Forecasting Sales of Specific Items Across Grocery Stores

Beckham Wee

Contents

1. Abstract

The dataset analysed in this paper consists of 5 years of daily store-item sales data across 10 different stores. The goal is to conduct time series analysis on the monthly sales of one of the items, evaluating the data using Box-Jenkins Methodology, and ultimately fitting it into a an appropriate seasonal autoregressive integrated moving model (SARIMA) model to forecast its 3-month ahead monthly mean future values.

2. Introduction

The training data spans from 2013 to 2017, and can be found here on Kaggle. This dataset was made as part of a Kaggle competition to forecast 50 different items and forecasting daily data.

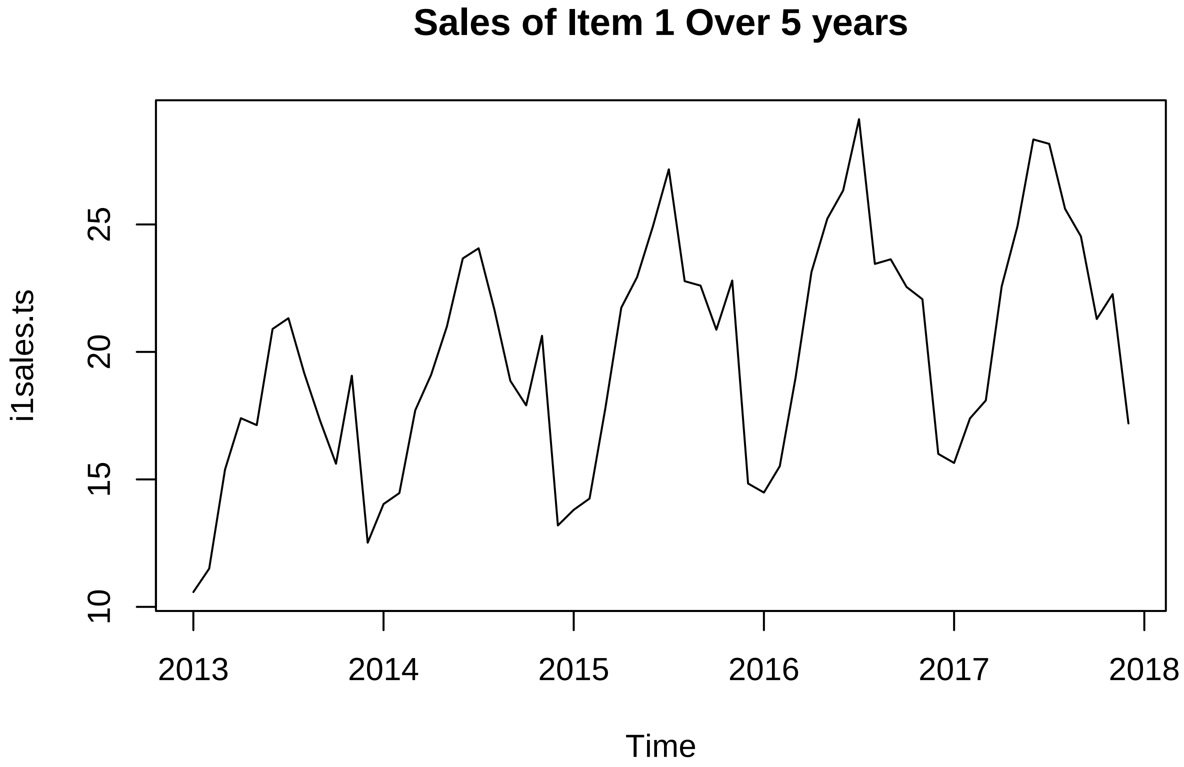
However, our goal is to to try and analyse the seasonality and trends of one item across multiple store locations. We conduct our analysis by first doing exploratory data analysis to identify trends and seasonality, before applying appropriate transformations to attain stationarity. We then difference to remove the seasonality and trends, conducting parameter estimation and train our SARIMA model and conducting model diagnostics. Finally, we forecast and compare the values to our test set, before coming to a final conclusion on the model.

2. Exploratory Data Analysis

Firstly, we load the training dataset and get the the very first item from the dataset, converting it into time series data and plotting it.

We convert this into monthly data instead

```
## # A tibble: 60 x 2
##    month      sales
##    <date>     <dbl>
##  1 2013-01-01  10.6
##  2 2013-02-01  11.5
##  3 2013-03-01  15.4
##  4 2013-04-01  17.4
##  5 2013-05-01  17.1
##  6 2013-06-01  20.9
##  7 2013-07-01  21.3
##  8 2013-08-01  19.2
##  9 2013-09-01  17.3
## 10 2013-10-01  15.6
## # i 50 more rows
```

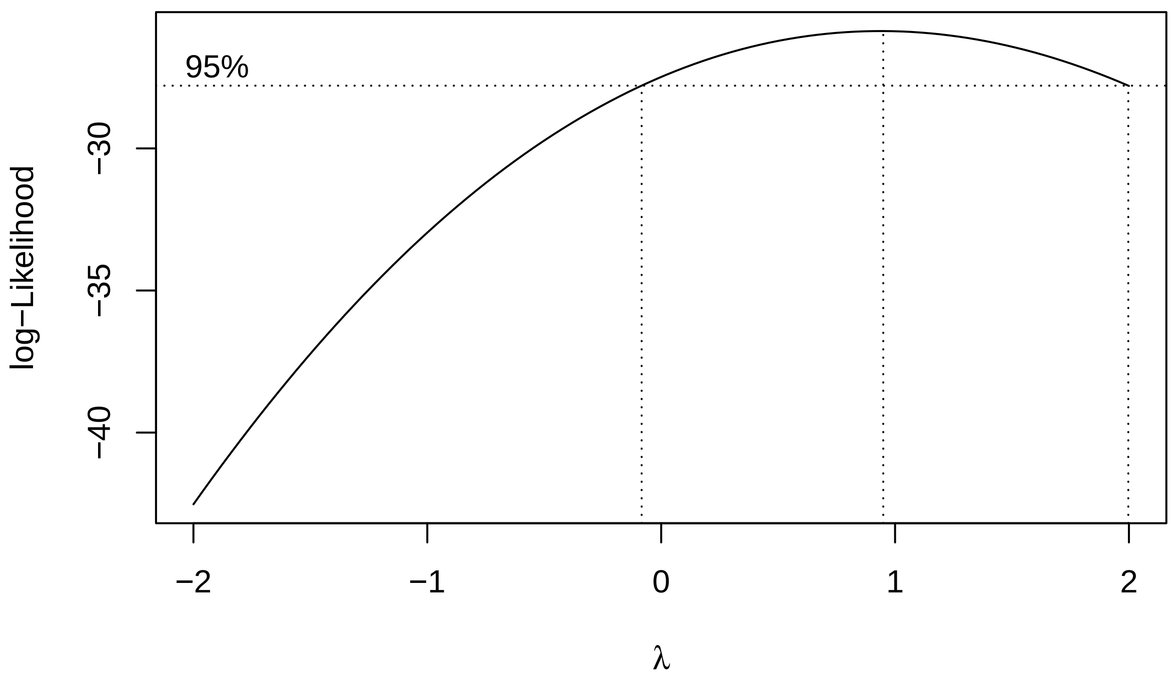


From the plot, there seems to be a seasonal component. Sales always seem to increase in the middle of the year, before going back down. Variance also seems to slightly increase over time. As observed, there is also an increasing trend in the data.

3. Data Transformation

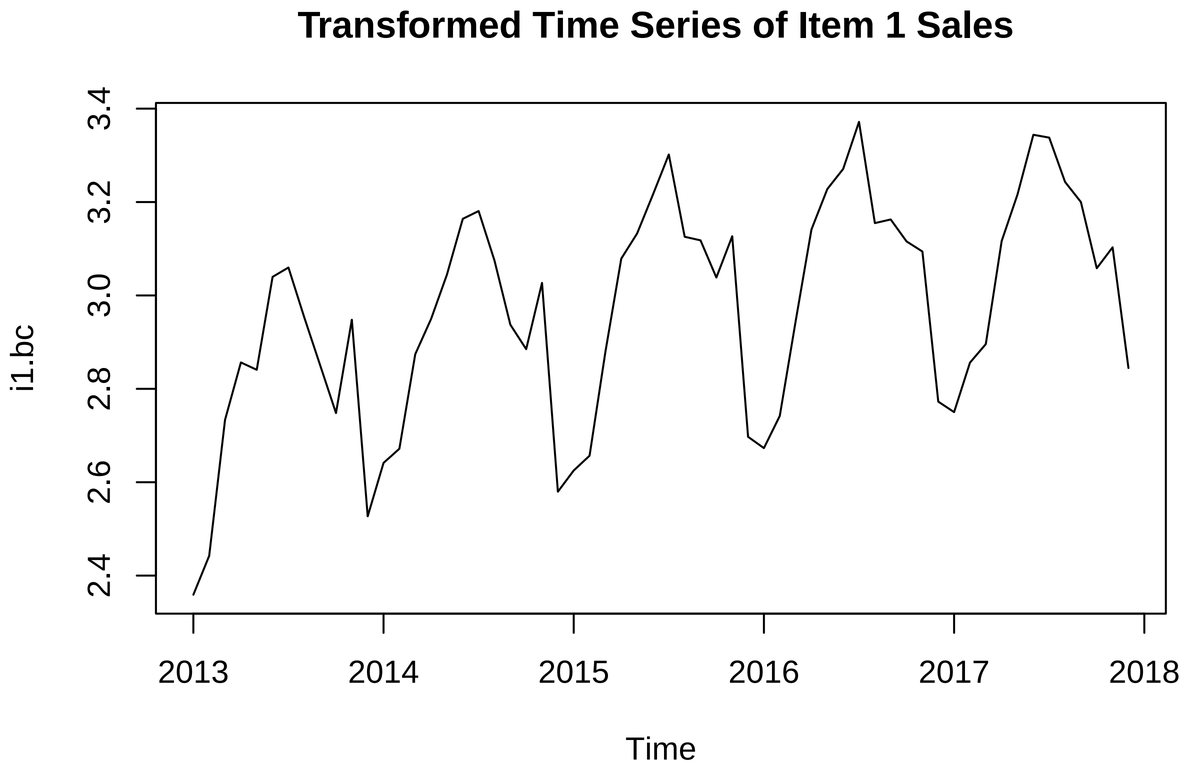
In an attempt to stabilise the variance, a Box-Cox transformation would be necessary. We do so by first finding the optimal λ parameter using `boxcox()`.

```
# Finding Lambda
t <- 1:length(i1sales.ts)
transformed <- boxcox(i1sales.ts~t, plotit=TRUE)
```



Within the 95% Confidence Interval, we see that $\lambda = 0$ is in it, hence we stabilise the variance (or seasonal effects if any), and make the data closer to a Normal Distribution using the following:

$$Y_t^{(\lambda)} = log(Y_t)$$

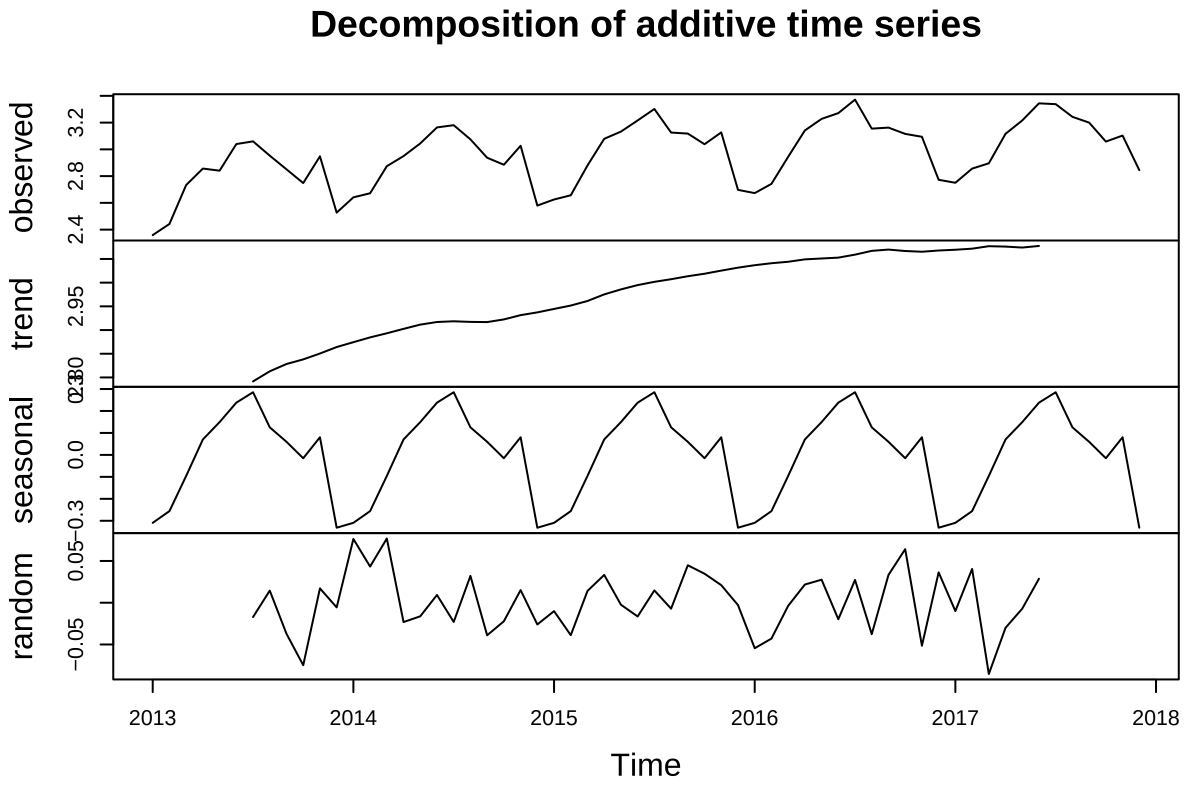


```
## Original variance:  20.04183
```

```
## Transformed variance:  0.05578973
```

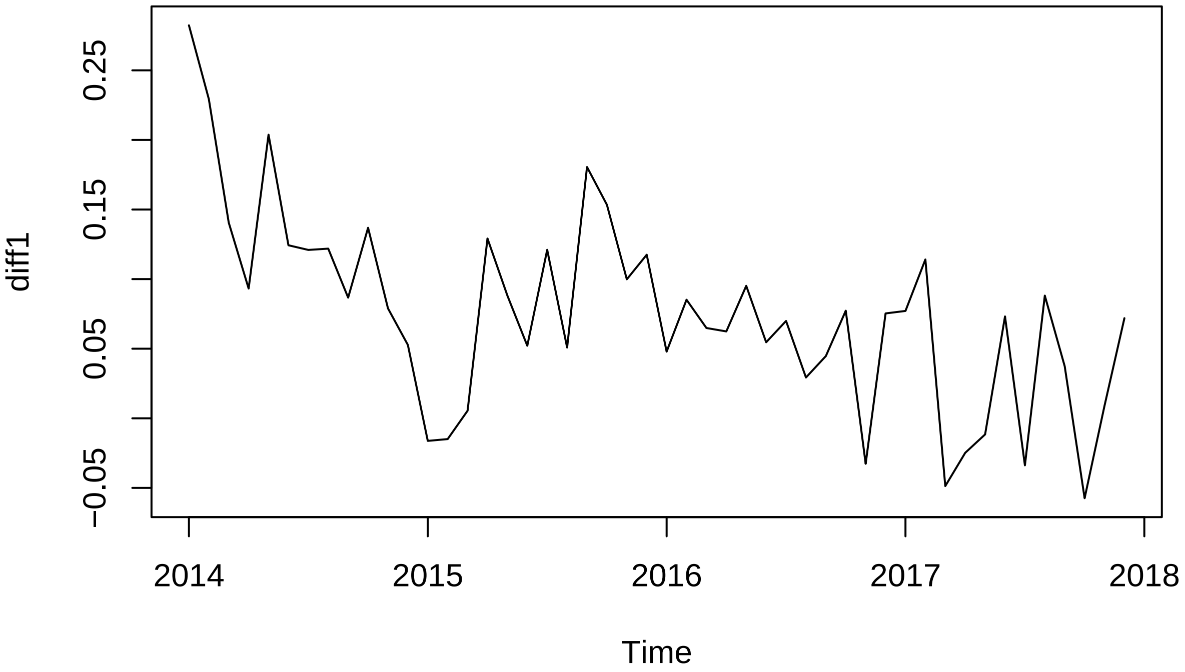
After transforming, we find that the variance of the time series has significantly decreased, implying that an appropriate transformation has been performed.

```
# Decomp graph
decomp <- decompose(i1.bc)
plot(decomp)
```



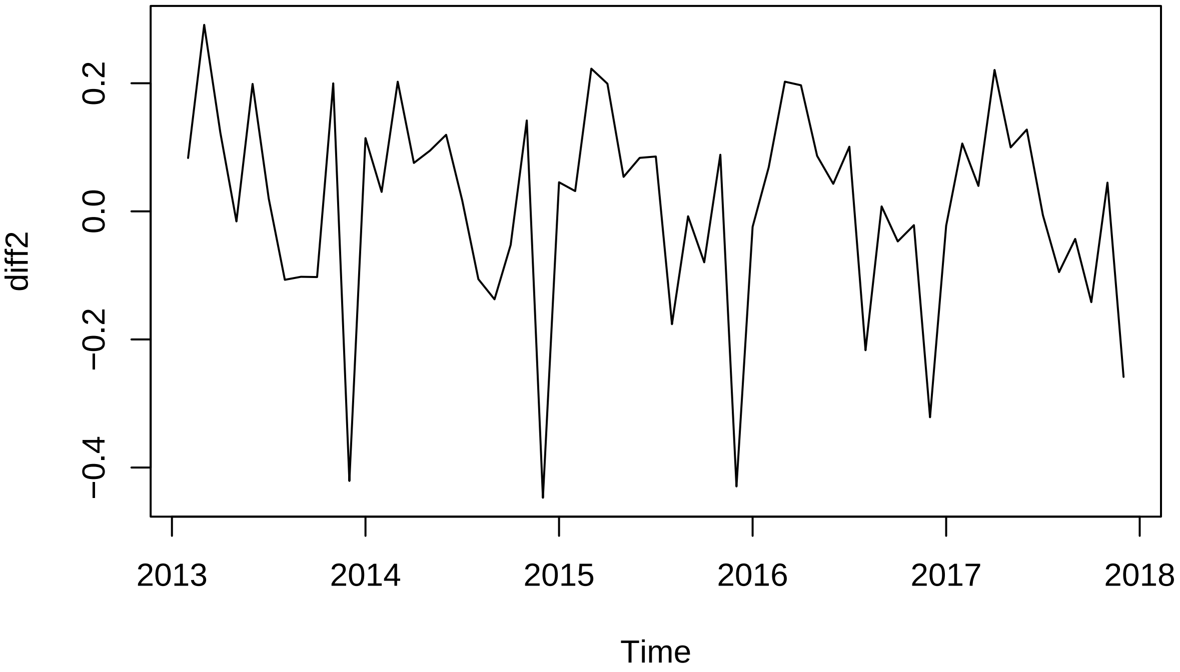
From the decomposition of our time series, there is clearly still a seasonal component. We hence attempt to make the time series by differencing at lags $k = 1$ and $k = 12$.

4. Differencing to Attain Stationarity



```
## [1] 0.004950946
```

It becomes clear that after this differentiating at lag $k = 12$, seasonality is removed. We difference again to remove trend. One thing to note however, is that variance did increase, but the clear trend removal supersedes

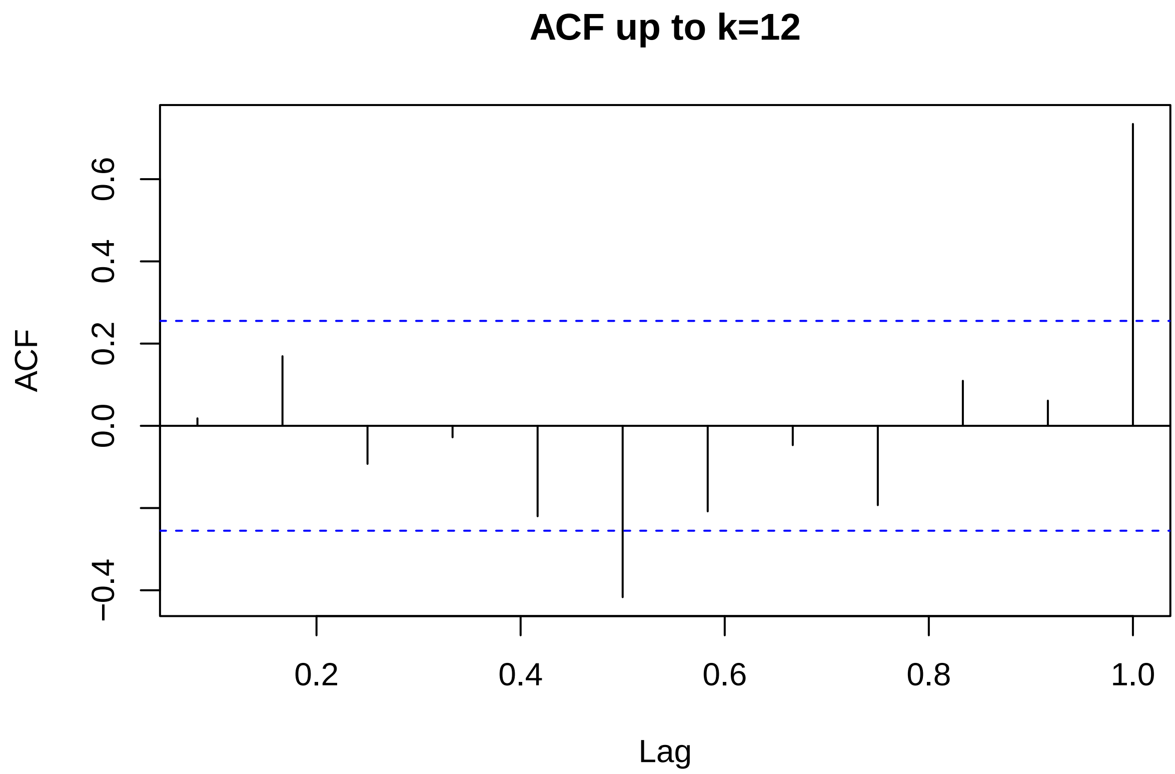


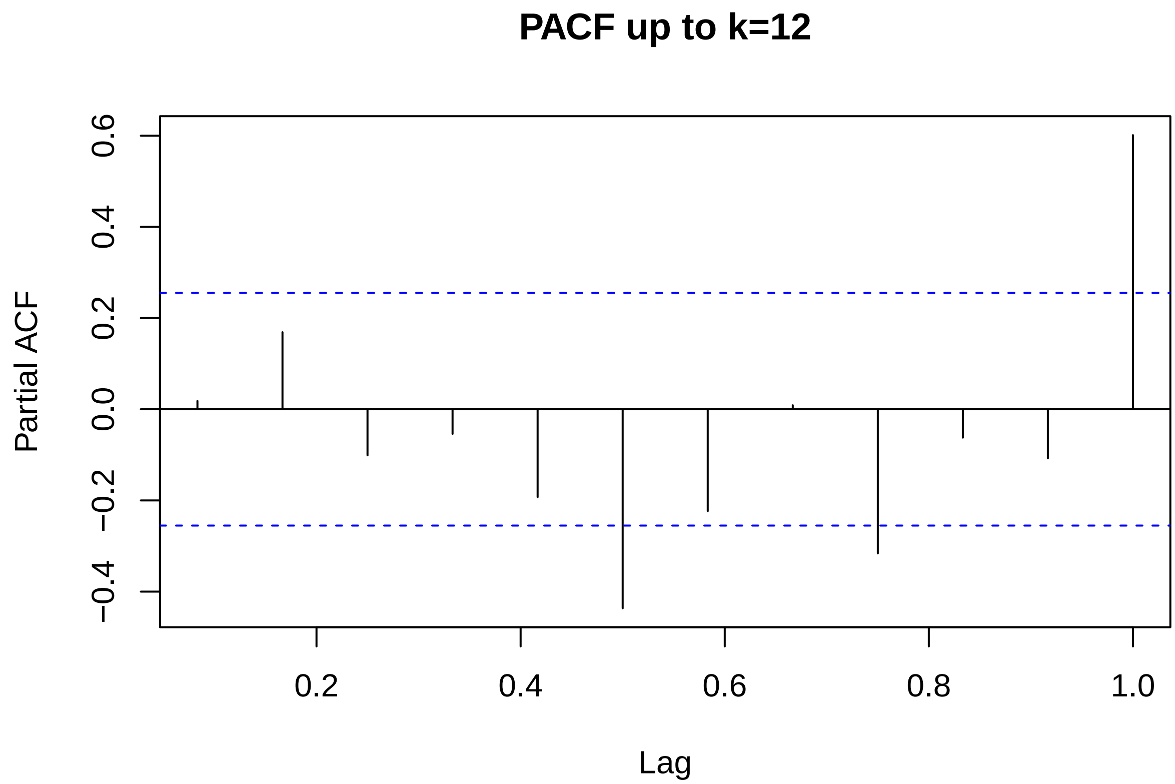
this.

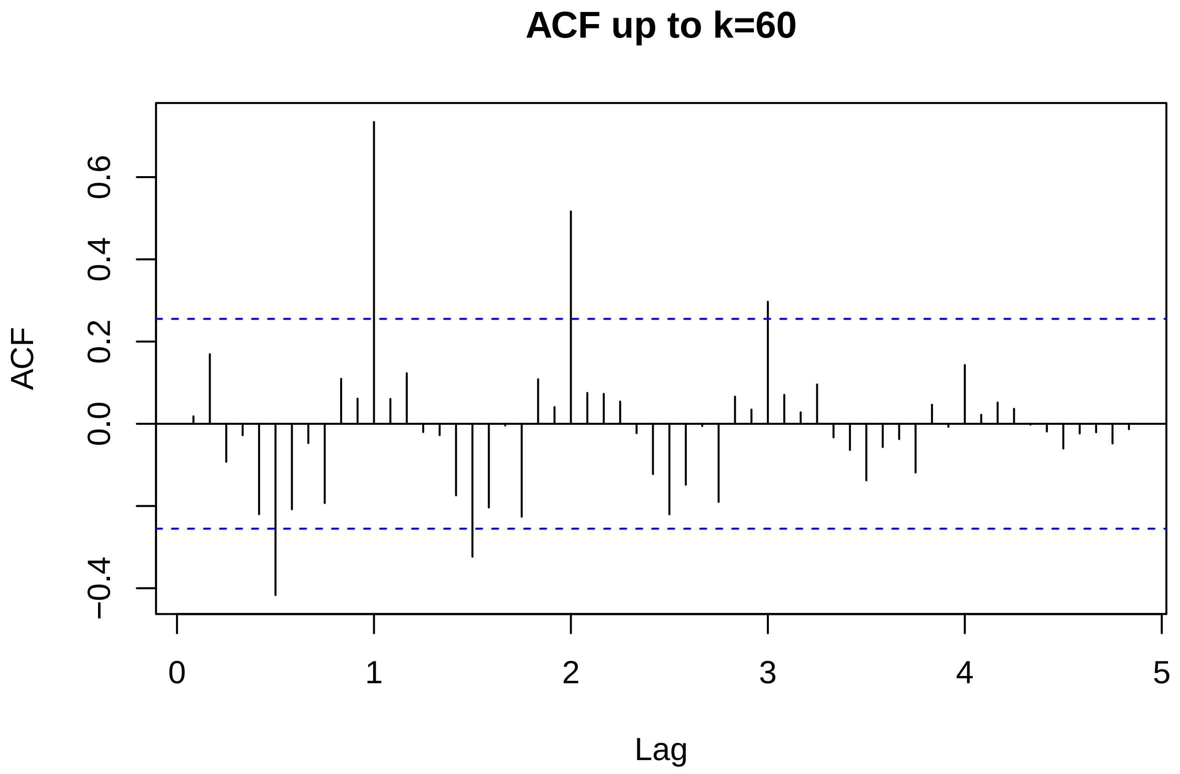
```
## After differencing lag 1: 0.02612055 Before:  0.004950946
```

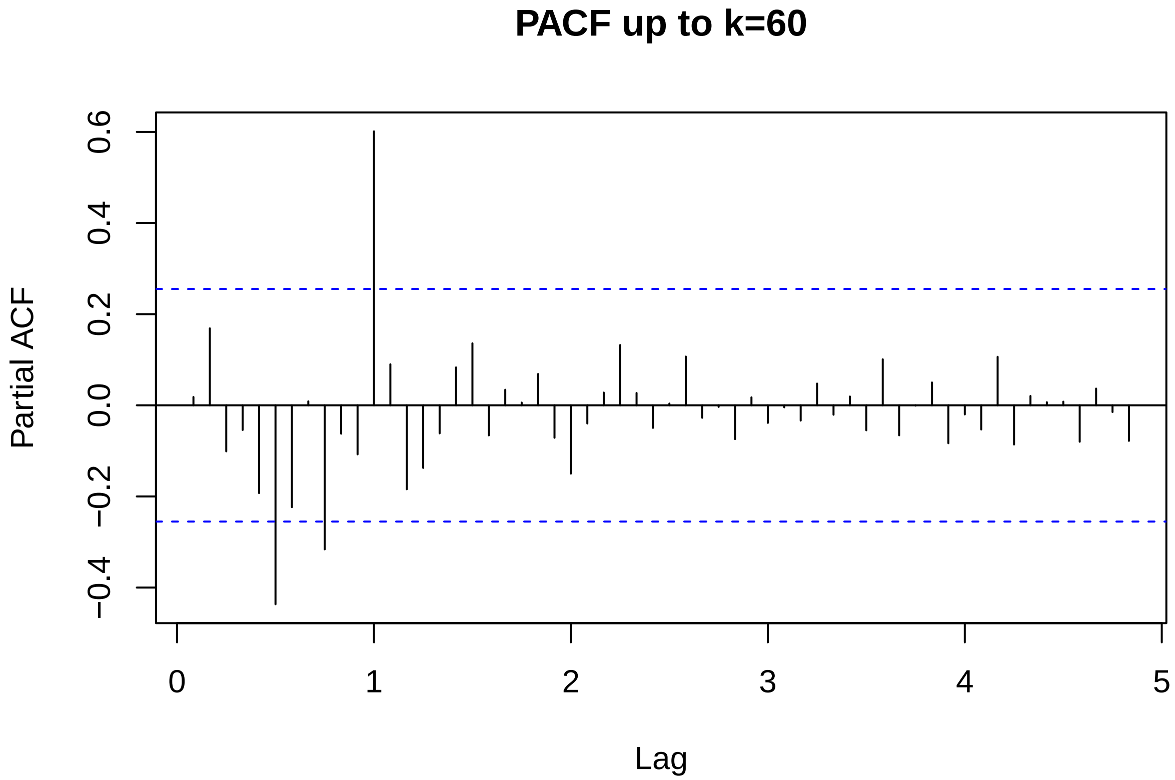
After differencing at lag k = 1 to remove trend, we can see that variance has further decreased.

5. ACF and PACF






ACF up to k=60
ACF
0.6
0.4
0.2
0.0
−0.4
0
1
2
3
4
5
Lag



Looking up to $k = 60$, there seemingly seems to be a pattern whereby every lag of 12 leads to a significant ACF/PACF. This leads us to suspect that $s = 12$ for our SARIMA model. ACF cuts off at $k = 12$, leading us to conclude that with $s = 12$, it is possible that we have a $SMA(1)$. PACF cuts off at $k = 12$, leading us to conclude that it is possible that we have a $SAR(1)$, or $SAR(0)$.

Looking at max lag $k = 12$, we see that the ACF only has last non-consecutive non-zero value at $k = 6$, hence we conclude that there is a $MA(0)$ component, but suspect that there is a possible $MA(1)$ or greater component. PACF has a significant peak $k = 6$, but it has values of 0 till the peak. Hence, we conclude that the model has a $AR(0)$ component.

We hence suspect that the model is $SARIMA(0, 1, 1)(1, 1, 1)_{12}$ or $SARIMA(0, 1, 1)(0, 1, 1)_{12}$

6. Parameter Estimation

We use Maximum Likelihood Estimation (MLE) to train our SARIMA model based on the orders we've observed.

```
##
## Call:
## arima(x = diff2, order = c(p, d, q), seasonal = seasonal_order, method = "ML")
##
## Coefficients:
##           ma1    sar1     sma1
##       -0.9999  0.8836  -0.9962
## s.e.   0.0692  0.2555   1.1347
##
## sigma^2 estimated as 0.004313:  log likelihood = 56.8,  aic = -107.59
```

```
##
## Call:
## arima(x = diff2, order = c(p, d, q), seasonal = seasonal_order2, method = "ML")
##
## Coefficients:
##           ma1     sma1
##       -1.0000  -0.0163
## s.e.   0.0674   0.2015
##
## sigma^2 estimated as 0.004575:  log likelihood = 56.7,  aic = -109.39
```

In order to determine the statistical significance of each parameter at the 95% significance level, we can build confidence intervals as such:

$$x \pm 1.96 S.E$$

For model 1, we find that 0 is within the confidence interval of the `sma1` parameter, and hence we remove it from our equation, attaining the final model:

$$(1 + 0.9999B)(1 - B)(1 - B^{12})X_t = (1 + 0.8836B^{12})Z_t$$

For model 2, we find that 0 is within the confidence interval of the `sma1` parameter, and hence we remove it from our equation, attaining the final model:

$$(1 + 1.0B)(1 - B)(1 - B^{12})X_t = Z_t$$

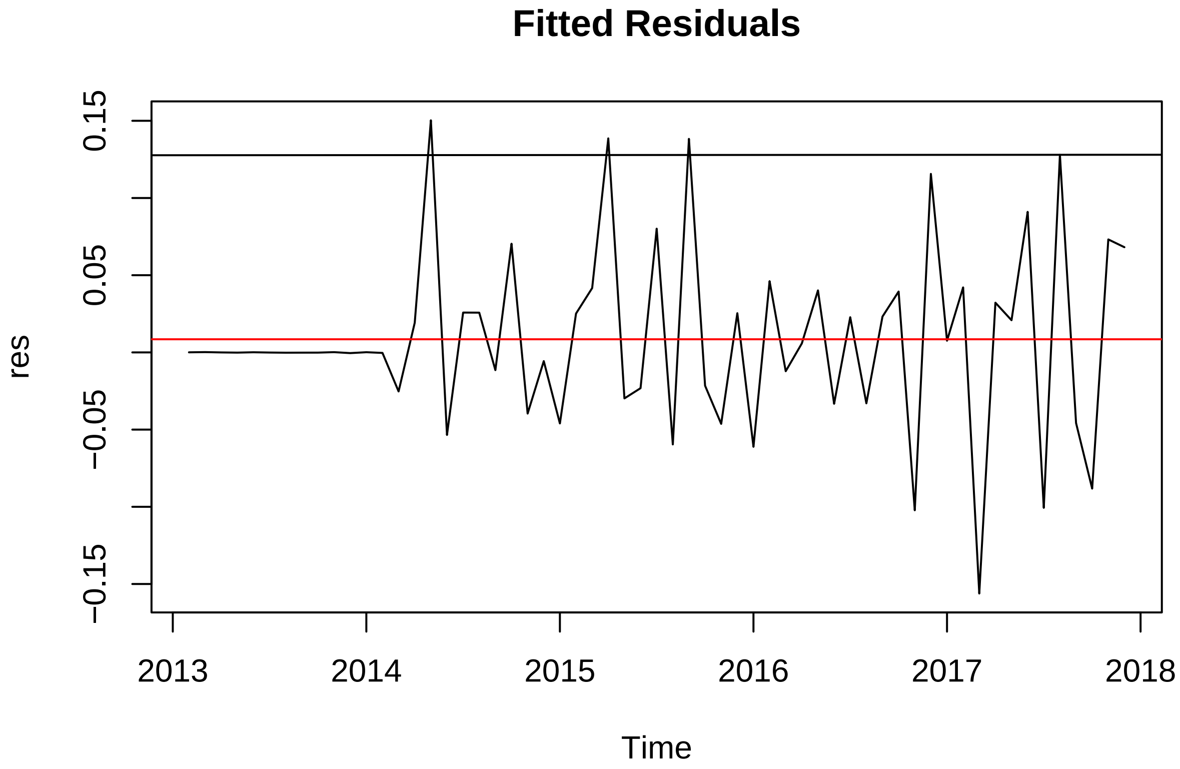
```
## Model 1 AICc:  -106.9542 Model 2 AICc:  -109.171
```

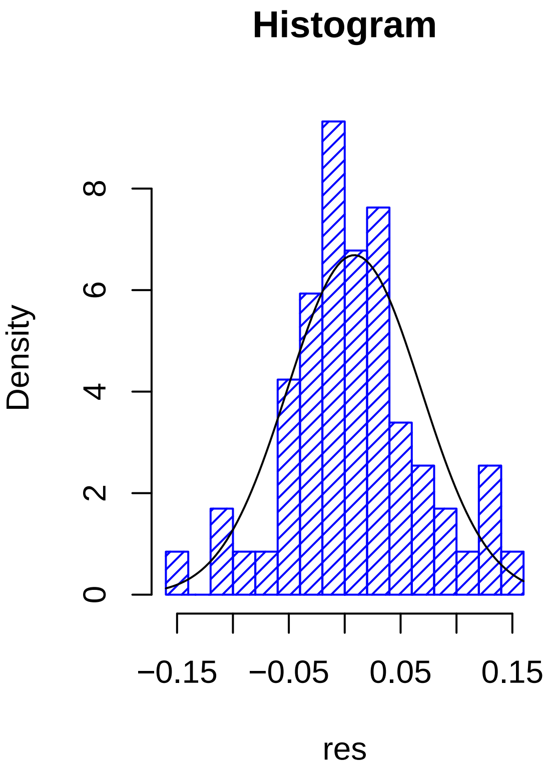
Just to further double check, we check for their Akaike Information Crtieria (AICc) to determine if which of the two models are better.

Despite the first model having a lower AICc than our model 2, by the Principal of Parsimony, we pick model 2 considering that AICc overestimate p, and we would prefer less parameters.

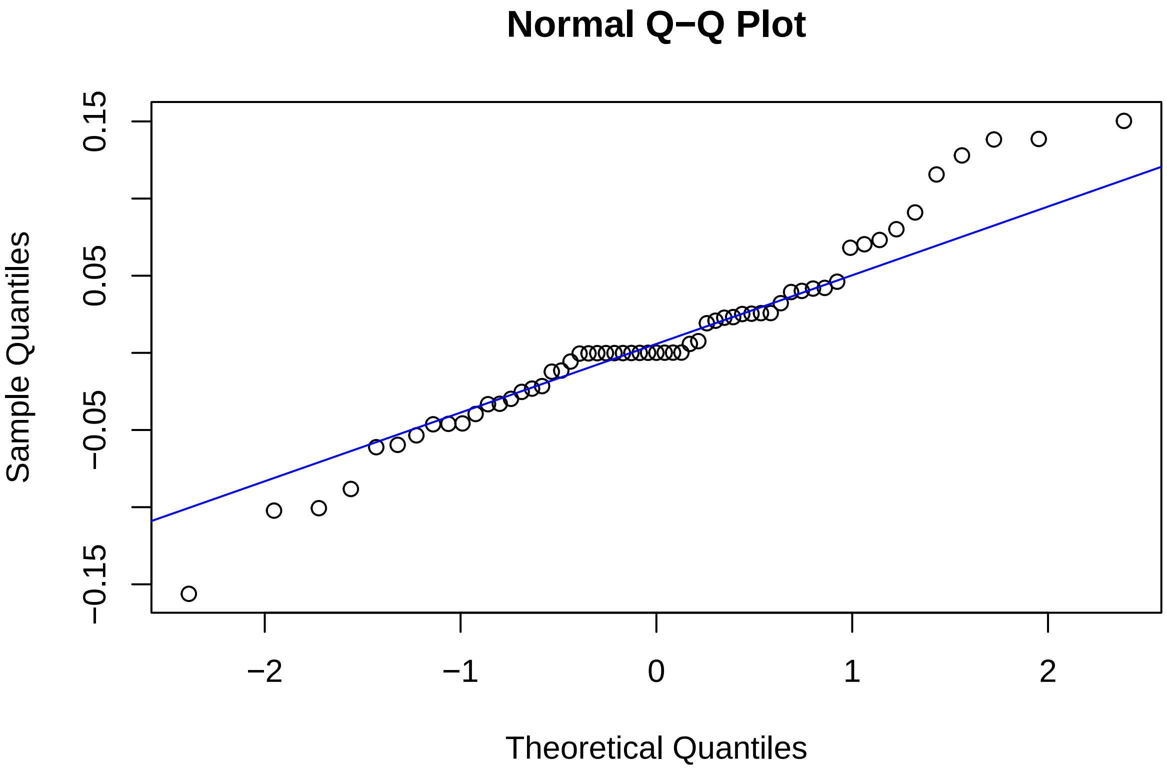
7. Diagonistic Checking

7.1 Residual Plots





7.1 Normality Test



From the QQ-plot, we can see that the residuals look relatively normal. However, there are extreme values on the tail ends, implying a possible heavy tail distribution.

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.96336, p-value = 0.0729
```

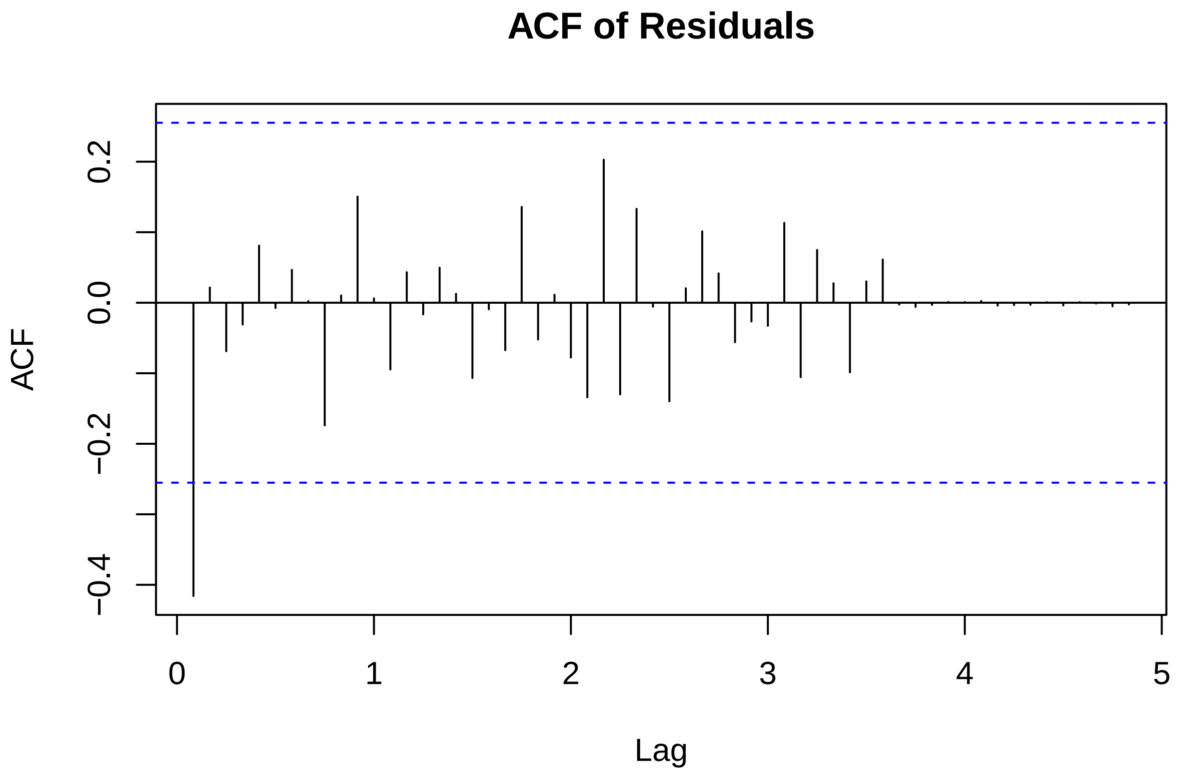
To perform a formality check, we perform the Shapiro-Wilk test.

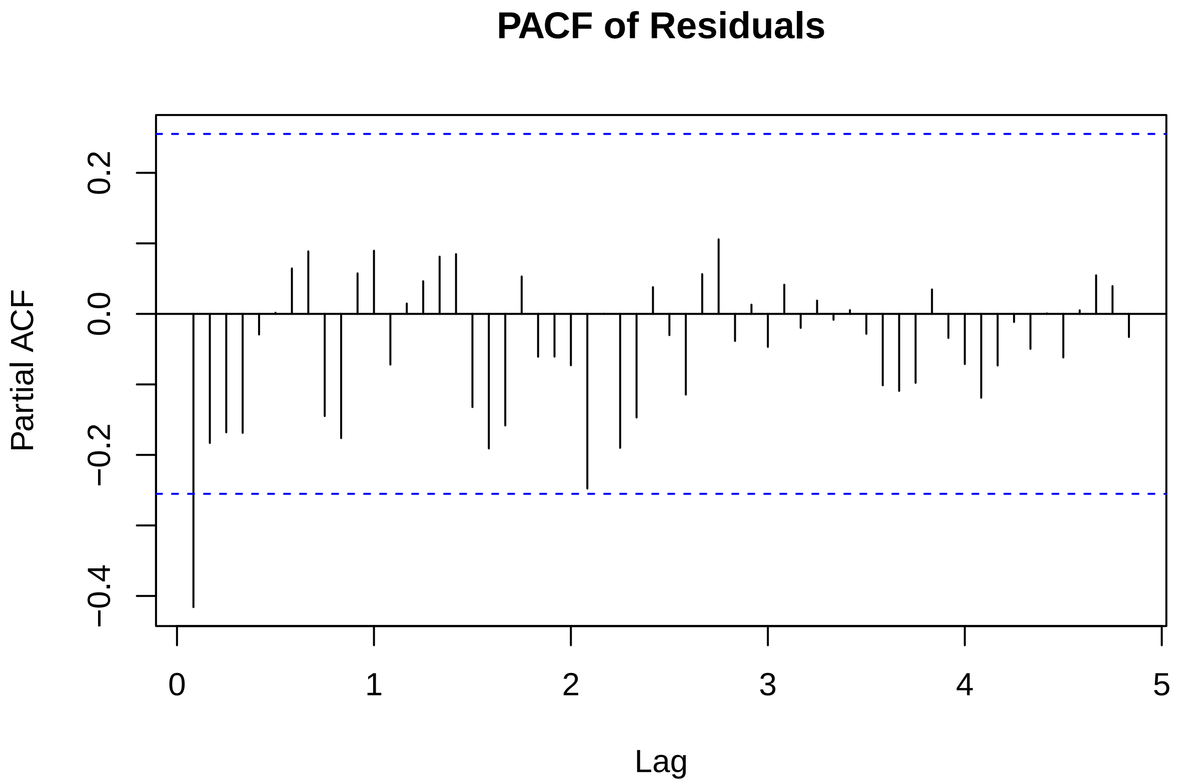
$$H_0: \text{ The data follows a normal distribution}$$

$$H_A: \text{ The data does not follow a normal distribution}$$

Since the $p > 0.05$, we don't reject the null hypothesis at the 95% significance level.

7.2 ACF/PACF of Residuals





There seems to be autocorrelation as indicated by ACF and PACF of lag 1. This seems to imply that we have $AR(1)$ and $MA(1)$ components. We conduct the Box tests to look at this problem from a statsitical perspective.

7.3 Portmanteau Statistics

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 11.084, df = 6, p-value = 0.0858
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 11.718, df = 6, p-value = 0.06856
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 6.9805, df = 6, p-value = 0.3227
```

For the Box-Pierce, and Ljung-Box tests, and the Mcleod-Li test, we obtain a p-value > 0.05, of which we can conclude that the squares of the residuals are uncorrelated, and that the residuals resemble Gaussian WN(0,1).

7.4 Diagnosising Correlated ACF/PACF of Residuals

However, we take note of the ACF and PACF problems. We might hence have to increase the orders of our existing model to fix this. We focus on trying to attain higher p and q orders.

```
## p= 0 q= 0 -66.39711
## p= 0 q= 1 -107.1775
## p= 0 q= 2 -117.08
## p= 0 q= 3 -114.9317
## p= 0 q= 4 -112.5465
## p= 0 q= 5 -110.5508
## p= 0 q= 6 -107.9777
## p= 1 q= 0 -89.88161
## p= 1 q= 1 -112.5159
## p= 1 q= 2 -114.9313
## p= 1 q= 3 -112.5414
## p= 1 q= 4 -110.1563
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, xreg = xreg,
## : possible convergence problem: optim gave code = 1
```

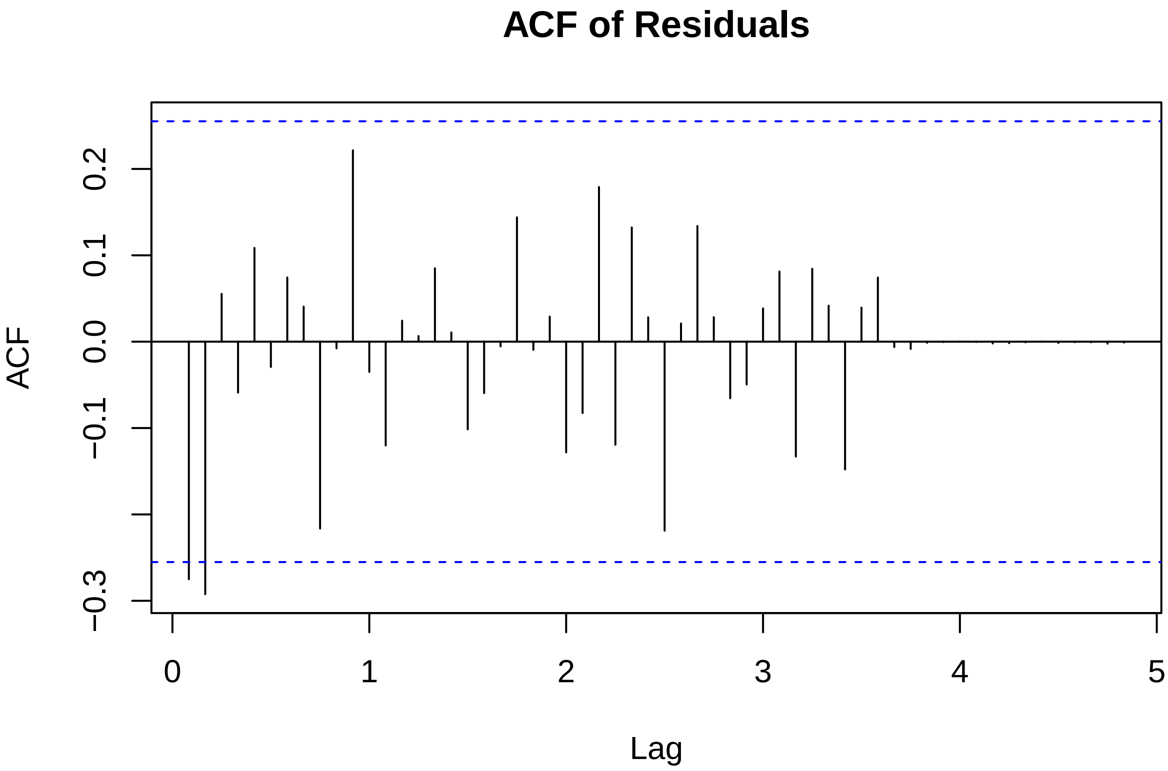
```
## p= 1 q= 5 -107.8523
## p= 1 q= 6 -105.3762
```

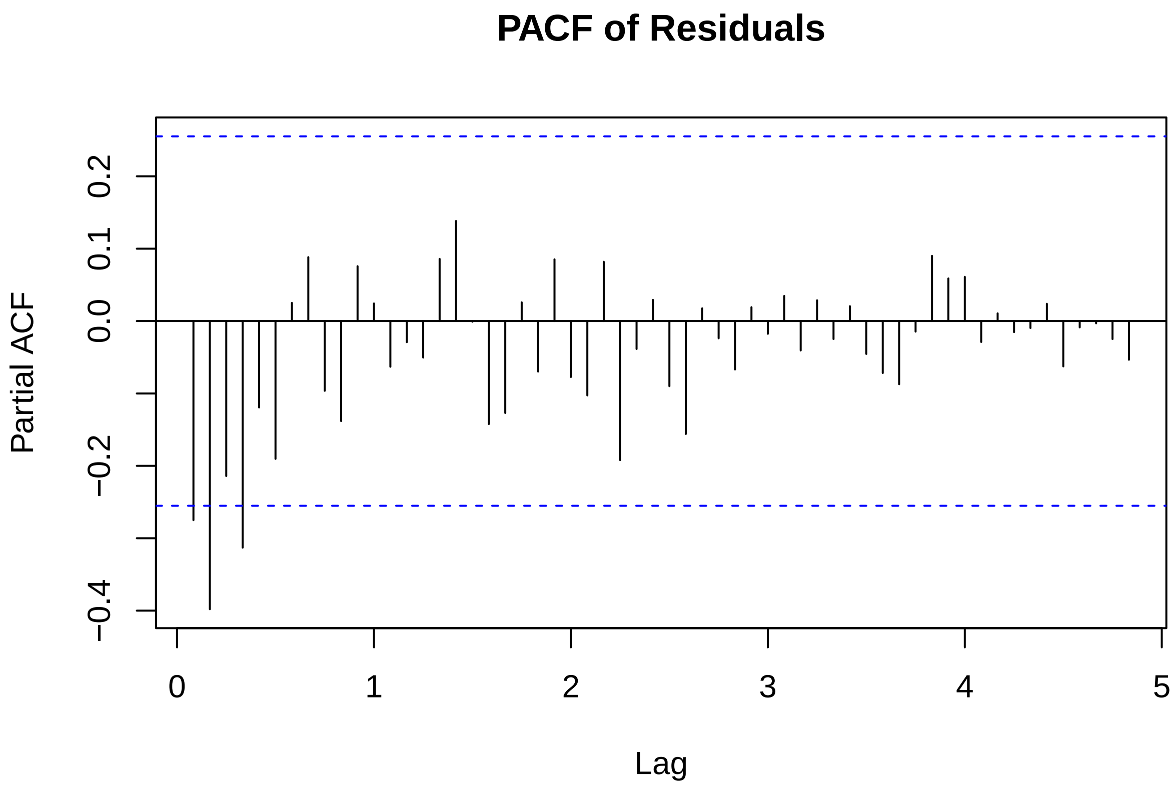
We see the following pairs of (p, q) - $p = 1, q = 0$ -89.88161 - $p = 0, q = 0$ -66.39711 We check the model with $p = 1, q = 0$ first.

```
##
## Call:
## arima(x = diff2, order = c(1, 1, 0), seasonal = seasonal_order2, method = "ML")
```

```
##
## Coefficients:
##           ar1     sma1
##       -0.6460  -0.0672
## s.e.   0.1085   0.2181
##
## sigma^2 estimated as 0.007156:  log likelihood = 48.05,  aic = -92.1
```

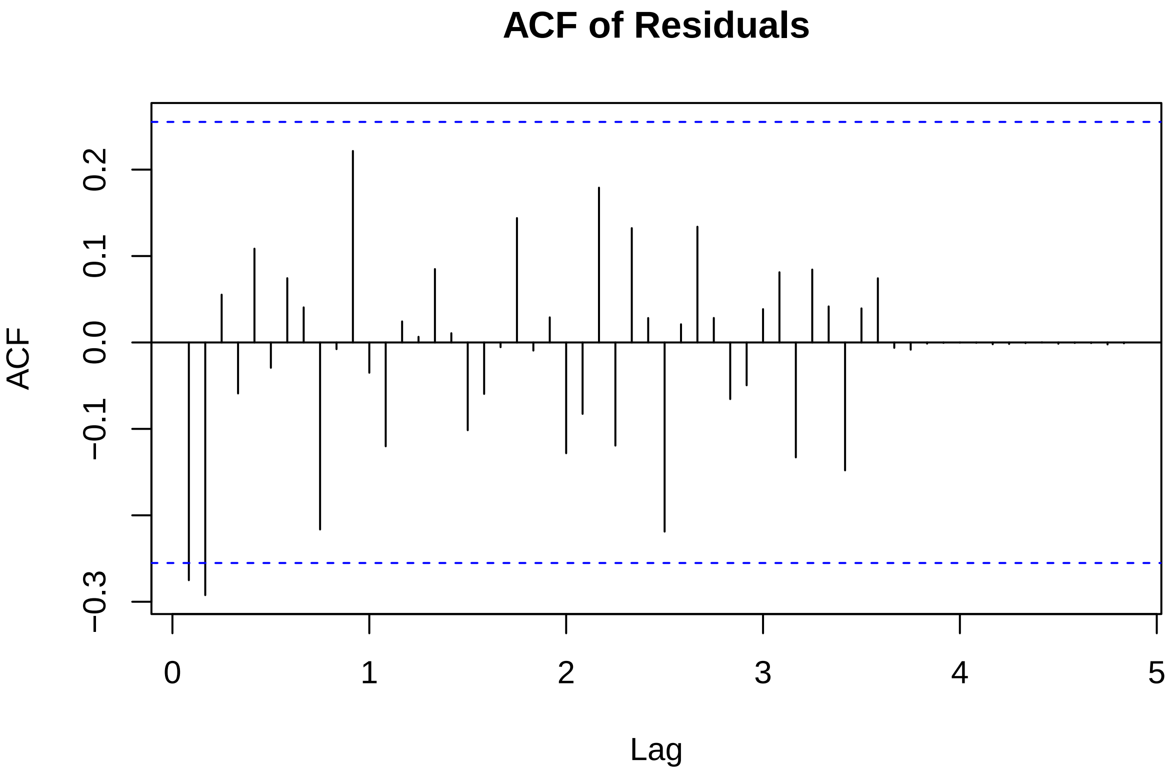
We remove the coefficient with 0 within its 95% CI and conduct model diagnostics again.

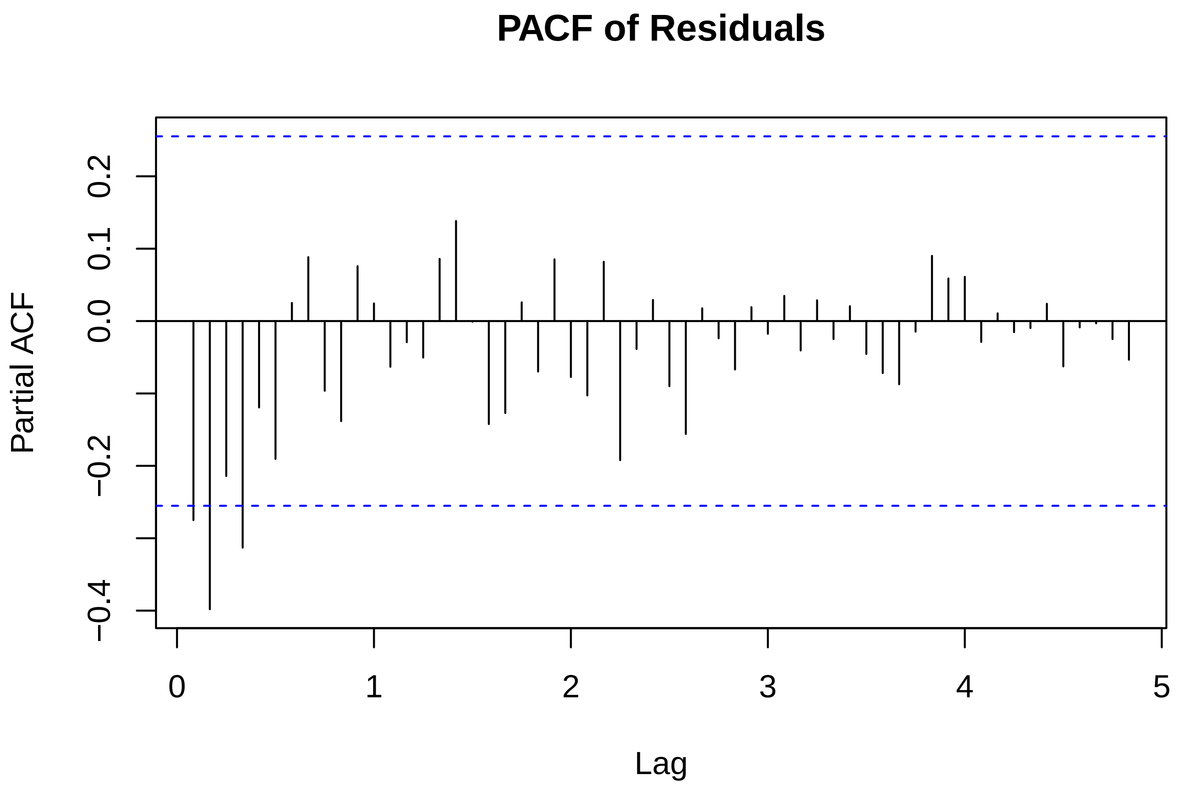




As the model seems to have worse ACF/PACF of residuals, we use the model with $p = 0, q = 0$.

```
##
## Call:
## arima(x = diff2, order = c(0, 1, 0), seasonal = seasonal_order2, method = "ML")
##
## Coefficients:
##          sma1
##       -0.0524
## s.e.   0.1831
##
## sigma^2 estimated as 0.01265:  log likelihood = 35.23,  aic = -68.47
```

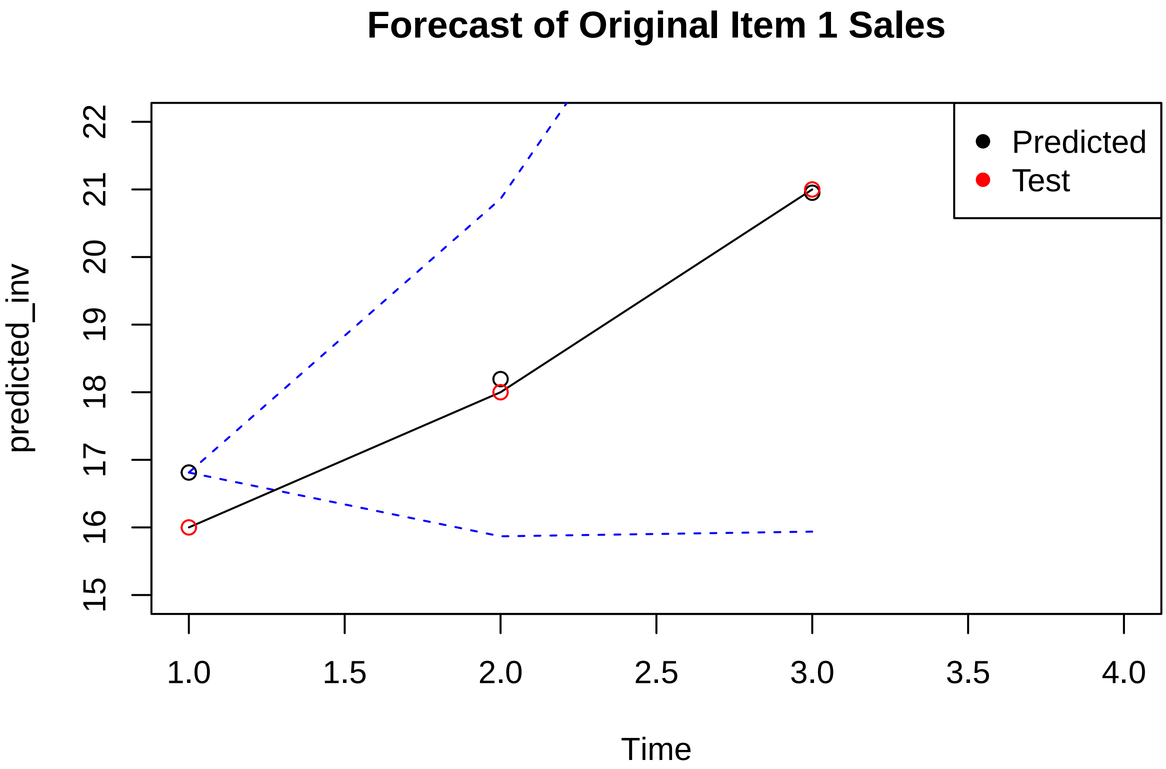

ACF of Residuals
ACF
0.2
0.1
0.0
−0.1
−0.3
Lag
0
1
2
3
4
5

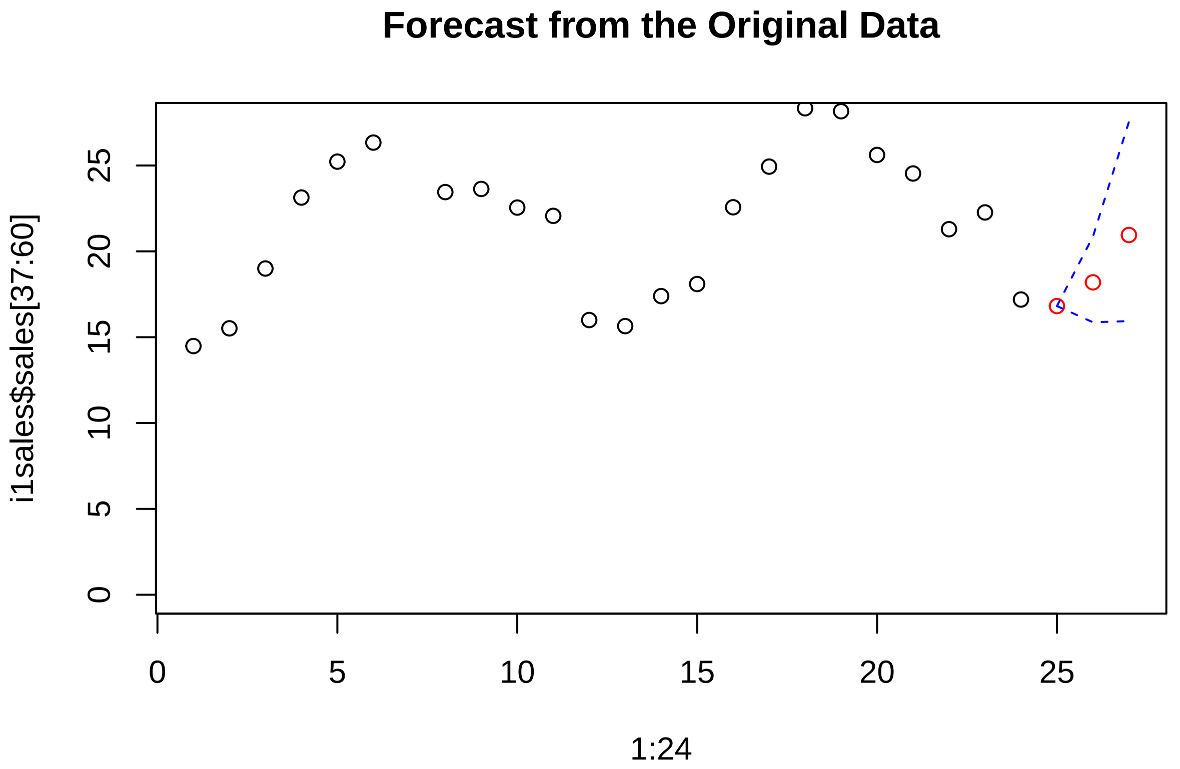


The lags don't improve with the newer models, and hence our hypothesis that increasing orders to remove autocorrelation is wrong. We hence use the initial model $SARIMA(0,1,1)(0,1,1)_{12}$, keeping in mind that the residuals are possibly autocorrelated at lag $k = 1$.

8. Forecasting

Here, we make our forecasts and back transform the data by inverse differencing at lag 1, lag 12, and applying inverse log before plotting our predictions and their confidence intervals.





From this plot, we can see that our 3-months ahead forecasted values are fairly accurate. Apart from the first value, the rest of the test data is also within our confidence interval.

9. Conclusion

The objective of this paper was to forecast the average 3-months ahead sales of Item 1 across different grocery stores using daily historical data. In order to do this, we settled on the SARIMA model $SARIMA(0,1,1)(0,1,1)_{12}$, using Boxcox transformed data that was also differenced at lags $k = 12$ and $k = 1$ to remove seasonality and trend.

The transformed data used to train the model passed almost all residual diagnostics, though PACF/ACF of the residuals show autocorrelation. This was in contrast to the Box tests, which concluded that the transformed data follows a Gaussian WN distribution.

The model was relatively accurate at forecasting the sales of Item 1 3 months into the future, although a larger test set would be ideal in further evaluating it.

10. References

- Dr. Raya Feldman, UCSB PSTAT 174/274, Time Series Lecture Notes
- Kaggle.
- R Documentation, Last Accessed 08/12/2023

Appendix

The R code used in this report is as shown

```r
# Libraries
library(dplyr)
library(lubridate)
library(forecast)
library(MASS)
library(tseries)
library(ggplot2)
library(TSA)
library(qpcR)
library(ggfortify)
library(UnitCircle)

# Reading data
data <- read.csv('train.csv')
# just get date and sales data for store 1 and item 1
i1sales <- subset(data, store == 1 & item == 1, select = c(date, sales))

i1sales$date <- as.Date(i1sales$date)

i1sales <- i1sales %>%
  mutate(month = floor_date(date, "month"))

i1sales <- i1sales %>%
  group_by(month) %>%
  summarise(sales = mean(sales, na.rm = TRUE))


# Converting to time series
i1sales.ts <- ts(i1sales$sales, start=c(2013,1,1), frequency=12)

# Finding Lambda
t <- 1:length(i1sales.ts)
transformed <- boxcox(i1sales.ts~t, plotit=TRUE)

# Transforming item 1 sales data
i1.bc = log(i1sales.ts)
plot.ts(i1.bc, main="Transformed Time Series of Item 1 Sales ")

# Finding variance
og_var <- var(i1sales.ts)
new_var <- var(i1.bc)
cat("Original variance: ", og_var, "\n")
cat("Transformed variance: ", new_var)

# Decomp graph
decomp <- decompose(i1.bc)
plot(decomp)

# Diff at lag 12
diff1 <- diff(i1.bc, lag=12)
plot.ts(diff1)
```

```r
var(diff1)

# DIff at lag 1
diff2 <- diff(i1.bc, lag=1)
plot.ts(diff2)
cat("After differencing lag 1:", var(diff2), "Before: ", var(diff1))

# ACF and PACF of Time Series
acf(diff2, lag.max=12, main="ACF up to k=12")
pacf(diff2, lag.max=12, main="PACF up to k=12")

acf(diff2, lag.max=60, main="ACF up to k=60")
pacf(diff2, lag.max=60, main="PACF up to k=60")

# Parameter Estimation of 2 models
p <- 0  # AR order
d <- 1  # Differencing order
q <- 1  # MA order
P <- 1  # Seasonal AR order
D <- 1  # Seasonal differencing order
Q <- 1  # Seasonal MA order
s <- 12  # Seasonal period (e.g., for monthly data)

seasonal_order <- c(P, D, Q, s)
seasonal_order2 <- c(0, D, Q, s)

# Model 1: SAR(1) component
sarima_model <- arima(diff2, order = c(p, d, q), seasonal = seasonal_order, method="ML")

# Model 2: SAR(0) component
sarima_model2 <- arima(diff2, order = c(p, d, q), seasonal = seasonal_order2, method="ML")

sarima_model
sarima_model2

# Removal of SMA1 parameter of selected Model 2
sarima_model <- arima(diff2, order = c(p, d, q), seasonal = seasonal_order, method="ML", fixed=c(NA, NA

sarima_model2 <- arima(diff2, order = c(p, d, q), seasonal = seasonal_order2, method="ML", fixed=c(NA, 

cat("Model 1 AICc: ", AICc(sarima_model), "Model 2 AICc: ", AICc(sarima_model2))

# Model 2 Residual Analysis
res <- residuals(sarima_model2)
mu <- mean(res)
sigma_sq <- var(res)
par(mfrow=c(1,1))
ts.plot(res,main = "Fitted Residuals")
t = 1:length(res)
fit.res = lm(res~t)
abline(fit.res)
abline(h = mu, col = "red")
par(mfrow=c(1,2))
```

```r
hist(res, density = 20, breaks = 20, main = "Histogram", col = 'blue', prob = T)
curve( dnorm(x, mu, sqrt(sigma_sq)), add=TRUE )

# QQ Plot
qqnorm(res)
qqline(res, col='blue')

# Shapiro-Wilk Normality test
shapiro.test(res)

# ACF/PACF of residuals
acf(res, main="ACF of Residuals", lag.max=60)
pacf(res, main="PACF of Residuals", lag.max=60)

# PORTMANTEAU Stats
# h as the square root of n
h = floor(sqrt(length(diff2)))

# df is the number of parameters
df = 1
Box.test(res, lag = h, type = c("Box-Pierce"), fitdf = df)
Box.test(res, lag = h, type = c("Ljung-Box"), fitdf = df)
Box.test(res^2, lag = h, type = c("Ljung-Box"), fitdf = df)

# Iterating through possible ps and qs for better model
for (i in 0:1) {
  for (j in 0:1) {
    cat('p=', i, 'q=', j, AICc(arima(diff2, order=c(i, 1, j), seasonal = seasonal_order2, method="ML"))
  }
}

# Training first possible better model 1 + analysis
second_model <- arima(diff2, order=c(1, 1, 0), seasonal=seasonal_order2, method="ML")
second_model

second_model <- arima(diff2, order=c(1, 1, 0), seasonal=seasonal_order2, method="ML", fixed=c(NA, 0))

res <- residuals(second_model)

acf(res, main="ACF of Residuals", lag.max=60)
pacf(res, main="PACF of Residuals", lag.max=60)

# Training first possible better model 2 + analysis
second_model <- arima(diff2, order=c(0, 1, 1), seasonal=seasonal_order2, method="ML")
second_model

second_model <- arima(diff2, order=c(1, 1, 0), seasonal=seasonal_order2, method="ML", fixed=c(NA, 0))
res <- residuals(second_model)

acf(res, main="ACF of Residuals", lag.max=60)
pacf(res, main="PACF of Residuals", lag.max=60)

# Loading test dataset
```

```r
datatest <- read.csv('test.csv')
i1test <- subset(datatest, select = c(date, sales))
i1test$date <- as.Date(i1test$date)
i1test.ts <- ts(i1test$sales, start=c(2018,1,1), frequency=12)
i1test.ts

# Predicting
library(forecast)
pred.tr <- predict(sarima_model2, n.ahead=3)

# Getting results
predicted <- pred.tr$pred

# CI
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se

# Inverse transformation of CI and Forecasted Values

# Inverse diff at lag 1
predicted_inv <- cumsum(c(diff1[length(diff1)], predicted))
U <- cumsum(c(diff1[length(diff1)], U.tr))
L <- cumsum(c(diff1[length(diff1)], L.tr))

# Inverse diff at lag 12
bc_1 = i1.bc[length(i1.bc) - 11]
bc_2 = i1.bc[length(i1.bc) - 10]
bc_3 = i1.bc[length(i1.bc) - 9]
changed1 = bc_1 + predicted_inv[1]
changed2 = bc_2 + predicted_inv[2]
changed3 = bc_3 + predicted_inv[3]
predicted_inv <- c(changed1, changed2, changed3)

changed1 = bc_1 + U[1]
changed2 = bc_2 + U[2]
changed3 = bc_3 + U[3]
U <- c(changed1, changed2, changed3)

changed1 = bc_1 + L[1]
changed2 = bc_2 + L[2]
changed3 = bc_3 + L[3]
L <- c(changed1, changed2, changed3)


# Inverse log
predicted_inv <- exp(predicted_inv)
L <- exp(L)
U <- exp(U)

# Plotting preds and test data for the next 3 months
plot(1:3, predicted_inv, xlab='Time', xlim=c(1, 4), ylim=c(15, 22), main="Forecast of Original Item 1 Sa
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
lines(i1test$sales)
points(1:3, i1test.ts, col='red')
legend("topright", legend=c("Predicted", "Test"), col=c("Black", "Red"), pch=16)

plot(1:24, i1sales$sales[37:60], xlim=c(1, 27), ylim = c(0,max(U)), main="Forecast from the Original Dat
points(25:27, predicted_inv, col="red")
lines(25:27, U, lty=2, col="blue")
lines(25:27, L, lty=2, col="blue")
```

“‘