formula/

[8]Using tapply for the subset group of data. (n.d.). Retrieved from https://stackoverflow.com/questions/21162260/using-tapply-for-the-subset-group-of-data

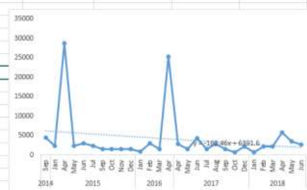
Appendix

Appendix 1A

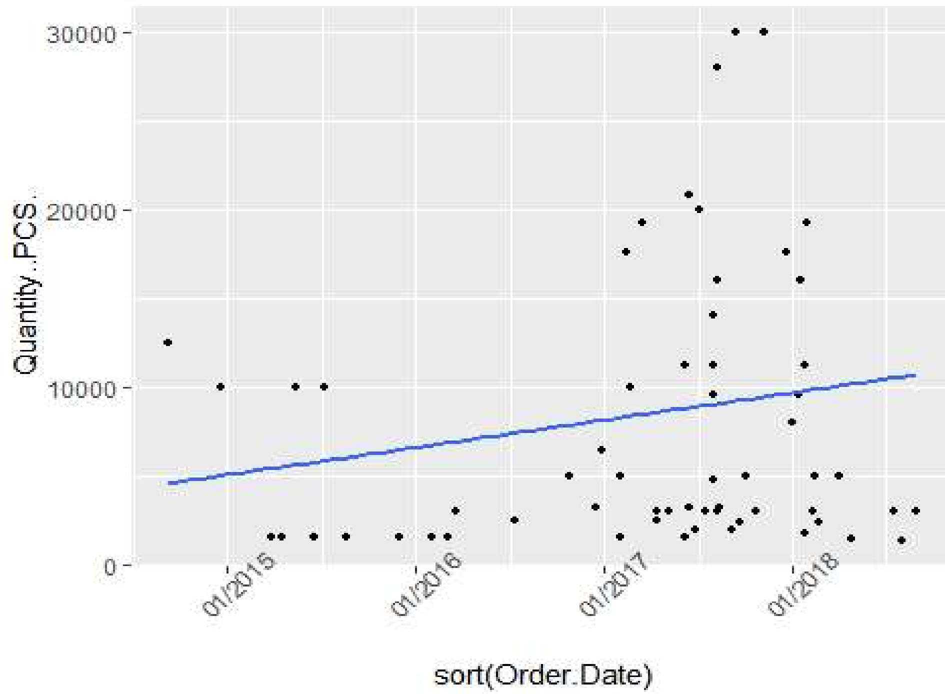
Item #	3LC120-2R10UL				
Sum of Quantity (PCS)	Column Labels				
	193210	193210 Total	9996AA	9996AA Total	Grand Total
Row Labels	17		2		
2015			49500	49500	49500
2016			4500	4500	4500
2017			24750	24750	24750
2018	2250	2250	27000	27000	29250
Grand Total	2250	2250	105750	105750	108000

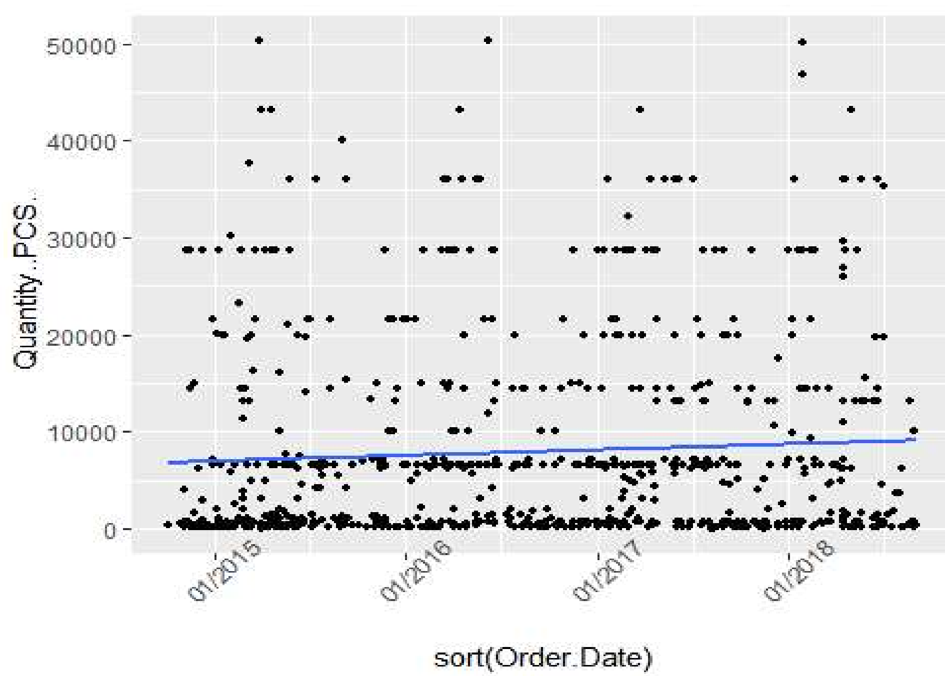


Item #	(All)
Item Group (Product Family)	10210
Row Labels	Sum of Quantity (PCS)
2014	4500
Sep	4500
2015	44400
Jan	2250
Apr	28650
May	2250
Jun	3000
Jul	2250
Sep	1500
Oct	1500
Nov	1500
Dec	1500
2016	30450
Jan	750
Feb	3000
Mar	1500
Apr	25200
2017	16800
Apr	2800
May	1400
Jun	4200
Jul	1400
Aug	2800
Sep	1400
Oct	700
Dec	2100
2018	16856
Jan	700
Feb	2100
Mar	2100
Apr	5800
May	3500
Jun	2656



Appendix 1B


30000
20000
10000
0
Quantity..PCS.
01/2015
01/2016
01/2017
01/2018
sort(Order.Date)


50000
40000
30000
20000
10000
0
Quantity..PCS.
01/2015
01/2016
01/2017
01/2018
sort(Order.Date)

Appendix 1C

Order Date	Request Date	Ship Date Promise	Ship Date Actual	Item #	Item Group (Product Family)	Quantity (PCS)	LT at Time of Order (WKS)	Production Date
7/13/2015	8/18/2015	8/18/2015	42234	C137LT-U8-8	12100	25650	11	42157
9/26/2018	9/27/2018	9/27/2018	43370	MGCW-R8U	12610	720	6	43328
8/20/2018	8/23/2018	9/26/2018	43370	MGC-R8U	12610	10000	6	43328
7/30/2018	8/2/2018	9/13/2018	43370	MGCW-R8U	12610	28800	6	43328
9/25/2018	10/1/2018	10/1/2018	43369	MGC-R8U	12610	10932	6	43327
8/27/2018	9/26/2018	9/26/2018	43369	MGP98T-R6-10GS1	12353	13500	6	43327
8/2/2018	9/12/2018	9/19/2018	43369	MGCW-R8U	12610	18000	6	43327
8/27/2018	9/26/2018	9/26/2018	43369	MGPB-R8-10GS1	12353	22500	6	43327
7/2/2018	9/19/2018	9/19/2018	43369	MGCW-R8U	12610	27000	6	43327
6/25/2018	9/26/2018	9/26/2018	43369	MGC-R8U	12610	50000	6	43327
8/9/2018	8/13/2018	10/1/2018	43369	MGCW-R8U	12610	111600	6	43327
7/23/2018	9/17/2018	9/17/2018	43369	MGCW-R8U	12610	113400	6	43327
9/10/2018	9/26/2018	9/26/2018	43369	MGC-R8U	12610	150000	6	43327
8/27/2018	9/25/2018	9/25/2018	43368	MGP98T-R6-10GS1	12353	6750	6	43326
9/10/2018	9/25/2018	9/25/2018	43368	MGPB-R8-10GS1	12353	45000	6	43326
9/20/2018	9/25/2018	9/25/2018	43368	MGPB-R8-10GS1	12353	45000	6	43326
9/10/2018	9/25/2018	9/25/2018	43368	MGC-R8U	12610	150000	6	43326
9/21/2018	9/26/2018	9/26/2018	43369	MGPB-R8-20GS1	12353	3000	7	43320
8/20/2018	9/19/2018	9/19/2018	43362	MGP98T-R6-10GS1	12353	6750	6	43320
7/31/2018	7/31/2018	9/18/2018	43361	MGC-R8U	12610	50000	6	43319
7/17/2018	9/17/2018	9/17/2018	43360	MGC-R8U	12610	20000	6	43318
7/17/2018	9/17/2018	9/17/2018	43360	MGC-R8U	12610	150000	6	43318
7/17/2018	9/17/2018	9/17/2018	43360	MGC-R8U	12610	150000	6	43318
7/17/2018	9/17/2018	9/17/2018	43360	MGC-R8U	12610	150000	6	43318
7/17/2018	9/17/2018	9/17/2018	43360	MGC-R8U	12610	450000	6	43318
8/13/2018	9/12/2018	9/12/2018	43371	SBS62MGIC	15300	4000	8	43315
9/24/2018	10/3/2018	10/3/2018	43371	SBS62MGIC	15300	12000	8	43315
7/11/2018	9/14/2018	9/14/2018	43371	MGCW-R6U	12600	17000	8	43315
6/21/2018	9/14/2018	9/14/2018	43371	MGCW-R6U	12600	21250	8	43315
9/10/2018	9/26/2018	9/26/2018	43371	SBS62MGIC	15300	24000	8	43315
8/3/2018	8/8/2018	10/3/2018	43371	MGC-R6U	12600	30000	8	43315
9/3/2018	10/3/2018	10/3/2018	43371	MGC-R6U	12600	40000	8	43315
7/23/2018	7/26/2018	9/6/2018	43357	MGCW-R8U	12610	50400	6	43315
9/10/2018	10/10/2018	10/10/2018	43371	MGC-R6U	12600	60000	8	43315
8/7/2018	9/5/2018	10/10/2018	43371	MGC-R6U	12600	60000	8	43315
9/10/2018	10/3/2018	10/3/2018	43371	MGC-R6U	12600	100000	8	43315
7/30/2018	8/2/2018	9/13/2018	43357	MGCW-R8U	12610	100800	6	43315

````
```{r}
#Choosing the .csv file
Arc<-read.csv(file.choose(),header=T)
# Neglecting the data  which contain NA
df<-Arc[,c(5,9,10,11)]
df=na.omit(df)
df$Order.Date=as.POSIXct(df$Order.Date)
```
````

````
```{r}
#Run this block to get Unique Item Number
df$Item..<-as.character(df$Item..)
sort(unique(df$Item..))
```
````

````
```{r}
#Generating a new data frame with Required Item Number as input
x<-subset(df,Item..=="MGLP-U6-4        ")##Replace with required Item Number
```
````

````
```{r}
#Cleaning data assuming Normally Distributed Demand and 2 sigma levels
X<-mean(x$Quantity..PCS.)
s<-sd(x$Quantity..PCS.)
Y<-subset(x,(x$Quantity..PCS.<X+2*s | x$Quantity..PCS.>X-2*s) )
sorted.data<-Y[order(Y$Order.Date),]
#Defining the data as Irregular Time Series & Sorting according to dates
irts<-irts(sorted.data$Order.Date,sorted.data$Quantity..PCS.)
#Dividing data into training and testing sets with cut-off date
cutoff_date <-readline(prompt = "Enter a cut-off date for training
(YYYY-MM-DD): ")
train<-subset(irts,irts$time<cutoff_date)
h <- readline(prompt = "Enter the desired forecast period: ")
h<-as.integer(h)
````

Appendix 1D

---

title: "Arconic"
````

~~~
output: pdf_document
---
```
install.packages("forecast")
install.packages("tstools")
install.packages("MAPA")
install.packages("forecTheta")
```

```{r}
library(forecast)
library(tstools)
library(stats)
##library(MAPA)
library(forecTheta)
```



```{r}
Arc<-read.csv(file.choose(),header=T)
```

```{r}
df<-Arc[,c(5,9,11)]
df=na.omit(df)
df
```

```{r}
df$Order.Date=as.POSIXct(df$Order.Date)
```



```{r}
df$Item..<-as.character(df$Item..)
df$Item..

```

```{r}
x<-subset(df,Item..=="MGPB-E6-10AS    ")

```

```{r}
x
```

```{r}
X<-mean(x$Quantity..PCS.)
s<-sd(x$Quantity..PCS.)
```



```{r}
Y<-subset(x,(x$Quantity..PCS.<X+2*s | x$Quantity..PCS.>X-2*s) )
~~~

````
```

```{r}
Y
```
```{r}
sorted.data<-Y[order(Y$Order.Date),]
```

```{r}
sorted.data
```

```{r}
irts<-irts(sorted.data$Order.Date,sorted.data$Quantity..PCS.)

```


```{r}
train<-subset(irts,irts$time<"2017-12-31")

```

```{r}
test<-subset(irts,irts$time>"2017-12-31")
```



```{r}
plot(train)
```

```{r}
fit.ets<-ets(train$value)
```


```{r}
fit.ets
```



```{r}
f.ets<-fit.ets
```

```{r}
fit.arima<-auto.arima(train$value)
```

```{r}
fit.arima
```
````
````

~~~
```{r}
f.arima<-fit.arima
```

```{r}
fit.tbats<-tbats(train$value)
```

```{r}
fit.tbats
```

```{r}
f.tbats<-fit.tbats
```


```{r}
fit.theta<-thetaf(train$value)
```

```{r}
fit.theta
```

```{r}
f.theta<-fit.theta
```
```{r}
forecast(f.arima,h=1)
plot(forecast(f.arima,h=1))
```


```{r}
fit.lm<-lm(train$value~train$time,train$time)
```

```{r}
fit.lm
```

```{r}

plot(train$time,train$value,type="l")
abline(fit.lm)
```

```
~~~

Appendix 1E
Item Group

~~~
---
title: "Arconic Project"
output: pdf_document
---
~~~

```
#```{r}
#Block to install all the packages used in the code
install.packages("forecast")
install.packages("tstools")
install.packages("stats")
install.packages("forecTheta")
install.packages("tseries")
#```

```{r}
#Block to call all the required libraries
library(forecast)
library(tstools)
library(stats)
library(forecTheta)
library(tseries)
```

```{r}
#Choosing the .csv file
Arc<-read.csv(file.choose(),header=T)
# Neglecting the data  which contain NA
df<-Arc[,c(5,9,10,11)]
df=na.omit(df)
df$Order.Date=as.POSIXct(df$Order.Date)
```

```{r}
#Run this block to get Unique Item Group
df$Item.Group..Product.Family.<-as.character(df$Item.Group..Product.Family.)
sort(unique(df$Item.Group..Product.Family.))
```

```{r}
#Generating a new data frame with Required Item Group as input
x<-subset(df,Item.Group..Product.Family.=="11287")##Replace with required Item Group Number
```

```{r}
#Cleaning data assuming Normally Distributed Demand and 2 Sigma levels
X<-mean(x$Quantity..PCS.)
s<-sd(x$Quantity..PCS.)
Y<-subset(x,(x$Quantity..PCS.<X+2*s | x$Quantity..PCS.>X-2*s) )
sorted.data<-Y[order(Y$Order.Date),]
#Defining the data as Irregular Time Series & Sorting according to dates
irts<-irts(sorted.data$Order.Date,sorted.data$Quantity..PCS.)
#Dividing data into training and testing sets with cut-off date
cutoff_date <-readline(prompt = "Enter a cut-off date for training (YYYY-MM-DD): ")
train<-subset(irts,irts$time > cutoff_date)
h <- readline(prompt = "Enter the desired forecast period: ")
h<-as.integer(h)
```

```{r}
plot(train)
```
```

~~~~
```{r}
#Forecasting using Exponential Time Series
fit.ets<-ets(train$value)
fit.ets
forecast(fit.ets,h=h)
plot(forecast(fit.ets,h=h))
```


```{r}
#Forecasting using Simple Exponential Smoothing with Drift (THETA Method)
fit.theta<-thetaf(train$value)
fit.theta
forecast(fit.theta,h=h)
plot(forecast(fit.theta,h=h))
```

Item
---
title: "Arconic Project"
output: pdf_document
---
#```{r}
#Block to install all the packages used in the code
install.packages("forecast")
install.packages("tstools")
install.packages("stats")
install.packages("forecTheta")
install.packages("tseries")
#```

```{r}
#Block to call all the required libraries
library(forecast)
library(tstools)
library(stats)
library(forecTheta)
library(tseries)
```

```{r}
#Choosing the .csv file
Arc<-read.csv(file.choose(),header=T)
# Neglecting the data  which contain NA
df<-Arc[,c(5,9,10,11)]
df=na.omit(df)
df$Order.Date=as.POSIXct(df$Order.Date)
```

```{r}
#Run this block to get Unique Item Number
df$Item..<-as.character(df$Item..)
sort(unique(df$Item..))
```
~~~~

```
```{r}
#Generating a new data frame with Required Item Number as input
x<-subset(df,Item..=="MGLP-U6-4       ")##Replace with required Item Number
```

```{r}
#Cleaning data assuming Normally Distributed Demand and 2 Sigma levels
X<-mean(x$Quantity..PCS.)
s<-sd(x$Quantity..PCS.)
Y<-subset(x,(x$Quantity..PCS.<X+2*s | x$Quantity..PCS.>X-2*s) )
sorted.data<-Y[order(Y$Order.Date),]
#Defining the data as Irregular Time Series & Sorting according to dates
irts<-irts(sorted.data$Order.Date,sorted.data$Quantity..PCS.)
#Dividing data into training and testing sets with cut-off date
cutoff_date <-readline(prompt = "Enter a cut-off date for training (YYYY-MM-DD): ")
train<-subset(irts,irts$time>cutoff_date)
h <- readline(prompt = "Enter the desired forecast period: ")
h<-as.integer(h)
```

```{r}
plot(train)
```

```{r}
#Forecasting using Exponential Time Series
fit.ets<-ets(train$value)
fit.ets
forecast(fit.ets,h=h)
plot(forecast(fit.ets,h=h))
```


```{r}
#Forecasting using Simple Exponential Smoothing with Drift (THETA Method)
fit.theta<-thetaf(train$value)
fit.theta
forecast(fit.theta,h=h)
plot(forecast(fit.theta,h=h))
```
```

Appendix 1F

Item

```
#Load the csv file
df=read.csv(file.choose(),header=T)
#install the following libraries if not already installed bu using command install.packages("packagename")
library(forecast)
library(tseries)
#Converting the dates into standard format to be run in following commands)
time1=as.POSIXct(df$Date)
#Assuming all the na values as 0
df[is.na(df)]=0
#plotting the time series
plot(irts(time1,df$MGLP.R8.6))
#Assigning functions to variables for ease of furhter use
a=irts(time1,df$MGLP.R8.6)
b=auto.arima(a$value)
```

```
x=ets(a$value)
#forecasting using fitted models
forecast(b,h=4)
forecast(x,h=4)
```

Item Group

```
#Load the csv file
df=read.csv(file.choose(),header=T)
#install the following libraries if not already installed bu using command install.packages("packagename")
library(forecast)
library(tseries)
#Converting the dates into standard format to be run in following commands)
time1=as.POSIXct(df$Date)
#Assuming all the na values as 0
df[is.na(df)]=0
#plotting the time series
plot(irts(time1,df$X12320))
#Assigning functions to variables for ease of furhter use
a=irts(time1,df$X12320)
b=auto.arima(a$value)
x=ets(a$value)
#forecasting using fitted models
forecast(b,h=4)
forecast(x,h=4)
```

## Appendix 1G

| | |
|---|---|
| Item # | (All) |
| Item Group (Product Family) | (All) |

| Row Labels | Sum of Quantity (PCS) |
|---|---|
| 2015 | 44812045 |
| 2016 | 593310 |
| 2017 | 477775 |
| 2018 | 177975296 |
| **Grand Total** | **223858426** |