Kyle Salitrik

STAT 463

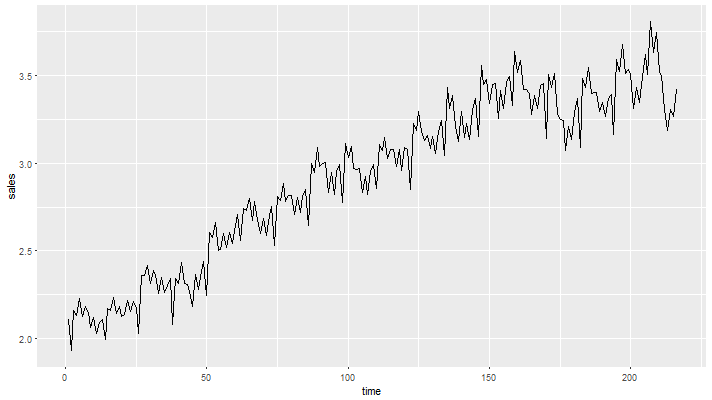


Figure 1: Sales Time Series Data

Final Project: Sales Trend Data Analysis

Upon initial examination of the sales data time series (Figure 1), there appears to be both a seasonality and mostly linear trend to the data, up until about 175 months. Examining the ACF and PACF (Figure 2) for the data reveals that a 1st or 2nd order AR model with a possible AR seasonality at 12 and 24 months. Working from the assumption that the seasonality is yearly, multiple trend estimation methods were used to calculate and remove the trend from the data.

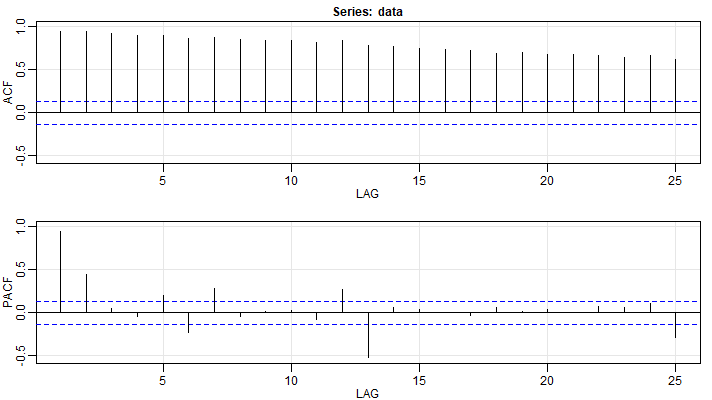


Figure 2: ACF and PACF of Sales Data

As shown in Figure 3, the da ta trend was calculated using Fractional Smoothing, Exponential Smoothing, Additive Decomposition, and Multiplicative Decomposition. These calculated trends, along with 12th and 1st differencing, were used to detrend the time series data and perform further analysis. Figure 4 displays these detrended time series plots. As one can see, the Fractional, Multiplicative, and Additive smoothing methods produced similar results, the Exponential smoothing seems to have faired the worst as it seems to have potentially tried to overfit the trend based on the trendline in Figure 3. Finally, examining the data detrended using the 12th and 1st differencing methods (displayed in Figure 5).

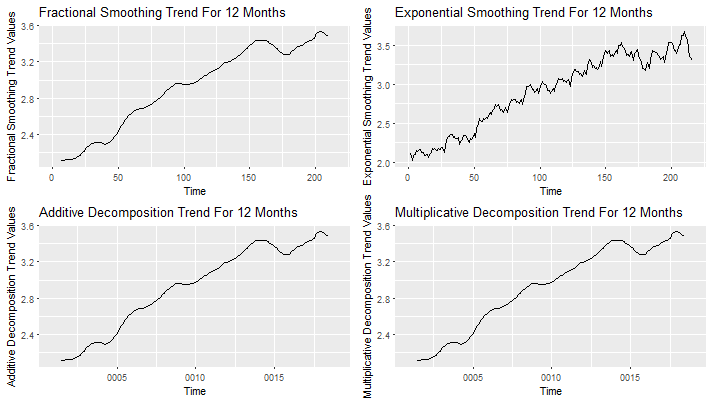


Figure 3: Data Trends Assuming 12 Month Seasonality

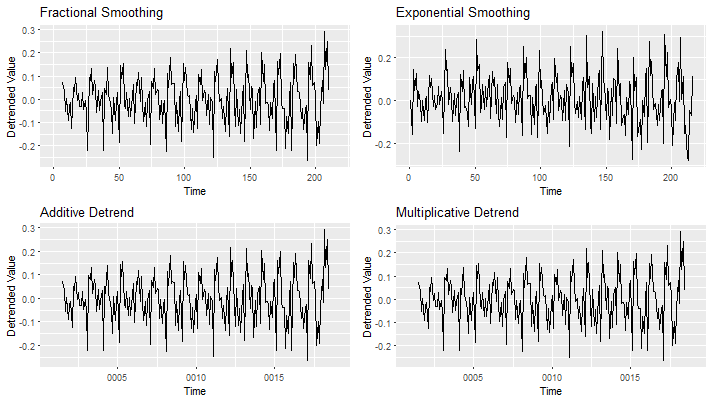


Figure 4: Detrended Data Assuming 12 Month Seasonality

The ACF and PACF of each of these detrended series were examined. The Fractional, Additive, and Multiplicative smoothing methods all produced very similar results; because of this only the series detrended using Additive Decomposition and the Differencing methods are displayed in Figures Figure 7 and Figure 6, respectively. From the ACF and PACF of the Additive Decomposition, there appears to be minor significant spikes in the early lags, but there is a significant spike at every year. The PACF shows significant spikes at 1 and 2 years. These patterns indicate potentially a SARIMA(1,1,1,2,1,4)[12] model. Examining the ACF plots for the Differenced Time Series, a potential model of SARIMA(1,1,0,1,1,0)[12] is indicated.

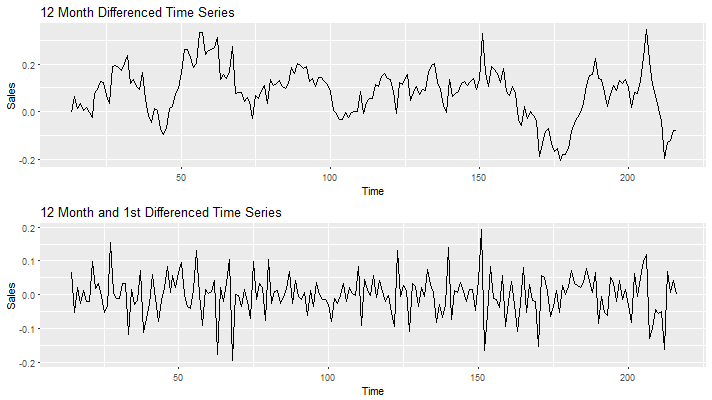


Figure 5: Differenced Time Series

These models were tested along with a thorough inspection of other similar models and the results for their selection criteria are listed in Table 1. The two best models were selected for performing predictions, shown in Figures Figure 8 and Figure 9. The prediction values are listed in the R Output Appendix. Looking back at the ACF plot for the time series detrended using Additive Decomposition (Figure 7), there appears to be a significant spike every 6 months, potentially indicating that there as a semiannual seasonality to the data instead of just purely annual.

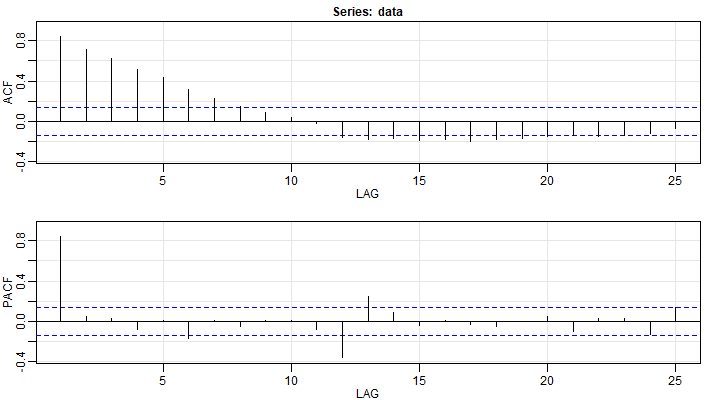


Figure 6: ACF and PACF of Differenced Time Series

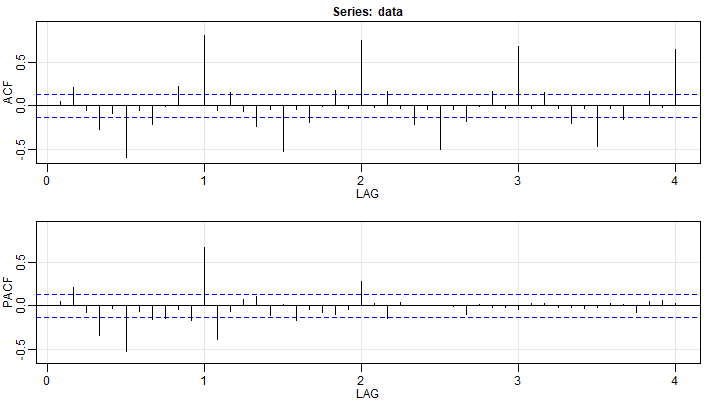


Figure 7: ACF and PACF of Additive Decomposition

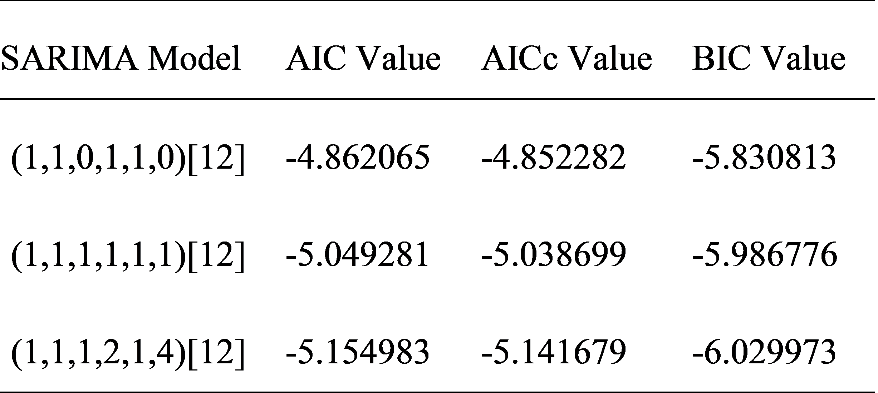
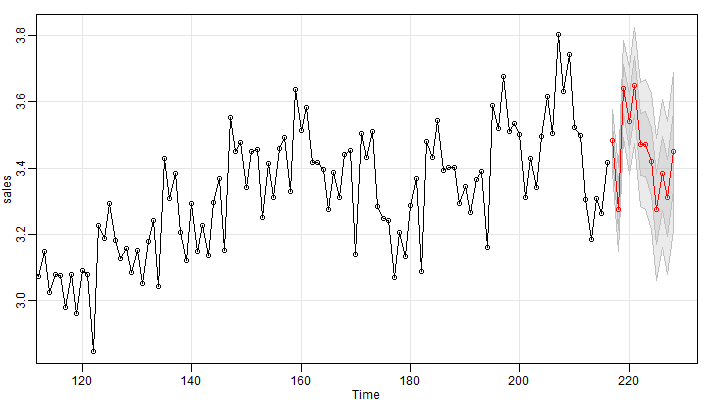
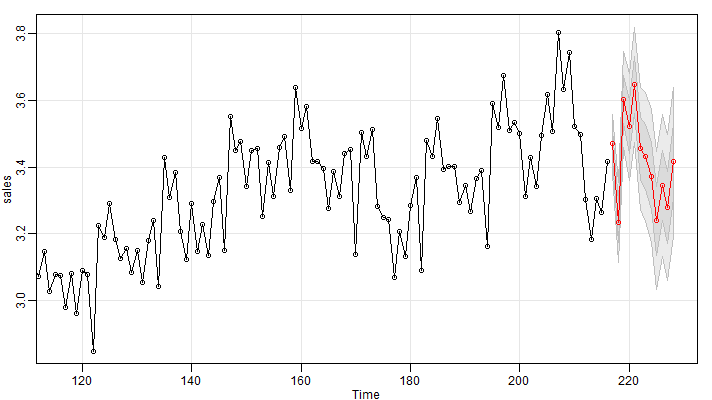


Table 1: 12 Month Seasonal ARIMA Model Selection Criteria

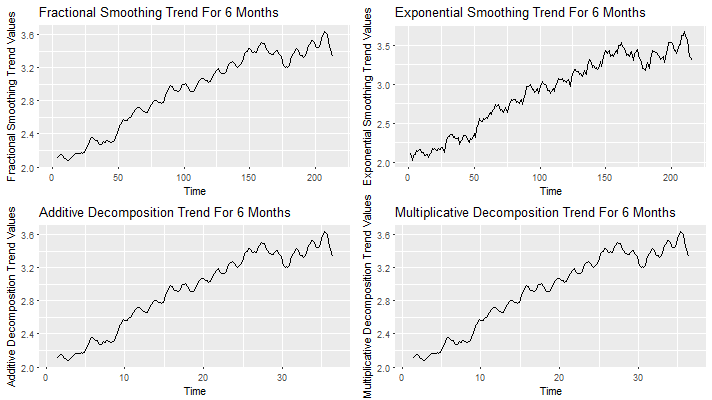


*Figure 8:* *Predictions for SARIMA(1,1,1,1,1,1)[12]*

The entire model analysis was repeated using a seasonal trend of 6 months. Figure 10 shows the trends that were obtained using the same repertoire of differencing methods while assuming a 6 month seasonality, while Figure 11 displays the differencing, applying a 6th difference followed by a 1st difference. Comparing to Figures 3 and 5, the trends fit the original data a bit better (possibly overfitting) and the differenced data is quite a bit noisier in the 6 month seasonality time series.



*Figure 9: Predictions for SARIMA(1,1,1,2,1,4)[12]*



*Figure 10: Data Trends Assuming 6 Month Seasons*

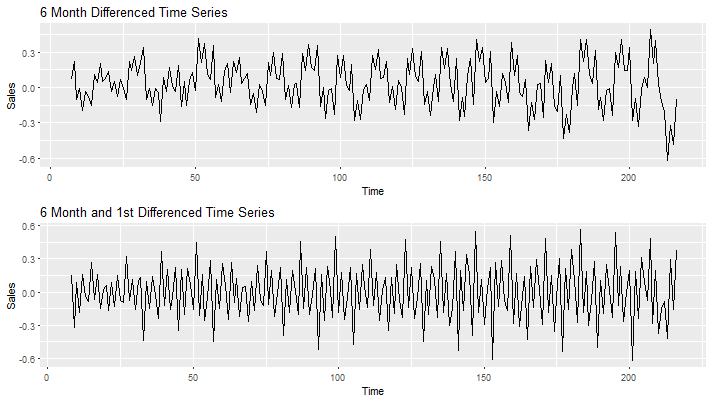
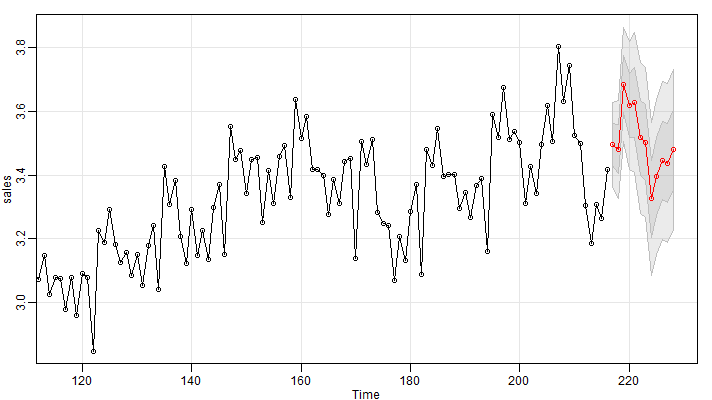


Figure 11: Differenced Time series Data

The same ACF and PACF plots were examined using the 6 month trend (shown in Figures 12 and 13). For the differenced data, Figure 13 shows a possible SARIMA(1,1,1,1,1,4)[6], and the ACF/PACF of the data detrended using the Additive Decomposition (Figure 12) trend shows a potential model of SARIMA(1,0,1,2,1,4)[6]. However, when attempting to fit the SARIMA models in R, an error of non-stationarity for the Seasonal AR component was encountered. Due to this, the AR component was set to 0 for all models tested. The best models are summarized in Table 2. Finally, predictions were made for these models, shown in Figures 14 and 15 with their outputs, again, in the R Output Appendix. All R code used is located in the Code Appendix.



*Figure 14: Predictions for SARIMA(1,1,1,0,1,4)[6]*

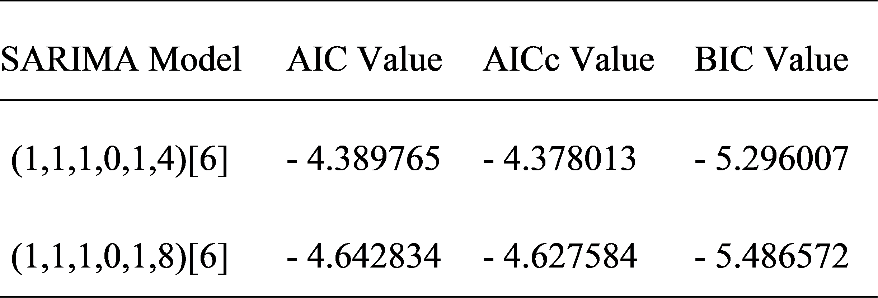
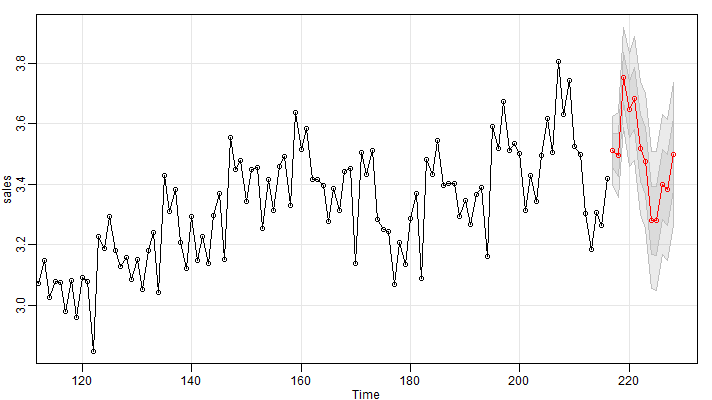
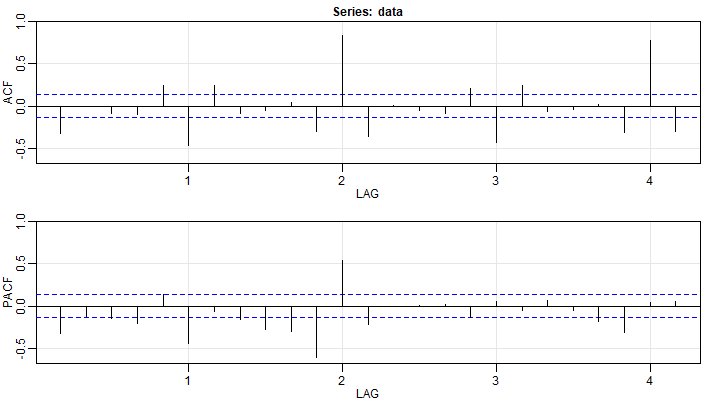


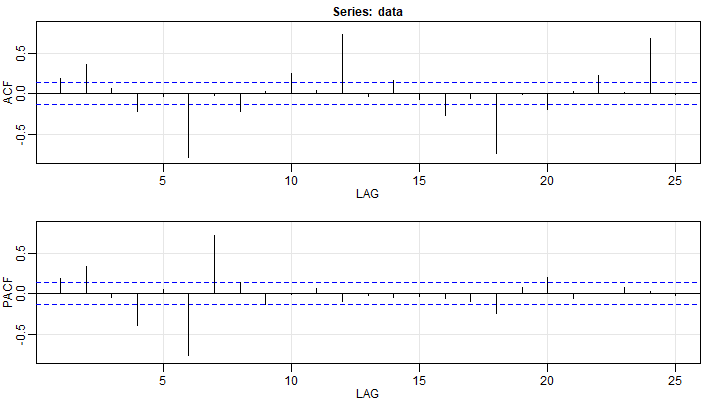
Table 2: 12 Month Seasonal ARIMA Model Selection Criteria



*Figure 15: Predictions for SARIMA(1,1,1,0,1,8)[6]*



*Figure 12: 6 mo. ACF/PACF (Additive Decomposition)*



*Figure 13: 6 mo. ACF/PACF (Differenced Values)*

# R Output Appendix

## Output Corresponding to *Figure 8: Predictions for SARIMA(1,1,1,1,1,1)[12]*

$pred

Time Series:

Start = 217

End = 228

Frequency = 1

[1] 3.481899 3.276038 3.638963 3.541439 3.650048 3.471530 3.471427 3.419460 3.275396 3.384059 3.310083 3.450566

$se

Time Series:

Start = 217

End = 228

Frequency = 1

[1] 0.04768405 0.06310634 0.07326288 0.08099787 0.08739004 0.09295581 0.09797341 0.10260446 0.10694813 0.11106811 0.11500695 0.11879420

## Output Corresponding to *Figure 9: Predictions for SARIMA(1,1,1,2,1,4)[12]*

$pred

Time Series:

Start = 217

End = 228

Frequency = 1

[1] 3.470079 3.233970 3.603222 3.520845 3.647307 3.456141 3.430839 3.373121 3.238863 3.345074 3.279167 3.416496

$se

Time Series:

Start = 217

End = 228

Frequency = 1

[1] 0.04442312 0.06047110 0.07144190 0.07973631 0.08633093 0.09174528 0.09629574 0.10019140 0.10357813 0.10656165 0.10922084 0.11161583

## Output Corresponding to *Figure 14: Predictions for SARIMA(1,1,1,0,1,4)[6]*

$pred

Time Series:

Start = 217

End = 228

Frequency = 1

[1] 3.495548 3.481527 3.684371 3.618268 3.627890 3.516632 3.502040 3.325169 3.394839 3.445085 3.437161 3.478845

$se

Time Series:

Start = 217

End = 228

Frequency = 1

[1] 0.06570373 0.07683521 0.08934421 0.09972872 0.10924578 0.11797366 0.11821807 0.12027388 0.12175039 0.12332191 0.12484853 0.12636219

## Output Corresponding to *Figure 15: Predictions for SARIMA(1,1,1,0,1,8)[6]*

$pred

Time Series:

Start = 217

End = 228

Frequency = 1

[1] 3.511041 3.495870 3.753514 3.645808 3.683702 3.519558 3.476245 3.280853 3.278258 3.397739 3.381312 3.498972

$se

Time Series:

Start = 217

End = 228

Frequency = 1

[1] 0.05689227 0.07064837 0.08254297 0.09290773 0.10222792 0.11076650 0.11121211 0.11274990 0.11419309 0.11562212 0.11703351 0.11842804

# Code Appendix

#####################################################################################

#### Setup / Important Functions

#####################################################################################

## Install and load libraries

ipak <- function(pkg) {

new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]

if (length(new.pkg))

install.packages(new.pkg, dependencies = TRUE)

sapply(pkg, require, character.only = TRUE)

}

packages <-

c("ggplot2", "ggfortify", "reshape2", "gridExtra", "TSA", "astsa", "orcutt", "nlme")

ipak(packages)

saveplot <- function(data, xaxis, yaxis, file)

{

plt = autoplot(data) + xlab(xaxis) + ylab(yaxis)

png(paste("./figures/", file, ".png", sep = ""), width = 712, height = 400)

grid.arrange(plt)

dev.off()

}

saveplot\_xy <- function(xdata, ydata, xaxis, yaxis, file)

{

plt = qplot(x=xdata, y=ydata) + geom\_line() + xlab(xaxis) + ylab(yaxis)

png(paste("./figures/", file, ".png", sep = ""), width = 712, height = 400)

grid.arrange(plt)

dev.off()

}

saveacf <- function(data, file)

{

png(paste("./figures/", file, ".png", sep = ""), width = 712, height = 400)

acf2(data)

dev.off()

}

#####################################################################################

#### Define Important Variables

#####################################################################################

## Load Data

sales\_data = read.csv(file = "SalesData.csv", header = TRUE, sep = ",")

## Set seasonality value

seasonal\_trend\_val = 12

fitModels = FALSE

forecastModels = TRUE

#####################################################################################

#### Initial Data Analysis

#####################################################################################

# Plot Time Series

sales = ts(sales\_data$Sales)

saveplot(sales, 'time', 'sales', 'time\_series\_sales') # Linear trend and seasonality. Detrend with linear transformation.

saveacf(sales, "acf\_sales") # 1st or 2nd order AR appearance with no transformation. 12 time step spike -> 12 month seasonality

# Examine periodogram

periodogram(sales)

base\_power = periodogram(sales)$spec

base\_frequencies = periodogram(sales)$freq

base\_periods = 1 / base\_frequencies

base\_periods\_yearly = base\_periods / 12

saveplot\_xy(

base\_periods\_yearly, base\_power,

"Base Periods in Years",

"Base Power",

"periodic\_base")

#####################################################################################

#### Calculate Trends

#####################################################################################

# Series Fractional Smoothing

sales\_trend = filter(sales, filter = c(1 / (seasonal\_trend\_val \* 2),

rep(1 / seasonal\_trend\_val, (seasonal\_trend\_val - 1)),

1 / (seasonal\_trend\_val \* 2)), sides = 2)

# Series Exponential Smoothing

sales\_arima = arima(sales, order = c(0, 1, 1))

sales\_exp = sales - sales\_arima$residuals

# Series Decompositions

sales\_freq = ts(sales\_data$Sales, freq = seasonal\_trend\_val)

# Additive Decomposition Trend

sales\_decomp\_add = decompose(sales\_freq, type = "additive")

# Multiplicative Decomposition Trend

sales\_decomp\_mul = decompose(sales\_freq, type = "multiplicative")

# Set Plot Titles

fractional\_title = paste("Fractional Smoothing Trend For", seasonal\_trend\_val, "Months")

exponential\_title = paste("Exponential Smoothing Trend For", seasonal\_trend\_val, "Months")

additive\_title = paste("Additive Decomposition Trend For", seasonal\_trend\_val, "Months")

multiplicative\_title = paste("Multiplicative Decomposition Trend For", seasonal\_trend\_val, "Months")

# Save Trends Plot

trend\_1 <- autoplot(sales\_trend) + xlab("Time") + ylab("Fractional Smoothing Trend Values") + ggtitle(fractional\_title)

trend\_2 <- autoplot(sales\_exp) + xlab("Time") + ylab("Exponential Smoothing Trend Values") + ggtitle(exponential\_title)

trend\_3 <- autoplot(sales\_decomp\_add$trend) + xlab("Time") + ylab("Additive Decomposition Trend Values") + ggtitle(additive\_title)

trend\_4 <- autoplot(sales\_decomp\_mul$trend) + xlab("Time") + ylab("Multiplicative Decomposition Trend Values") + ggtitle(multiplicative\_title)

png(paste("./figures/trends\_for", seasonal\_trend\_val, "month", "season.png", sep="\_"), width = 712, height = 400)

grid.arrange(trend\_1, trend\_2, trend\_3, trend\_4, ncol = 2)

dev.off()

#####################################################################################

#### Basic Differencing

#####################################################################################

# Seasonal Difference

sales\_seasonally = diff(sales, seasonal\_trend\_val)

# 1st Difference

sales\_diff = diff(sales\_seasonally, 1)

# Set Plot Titles

seasonal\_title = paste(seasonal\_trend\_val, "Month Differenced Time Series")

seasonal\_and\_1st\_title = paste(seasonal\_trend\_val, "Month and 1st Differenced Time Series")

# Differencing Plots

diff\_1 <- autoplot(sales\_seasonally) + xlab("Time") + ylab("Sales") + ggtitle(seasonal\_title)

diff\_2 <- autoplot(sales\_diff) + xlab("Time") + ylab("Sales") + ggtitle(seasonal\_and\_1st\_title)

png(paste("./figures/differencing\_for", seasonal\_trend\_val, "month", "season.png", sep="\_"), width = 712, height = 400)

grid.arrange(diff\_1, diff\_2, ncol = 1)

dev.off()

#####################################################################################

#### Detrend Time Series

#####################################################################################

# Fractional Smoothing Detrend

sales\_frac\_smooth = sales - sales\_trend

# Exponentials Smoothing Detrend

sales\_exp\_smooth = sales - sales\_exp

# Additive Decomposition Detrend

sales\_add\_detrend = sales\_freq - sales\_decomp\_add$trend

# Multiplicative Decomposition Detrend

sales\_mul\_detrend = sales\_freq - sales\_decomp\_mul$trend

#####################################################################################

#### Detrended Series Analysis

#####################################################################################

# Examine differenced periodogram

periodogram(sales\_seasonally)

diff\_power = periodogram(sales\_seasonally)$spec

diff\_frequencies = periodogram(sales\_seasonally)$freq

diff\_periods = 1 / diff\_frequencies

diff\_periods\_yearly = diff\_periods / 12

saveplot\_xy(diff\_periods\_yearly, diff\_power, paste(seasonal\_trend\_val, " Month Differenced Periods in Years", sep = ""), "Power",

paste("periodic", seasonal\_trend\_val, "month\_detrend", sep ="\_"))

# Detrended Series Plots:

dt\_1 <- autoplot(sales\_frac\_smooth) + ggtitle("Fractional Smoothing") + xlab("Time") + ylab("Detrended Value")

dt\_2 <- autoplot(sales\_exp\_smooth) + ggtitle("Exponential Smoothing") + xlab("Time") + ylab("Detrended Value")

dt\_3 <- autoplot(sales\_add\_detrend) + ggtitle("Additive Detrend") + xlab("Time") + ylab("Detrended Value")

dt\_4 <- autoplot(sales\_mul\_detrend) + ggtitle("Multiplicative Detrend") + xlab("Time") + ylab("Detrended Value")

png(paste("./figures/smoothed\_plots", seasonal\_trend\_val, "month", "season.png", sep="\_"), width = 712, height = 400)

grid.arrange(dt\_1, dt\_2, dt\_3, dt\_4, ncol = 2)

dev.off()

# ACF & PACF of Additive and Fractionally Smoothed Data

saveacf(sales\_add\_detrend, paste("acf", "sales\_additive\_detrend", seasonal\_trend\_val, "months", sep = "\_"))

saveacf(sales\_frac\_smooth, paste("acf", "sales\_fractional\_detrend", seasonal\_trend\_val, "months", sep = "\_"))

saveacf(sales\_seasonally, paste("acf", "sales\_seasonal\_detrend", seasonal\_trend\_val, "months", sep = "\_"))

#####################################################################################

#### Fit Time Series Models

#####################################################################################

if(fitModels == TRUE)

{

if (seasonal\_trend\_val == 12) {

# 12th and 1st difference

sarima(sales\_seasonally, 1, 1, 0, 1, 1, 0, 12) # -4.119955, -4.109563, -5.087425

sarima(sales\_seasonally, 1, 1, 1, 1, 1, 1, 12) # -4.805208, -4.793919, -5.740147

sarima(sales, 1, 1, 0, 1, 1, 0, 12) # -4.862065, -4.852282, -5.830813

sarima(sales, 1, 1, 1, 1, 1, 1, 12) # -5.049281, -5.038699, -5.986776

# 12 month additive decomposition

sarima(sales\_add\_detrend, 1, 1, 1, 2, 1, 4, 1) # -3.674233, -3.660929, -4.549223

sarima(sales, 1, 1, 1, 2, 1, 4, 12) # -5.154983, -5.141679, -6.029973

}

if (seasonal\_trend\_val == 6){

# 6th and 1st difference

sarima(sales\_seasonally, 1, 1, 1, 0, 1, 4, 6) # -4.155934, -4.14377, -5.060303

sarima(sales\_seasonally, 1, 1, 1, 0, 1, 8, 6) # -4.593377, -4.577504, -5.433991

sarima(sales, 1, 1, 1, 0, 1, 4, 6) # -4.389765, -4.378013, -5.296007

sarima(sales, 1, 1, 1, 0, 1, 8, 6) # -4.642834, -4.627584, -5.486572

}

}

#####################################################################################

#### Forecast Best Models

#####################################################################################

if(forecastModels == TRUE)

{

# 12 Month Trend Forecasting

if (seasonal\_trend\_val == 12)

{

png("./figures/sarima\_12\_month\_predict\_1.png", width = 712, height = 400)

sarima.for(sales, 12, 1, 1, 1, 1, 1, 1, 12)

dev.off()

png("./figures/sarima\_12\_month\_predict\_2.png", width = 712, height = 400)

sarima.for(sales, 12, 1, 1, 1, 2, 1, 4, 12)

dev.off()

}

# 6 Month Trend Forecasting

if (seasonal\_trend\_val == 6) {

png("./figures/sarima\_6\_month\_predict\_1.png", width = 712, height = 400)

sarima.for(sales, 12, 1, 1, 1, 0, 1, 4, 6)

dev.off()

png("./figures/sarima\_6\_month\_predict\_2.png", width = 712, height = 400)

sarima.for(sales, 12, 1, 1, 1, 0, 1, 8, 6)

dev.off()

}

}