Kyung Geun Kim
Sonia Park
Min Jae Shin
Julie Yi

Prediction of Chill-E-AC July Sales Around the Holidays

Executive Summary:

For our project, we chose to predict the sales of the first 10 days of July 2019 of AC units by (fictional) Chill-E-AC based on sales data from January 2015 through June 2019. We fit a sinusoidal model on the differenced log transformation of the dataset and used seasonal ARIMA models to analyze the residuals and then inverted the values from the analysis back into the format of the original dataset using inverse functions.

1. Exploratory Data Analysis

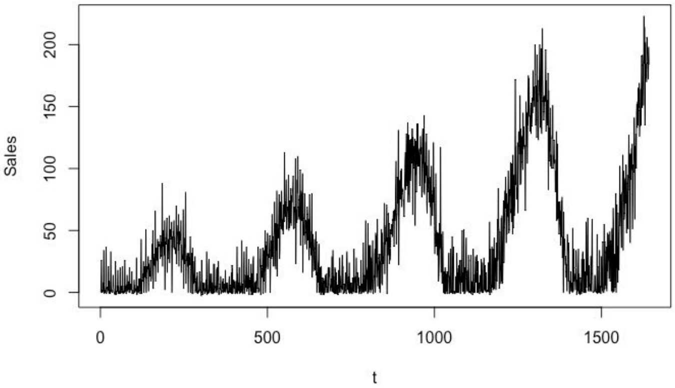

Figure 1.1: This is the time series data of AC unit sales plotted from January 2015 to June 2019.

This entire dataset has 1642 data entries. Despite the many data points, we could easily identify a seasonal trend presumably on a yearly scale. Figure 1.1 shows us somewhat regular changes to the number of sales of AC units occurring about every 365 days (every year). However, the seasonality is incongruous and there is evident heteroskedasticity in the data. The range of sales continues to grow every year as can be seen with the increasing peaks. To better understand these features, we focused on specific index ranges of the data.

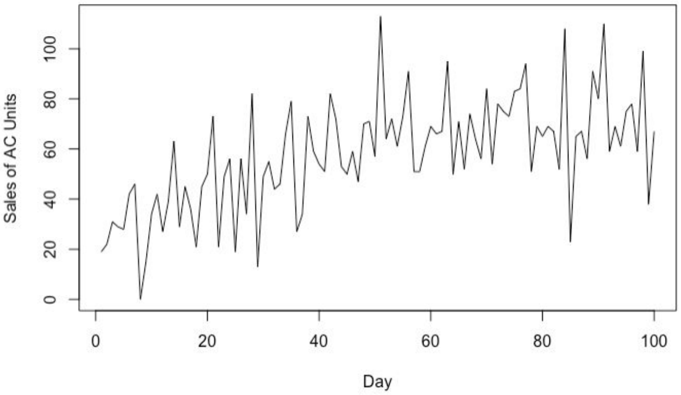

Figure 1.2: This is the time series data of the AC unit sales plotted from Day 501 to 600

Figure 1.2 is a plot of 100 indices for 100 days. The plot shows us that there is a definite noticeable upward spike in sales every 7 days, confirming the existence of a weekly seasonality. Therefore, we can confirm that there is heteroskedasticity in the data and both weekly and annual seasonal trend in the dataset.

2. Model of Time

Before applying ARIMA models, we first pursued stationarity through data transformation and detrending. Our first step was to use the log function to the original data to solve heteroskedasticity issue. Some data entries were negative, which are most likely returned goods. To account for this, we subtracted the smallest number (which is negative so this would shift all of the values to positive) and added a 0.01 to prevent a negative infinite value. After this step, there were still some irregularities in the data. Some parts of the data had many movements while others did not, indicating that the residuals would not be stationary.

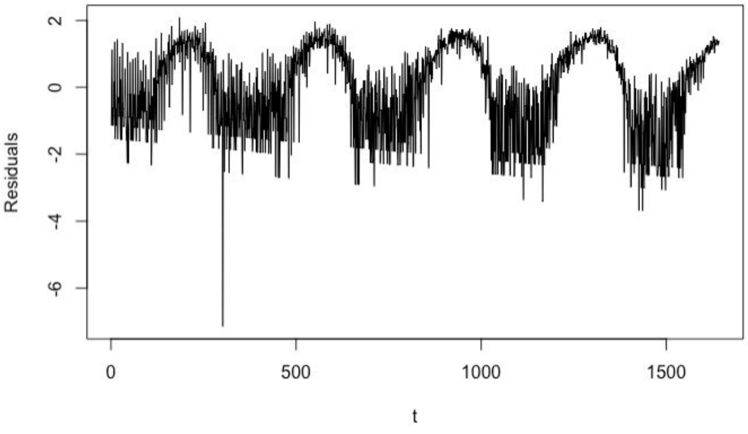

Figure 2.1: This is the shape of the residuals from the original dataset.

Figure 2.1 demonstrates this issue because there is more residual movement as time passes. We chose to difference the data by every 6 months or 365/2 rounding down to 182 lags to remove its time dependence and stabilize the function. After that, we fit a sinusoidal function of $Y \ = \ ax + c_1 \ \cos(2\pi ft) + c_2 \sin(2\pi ft)$ format to compensate for scaling amplitude over time.

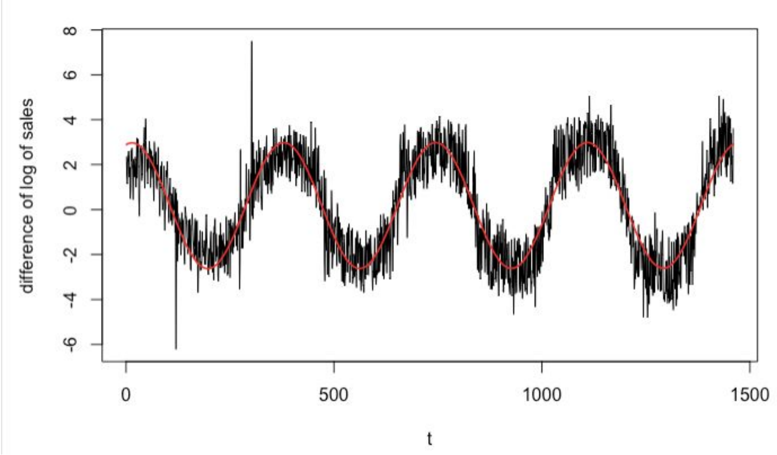

Figure 2.2: This plot shows the fitted sinusoidal model in red with the difference of the log of sales by half a year or 182 days.

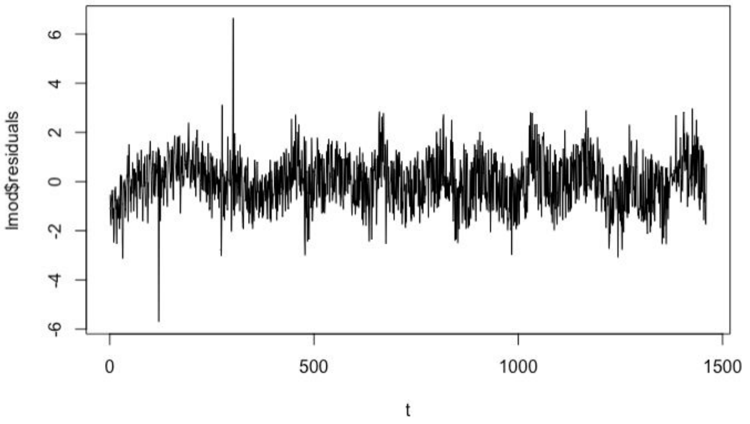

Figure 2.3: This plot shows the residuals of the dataset of the fitted sinusoidal model.

Figure 2.2 holds the differenced data fitted with the sinusoidal model with a period of a year. The data was fit much better and avoided overfitting because the model was simple and created a much more stationary residual plot. It is important to note that there are 2 big outliers, marked by the spiked lines in Figure 2.2, however, they are not extremely significant, especially in a sample size greater than 1000. Figure 2.3 shows an otherwise stationary dataset for the residuals of the fitted model around the 0 except for its initial start.

3. ARIMA Model Selection

For the Model Selection, we selected three ARIMA Models based on some speculations and guesses.

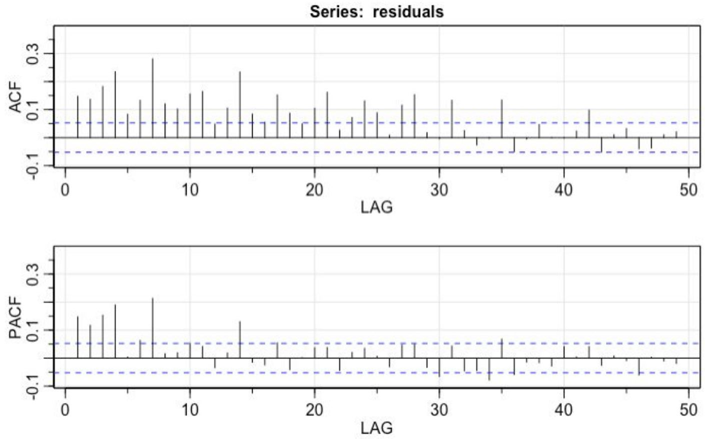

Figure 3.1: This is the ACF and PACF plot of the residuals

Based on Figure 3.1, there is a clear seasonality every 7 lags so we have set our seasonal period S = 7. Also, the values in the PACF seem significant up to the second seasonal lag; there is also significance in the 4th lag of the AR model. In terms of the MA, there seemed to be a clear MA variable that existed up to the 4th lag in the ACF plot but some sort of MA seasonal lag seemed to exist as observable in the PACF plot. Therefore, these are the three models and their scores that were tested based on some speculations and guesses from Figure 3.1 and the mean of the sum of squared errors performed on 250 iterations of cross validation for model 2 and 3 and 96 iterations for model 1 due to convergence error.

3.1 Cross-Validation and Model Comparison

	Model 1 (Fit a Sinusoidal Model and then $SARIMA\ (4,0,3)(2,1,1))_7$	Model 2 (Fit a Sinusoidal Model and then $SARIMA\ (4,0,4)(2,1,1))_7$	Model 3 (Fit a Sinusoidal Model and then $SARIMA\ (4,0,2)(2,1,3))_7$
AIC	2.815460	2.813717	2.809099
AICc	2.815585	2.813865	2.809247
BIC	2.858898	2.860775	2.856157
CV means	13.087485	12.007740	12.444484

The values for Model 2 and Model 3 have the best results but the cross validation mean for Model 2 is much lower than that of Model 3 on average. As a result, we have determined that Model 2 is the ideal model for forecasting since cross validation is a better indicator of prediction than any of the fitting criterion.

4. Results

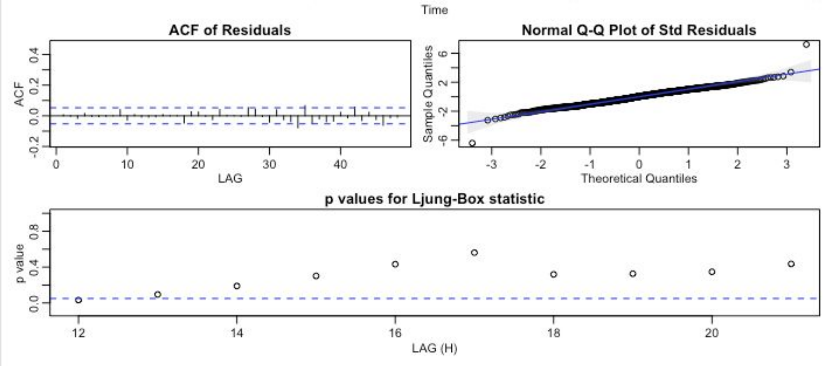

Figure 4.1: The model shows good ACF plots and there seems to be good normality despite some outliers in the end and the Ljung-Box statistic looks nice since a lot of the lags are significant.

The final resulting model takes a shape of the following.

$$(1-\Phi_1 B-\Phi_2 B^2)(1-\phi_1 B\ -\phi_2 B^2-\phi_3 B^3-\phi_4 B^4)\nabla_1{}^7=(1-\Theta_1 B)(1-\theta_1 B-\theta_2 B^2-\theta_3 B^3-\theta_4 B^4)Z_t$$

4.1 Estimation of Model Parameters

Parameters	Estimate (s.e.)
Φ_1	-0.0808 (0.0309)
Φ_2	0.0744 (0.0281)
Θ_1	-1.0000 (0.0065)
ϕ_1	-0.2324 (0.1392)
ϕ_2	-0.1703 (0.0672)
ϕ_3	0.5700 (0.0832)
ϕ_4	0.6515 (0.1423)
θ_1	0.2850 (0.1441)

θ_2	0.2592 (0.0661)
θ_3	-0.4686 (0.0996)
θ_4	-0.5138 (0.1419)

4.2 Prediction

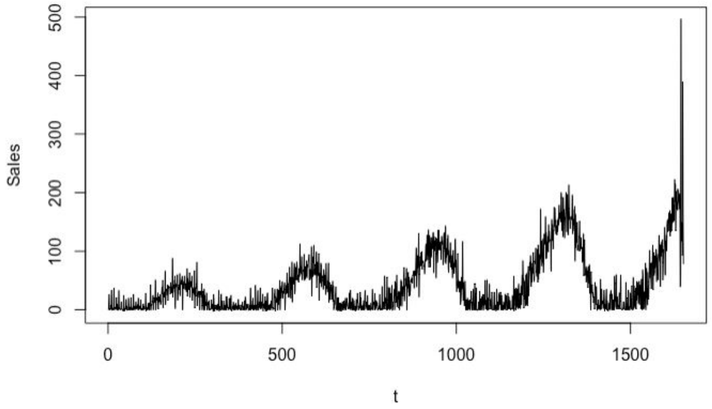

Figure 4.2.1: This is the plot of the original data with its 10 day forecast into July 2019.

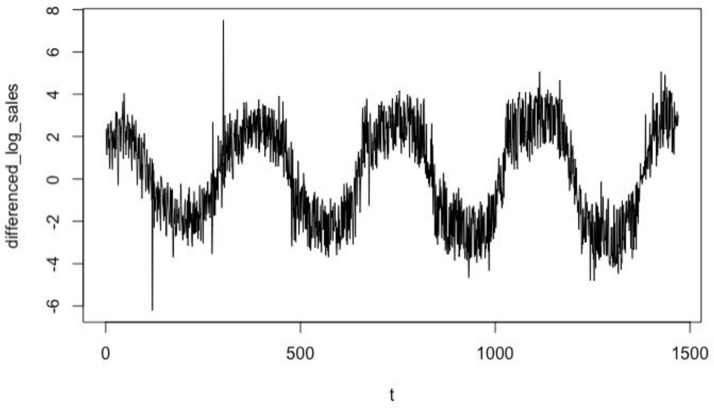

Figure 4.2.2: The plot of the differenced of the log of the original data and the forecast

After this value was given, the data was forecasted for 10 days and added to the initial sinusoidal fitted data. Then, all of the previous data was inverted using functions in R such as *diffinv* which inverts differenced data given its initial values, then exponentially multiplied the data and made the appropriate shifts that were made in the modifications in the beginning. The predicted forecast looks good on the modified data as shown in Figure 4.2.2, but when converted into the original values, the forecasts did not look too good as shown in Figure 4.2.1.

~~~
---
title: "Appendix for Project Code"
author: ""
date: ""
output:
  html_document:
    fig_height: 3
    fig_width: 5
---
<!-- Don't edit in between this line and the one below -->
```{r include=FALSE}
# Don't delete this chunk if you are using the DataComputing package
library(DataComputing)
library(astsa)
```
*Source file*
```{r, results='asis', echo=FALSE}
includeSourceDocuments()
```
<!-- Don't edit the material above this line -->


```{r}
sales <- read.csv("/Users/minjaeshin/Downloads/sales.csv")
plot(sales$sales,type = 'l',ylab="Sales",xlab = "t")#1642 observations is a lot to see in a plot
so zoomed in
original_sales<-sales$sales
plot(sales$sales[1:500],type = "l",ylab="Sales",xlab="t") #There is about yearly seasonality (can
assume somewhat of a 365 day period). but there might be another seasonality
plot(sales$sales[501:600],type = "l",ylab="Sales of AC Units",xlab="t") #There seems a weekly
seasonality as well.
plot(sales$sales[550:575],type = "l",ylab="Sales",xlab="t") #evidently a spike every 7 days or a
week
lmod1<-lm(sales~X,sales)
plot(lmod1$residuals) #This is heteroskedastic and has the trends We can see a trend in the
residuals so we must model this better
sales$sales<-log(sales$sales-min(sales$sales)+0.01) #Shift
lmod1<-lm(sales~X,sales)
plot(lmod1$residuals,type='l',xlab="t",ylab="Residuals") #there is an outlier but looks more
congruent plus the seasonality

```
```{r}
sales_diff<-diff(sales$sales,365/2)
X<-sales$X[183:nrow(sales)]
lmod<-lm(sales_diff~X+cos(2*pi*X/365)+sin(2*pi*X/365))
plot(sales_diff,type = 'l',ylab="difference of log of sales",xlab="t")
lines(lmod$fitted.values,lwd="1.5",col="red")
plot(lmod$residuals,type="l",xlab="t") #residuals have some trends, but once those trends are
solved, looks like white noise
```
```{r}
residuals <- lmod$residuals
acf2(residuals)
```
```{r}
model1<-sarima(residuals,p=4,d=0,q=3,P=2,D=1,Q=1,S=7)
model2<-sarima(residuals,p=4,d=0,q=4,P=2,D=1,Q=1,S=7)
model3<-sarima(residuals,p=4,d=0,q=2,P=2,D=1,Q=3,S=7) #Model 1 gives error when optimizing in CV
so separately run until it can.
```

```{r}
sum_squared_errors=c(0, 0)
  for (k in 1201:(length(residuals)-9)){
    train_set <- residuals[1:k-1]
    test_set <- residuals[k:k+9]
~~~

~~~
    #forecast1=sarima.for(train_set,n.ahead=10, p=4,d=0,q=3,P=2,D=1,Q=1,S=7)
    forecast2=sarima.for(train_set,n.ahead=10,p=4,d=0,q=4,P=2,D=1,Q=1,S=7)
    forecast3=sarima.for(train_set, n.ahead=10,p=4,d=0,q=2,P=2,D=1,Q=3,S=7)
    sum_squared_errors[1] = sum_squared_errors[1] + sum((forecast2$pred - test_set)^2)
    sum_squared_errors[2] = sum_squared_errors[2] + sum((forecast3$pred - test_set)^2)
    #sum_squared_errors[3] = sum_squared_errors[3] + sum((forecast1$pred - test_set)^2) #ran it
separately in the bottom
    #because Model 1 was the source of error
  }
```

```{r}
cv<-sum_squared_errors/(k-1201)
cv
```

```{r}
error<-0
for (k in 1201:(length(residuals)-9)){
    train_set <- residuals[1:k-1]
    test_set <- residuals[k:k+9]
    forecast1=sarima.for(train_set,n.ahead=10, p=4,d=0,q=3,P=2,D=1,Q=1,S=7)
    error = error + sum((forecast1$pred - test_set)^2)
} #There is an error at K = 1297 so there were a total of 96 cv run
```


```{r}
model1_score<-c(model1$AIC,model1$AICc,model1$BIC,error/(k-1201))
model2_score<-c(model2$AIC,model2$AICc,model2$BIC,cv[1])
model3_score<-c(model3$AIC,model3$AICc,model3$BIC,cv[2])

scores<-data.frame(model1_score,model2_score,model3_score)
rownames(scores)<-c("AIC","AICc","BIC","CV")
colnames(scores)<-c("Model 1","Model 2","Model 3")
scores
```

```{r}
forecast_final=sarima.for(residuals,n.ahead=10,p=4,d=0,q=4,P=2,D=1,Q=1,S=7)
```
```{r}
future_values <- (nrow(sales)+1):(nrow(sales)+10)
input<-data.frame(future_values)
names(input)<-"X"
predicted_model<- predict(lmod,input)
predicted_model+ forecast_final$pred
diff_inverse<-diffinv(c(sales_diff,predicted_model+ forecast_final$pred),lag =
365/2,xi=sales$sales[1:182])
forecast_unit<-exp(diff_inverse)-0.01+min(original_sales)
```

```{r}
result<- round(forecast_unit[1643:1652])
result
```

```{r}
model2$fit
write.table(data.frame(result),file="sales_26178042_3032139345_3032662843_3032222194.csv",row.names
 = FALSE,col.names = FALSE)
```

```{r}
plot(forecast_unit,type="l",ylab = "Sales",xlab="t")

plot_with_forecast<-diff(log(forecast_unit-min(forecast_unit)+0.01),182)
~~~

~~~~
plot(plot_with_forecast,type="l",ylab= "differenced_log_sales",xlab="t")
```
~~~~