Project 6: Time Series Forecasting

Import Data

Import

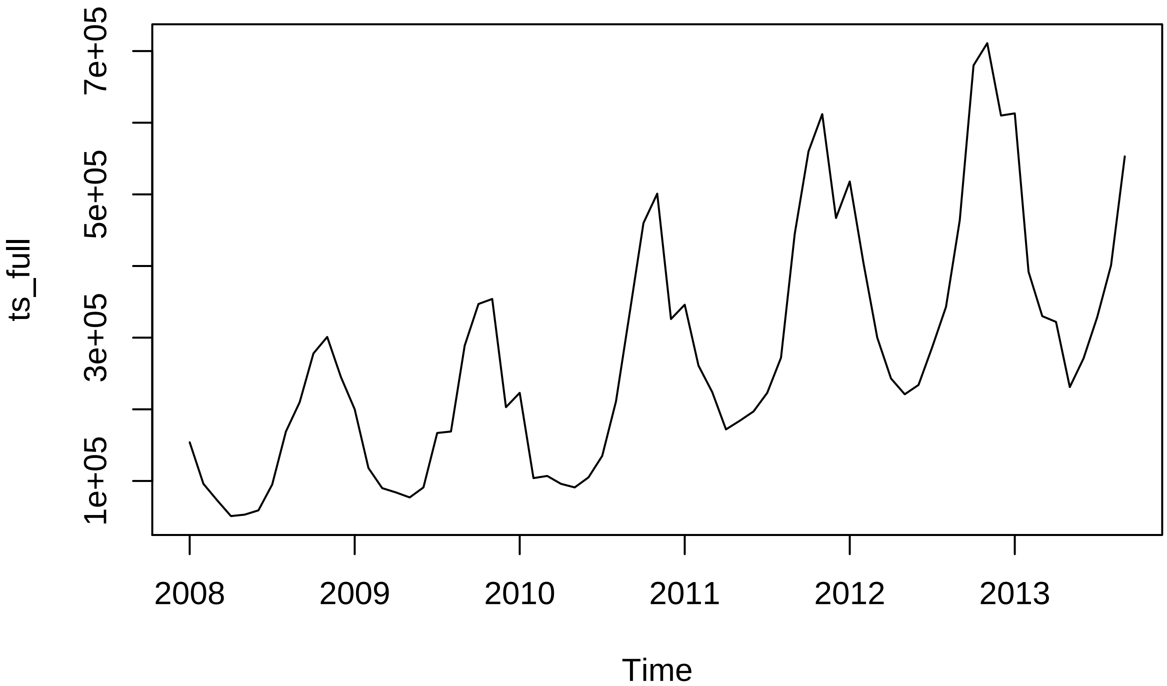
```
# Import data
ms <- read.csv('data/monthly-sales-clean.csv')
```

Convert Data to Time Series

```
# Load dependencies
library(PerformanceAnalytics)
```

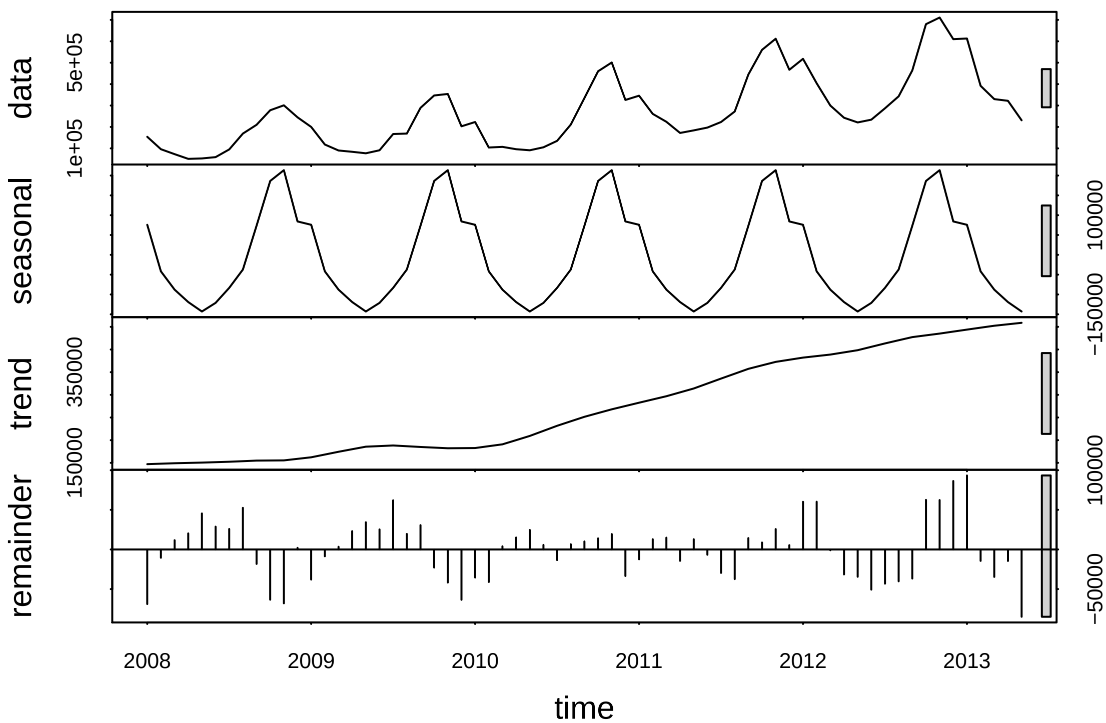
```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

```
# Convert bookings_df to time series object
ts_train <- ts(ms$monthly_sales, start=c(2008, 1), end=c(2013, 5), frequency=12)
ts_full <- ts(ms$monthly_sales, start=c(2008, 1), end=c(2013, 9), frequency=12)
plot(ts_full)
```



Determine ETS Components

```
# Fit time series decomposition
fit <- stl(ts_train, s.window='period')
# Plot
plot(fit)
```



As the above

time series decomposition plot shows, the time series displays:

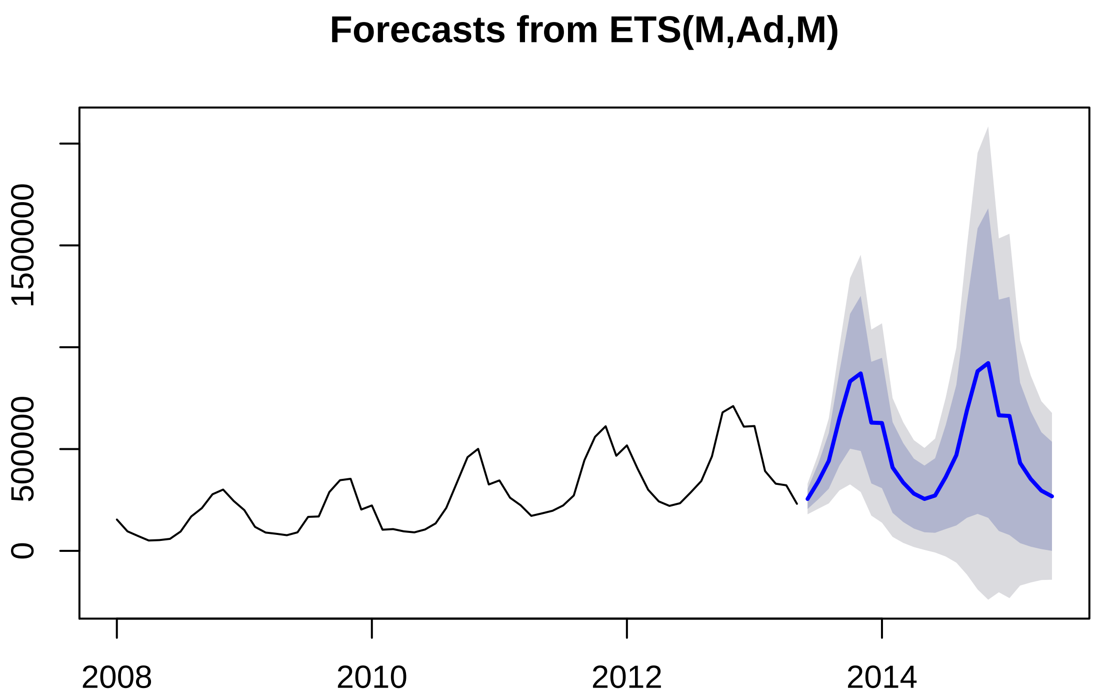
- Error: Multiplicative
- Trend: Additive
- Seasonality: Multiplicative

ETS Model

Build Model

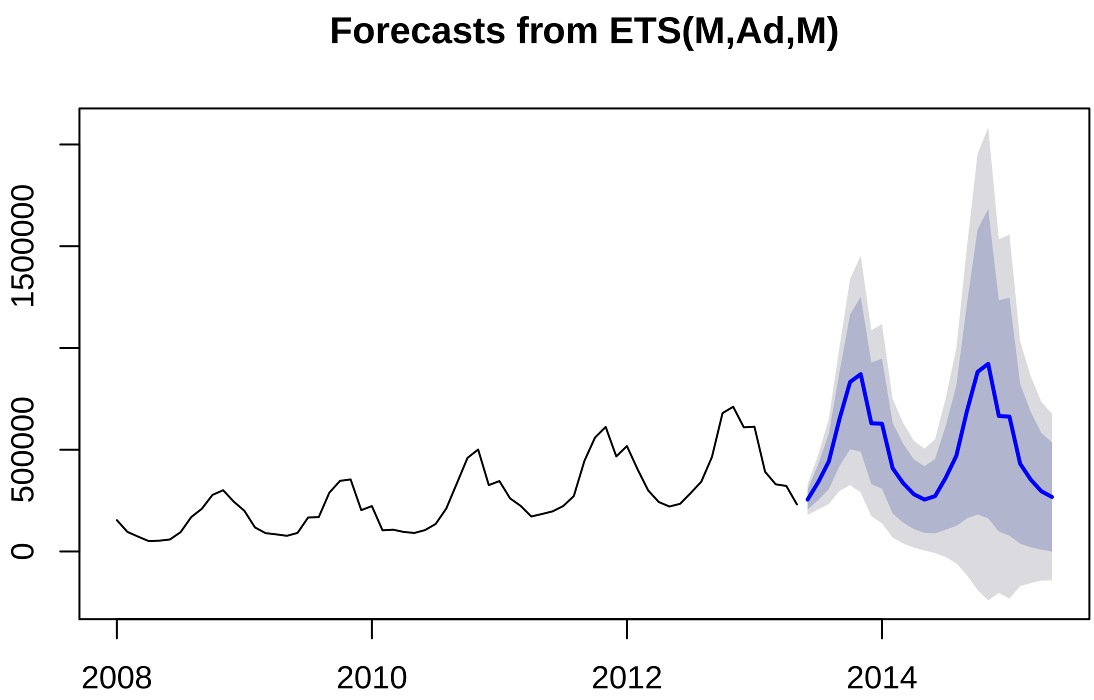
Manual ETS Model

```
# Load dependencies
library(forecast)

# Holt-Winters Seasonal Model
fit_ets_manual <- ets(ts_train, model='MAM')
plot(forecast(fit_ets_manual))
```



Automated ETS Model

```
# ETS Model with train dataset
fit_ets_auto <- ets(ts_train)
plot(forecast(fit_ets_auto))
```



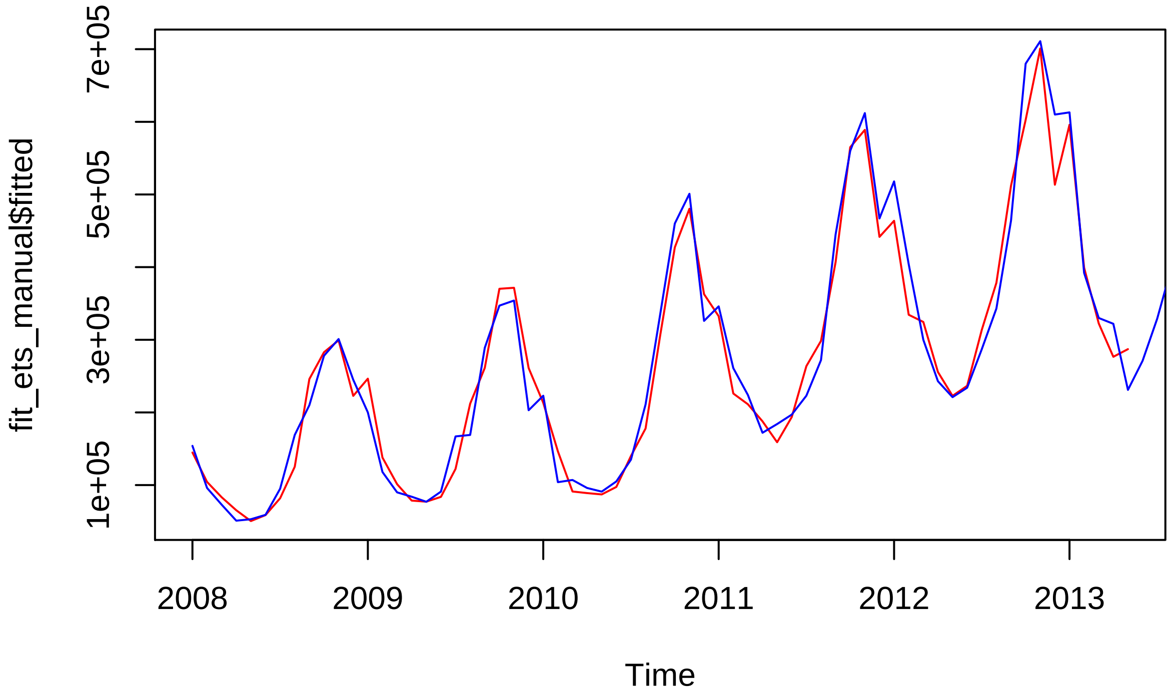
Forecast/Test Accuracy

Manual ETS Model

```
# fit_ets is the model prediction
# ts is the actual time series object
accuracy(forecast(fit_ets_manual, 4), ts_full[66:69])
```

```
##                       ME     RMSE      MAE       MPE      MAPE      MASE
## Training set   3243.47 31474.37 24188.22 -0.572395 10.305204 0.4019854
## Test set     -33469.61 53828.48 41542.76 -6.347585  9.326605 0.6904015
##                     ACF1
## Training set 0.008740233
## Test set              NA
```

```
# Plot accuracy
# Fitted in Red
plot(fit_ets_manual$fitted, col='red')
# Actual in Blue
lines(ts_full, col='blue')
```

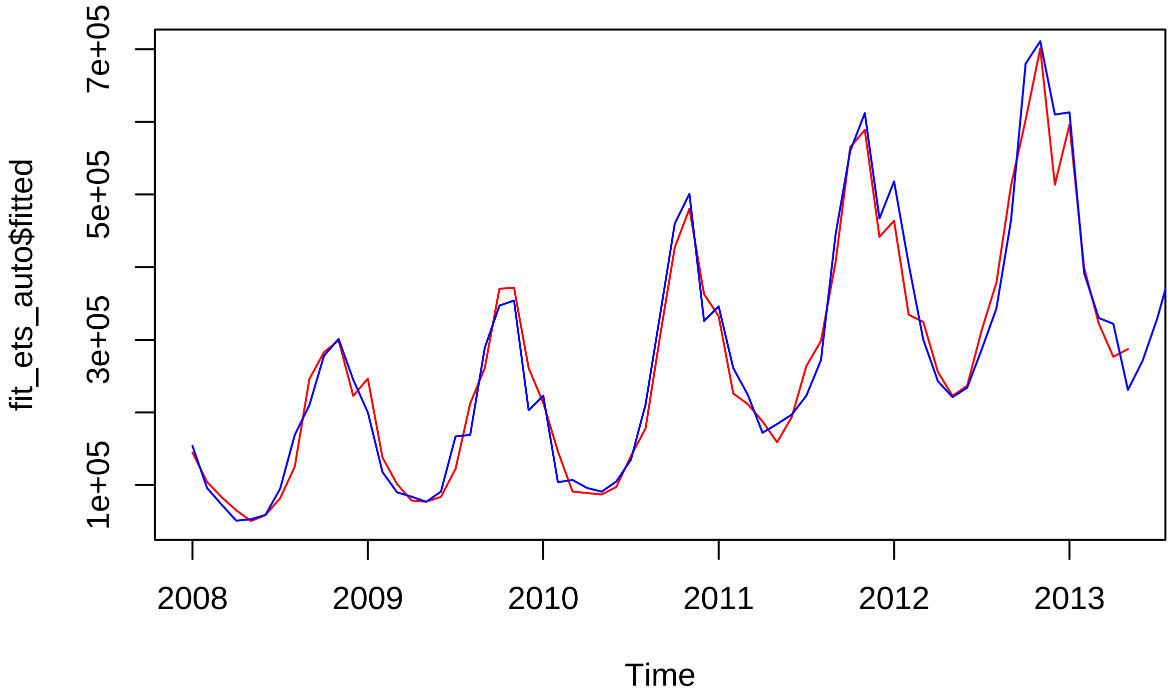


Automated ETS Model

```
# fit_ets is the model prediction
# ts is the actual time series object
accuracy(forecast(fit_ets_auto, 4), ts_full[66:69])
```

```
##                      ME     RMSE      MAE       MPE      MAPE      MASE
## Training set   3243.47 31474.37 24188.22 -0.572395 10.305204 0.4019854
## Test set     -33469.61 53828.48 41542.76 -6.347585  9.326605 0.6904015
##                     ACF1
## Training set 0.008740233
## Test set              NA
```

```
# Plot accuracy
# Fitted in Red
plot(fit_ets_auto$fitted, col='red')
# Actual in Blue
lines(ts_full, col='blue')
```



Both the manual and automated ets model, yield an MAPE of 9.326605%.

ARIMA Model

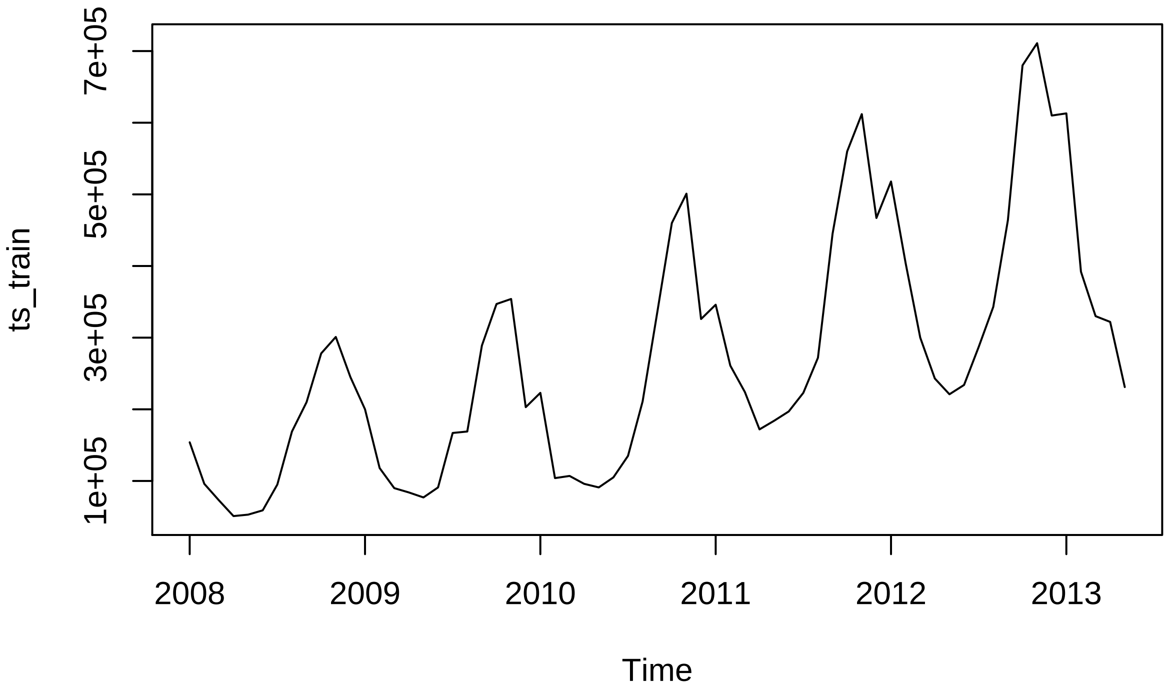
Build Model

Manual ARIMA Model

Stationarize Dataset

Plot the data to check if stationary

```
# Plot data to check if constant mean/variance
plot(ts_train)
```

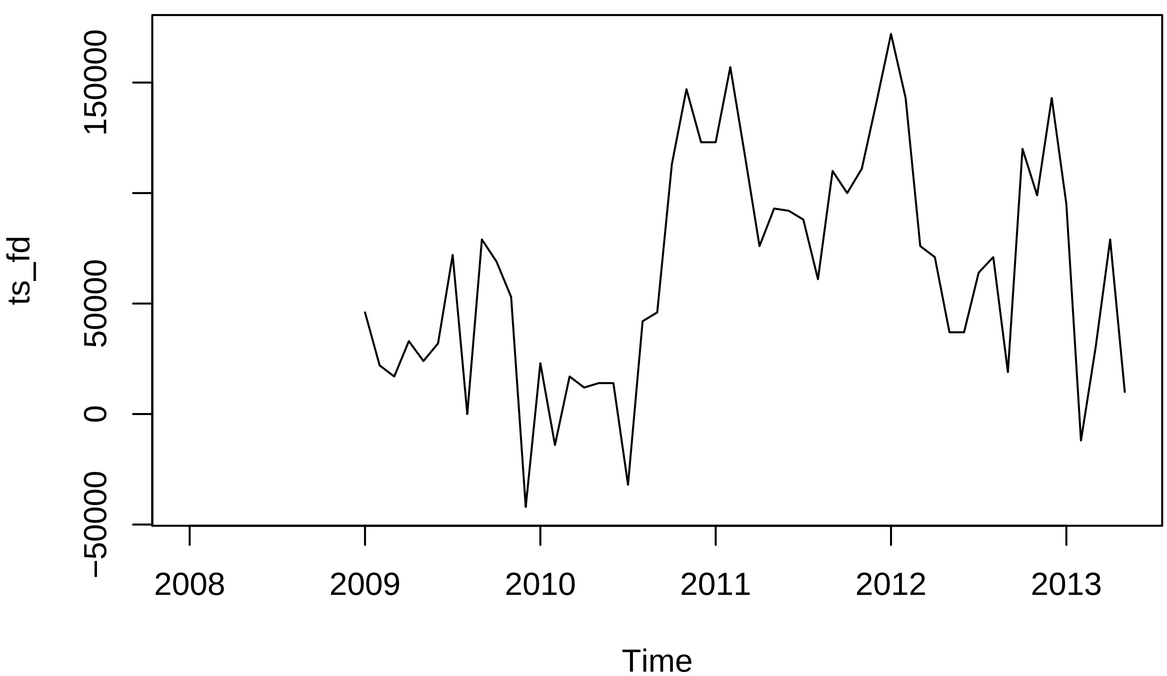


The data is not stationary and is seasonal, so let's seasonally stationarize the data.

```
# First Seasonal Difference
ms$first_difference <- c(rep(NA,12), diff(ms$monthly_sales, lag=12))
```

Plot the data again, to check if stationary

```
# Make first_difference time series
ts_fd <- ts(ms$first_difference, start=c(2008, 1), end=c(2013, 5), frequency=12)

# Plot first_difference
plot(ts_fd)
```



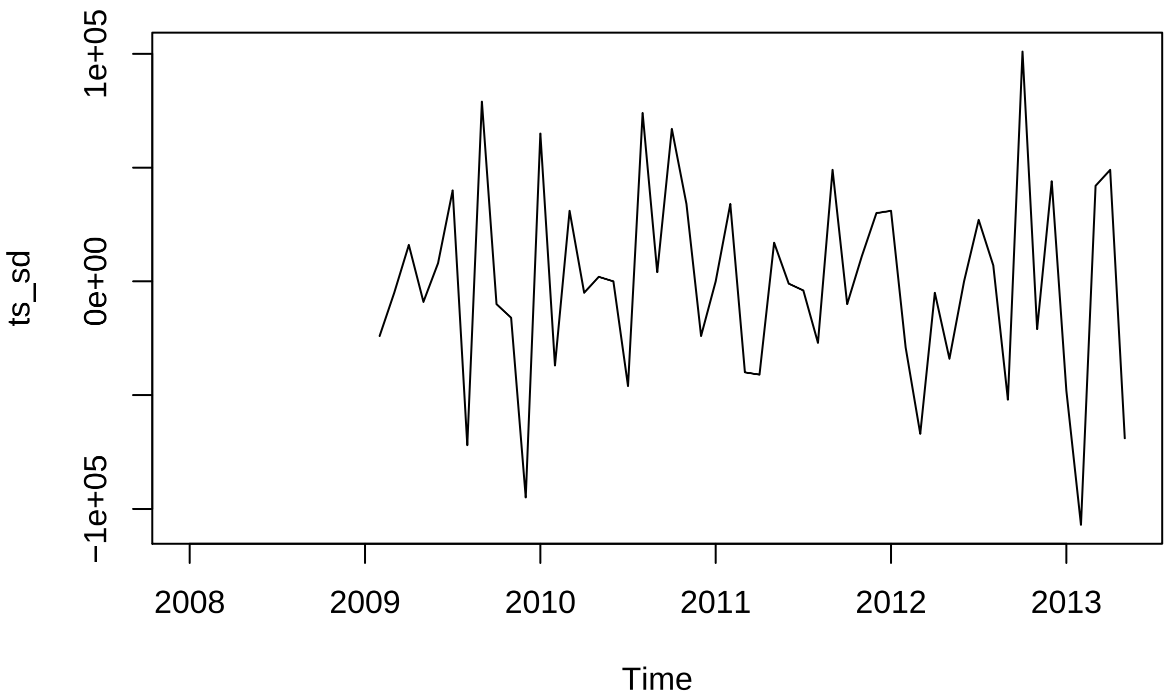
The

first_difference does not appear seasonal, but is still not stationary. Let's take a second, non-seasonal difference.

```
# Second, non-seasonal difference
ms$second_difference <- c(NA, diff(ms$first_difference, lag=1))
```

Plot the data again, to check if stationary

```
# Make first_difference time series
ts_sd <- ts(ms$second_difference, start=c(2008, 1), end=c(2013, 5), frequency=12)

# Plot first_difference
plot(ts_sd)
```



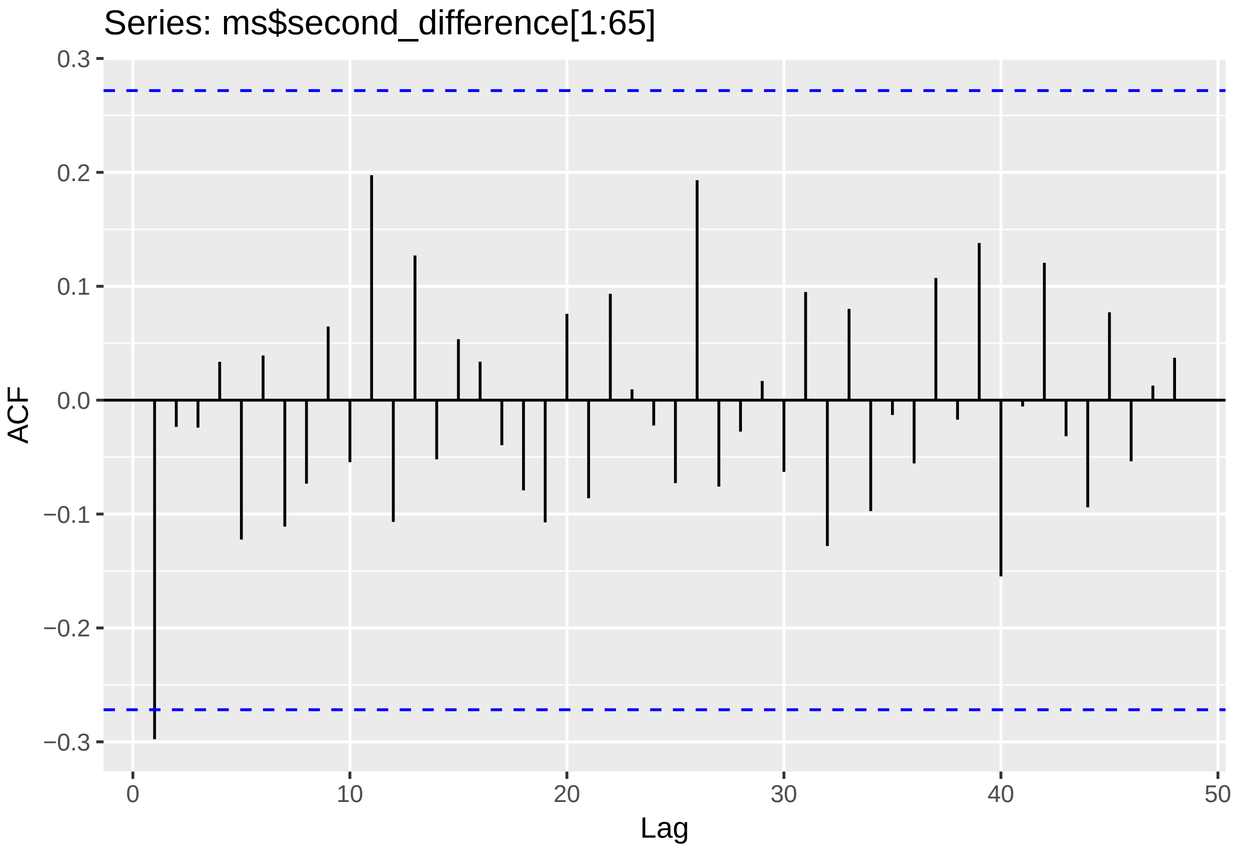
Now, the time series displays a constant mean and variance, without any seasonality.

The model structure thus far, after taking a seasonal (D=1) difference and non-seasonal difference (d=1) to stationarize the data, with a period of 12 is: - ARIMA(0,1,0)(0,1,0)[12]

AR and MA Terms

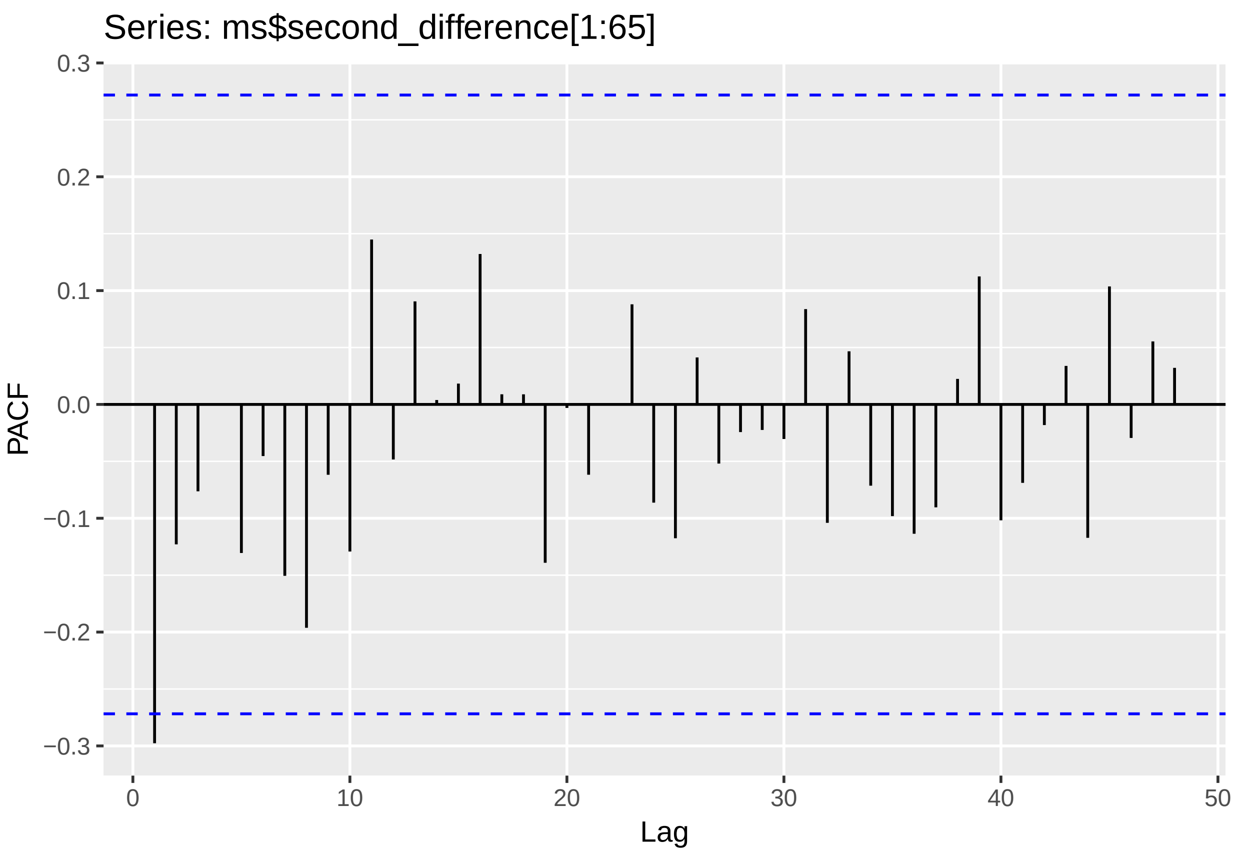
ACF Plot

```
# Plot the ACF of second_difference
ggAcf(ms$second_difference[1:65], lag.max=48)
```



PACF Plot

```
# Plot the PACF of second_difference
ggPacf(ms$second_difference[1:65], lag.max=48)
```



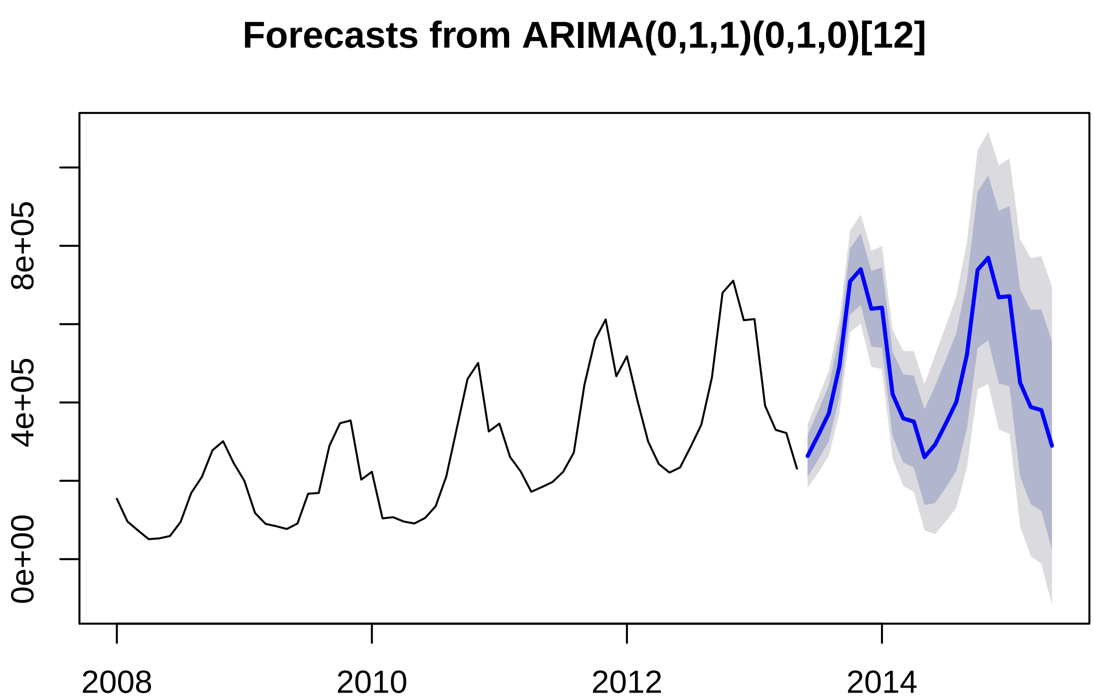
-

The ACF and PACF have negative values at lag 1, suggesting a non-seasonal MA Term, signified as q=1. - The ACF and PACF show little AC and PAC at the first seasonal lag, lag 12, suggesting Q=0.

Thus, the model structure after taking a non-seasonal MA Term (q=1) is: - ARIMA(0,1,1)(0,1,0)[12]

Build Model

```
# ARIMA Model
fit_arima_manual <- Arima(ts_train, order=c(0,1,1), seasonal=c(0,1,0))
plot(forecast(fit_arima_manual))
```



Automated ARIMA Model

```
# Build Auto Arima Model
fit_arima_auto <- auto.arima(ts_train)
fit_arima_auto
```

```
## Series: ts_train
## ARIMA(0,1,1)(0,1,0)[12]
##
## Coefficients:
##           ma1
##       -0.3780
## s.e.   0.1462
##
## sigma^2 estimated as 1.722e+09:  log likelihood=-626.3
## AIC=1256.6   AICc=1256.84   BIC=1260.5
```

Forecast/Test Accuracy

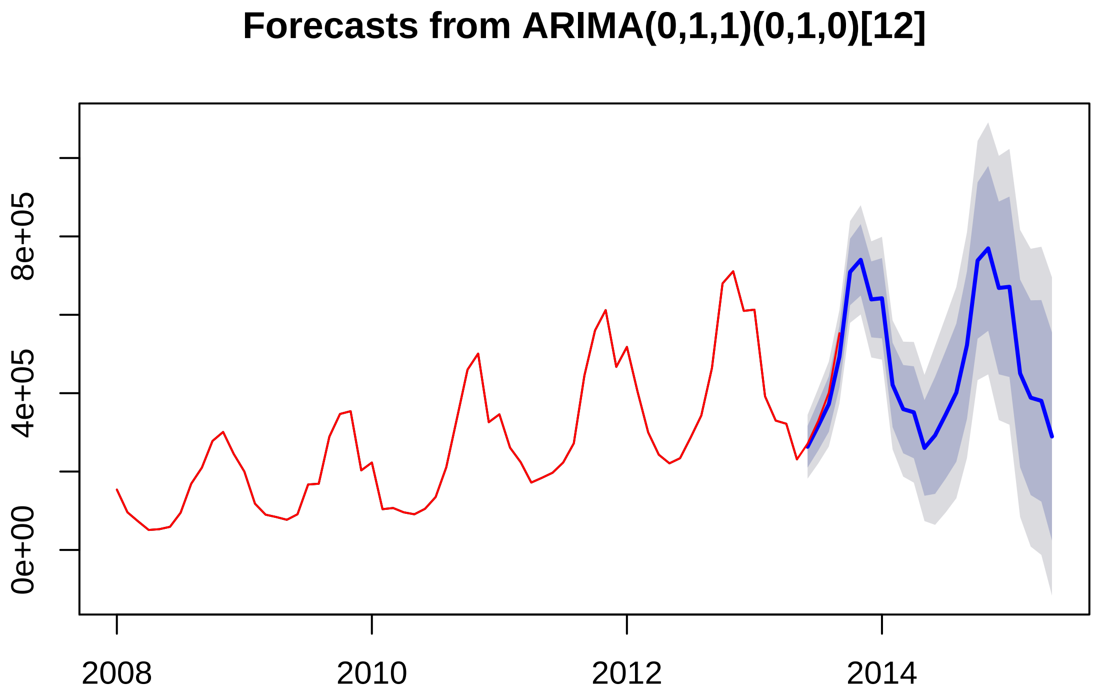
Manual ARIMA Model

```
# fit_arima_auto is the model prediction
# ts_full is the actual time series object
accuracy(forecast(fit_arima_manual, 4), ts_full[66:69])
```

```
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set  -356.2665 36761.53 24993.04 -1.802137 9.824411 0.4153609
## Test set     27271.5199 33999.79 27271.52  6.183294 6.183294 0.4532270
##                    ACF1
## Training set 0.01641446
```

```
## Test set                  NA
```

```r
# Plot accuracy
# Fitted in Blue
plot(forecast(fit_arima_manual))
# Actual in Red
lines(ts_full, col='red')
```

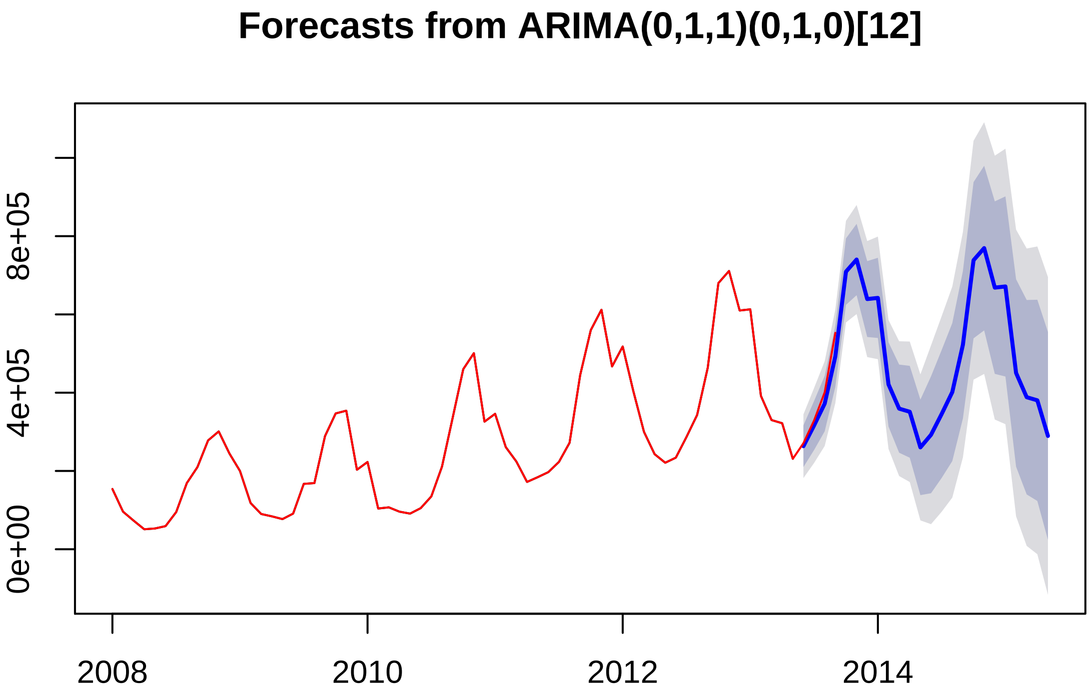


Automated ARIMA Model

```r
# fit_arima_auto is the model prediction
# ts_full is the actual time series object
accuracy(forecast(fit_arima_auto, 4), ts_full[66:69])
```

```
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set  -356.2665 36761.53 24993.04 -1.802137 9.824411 0.4153609
## Test set     27271.5199 33999.79 27271.52  6.183294 6.183294 0.4532270
##                    ACF1
## Training set 0.01641446
## Test set             NA
```

```r
# Plot accuracy
# Fitted in Blue
plot(forecast(fit_arima_auto))
# Actual in Red
lines(ts_full, col='red')
```



The ARIMA Model (both manual and automated) yields an MAPE of 93.8%.

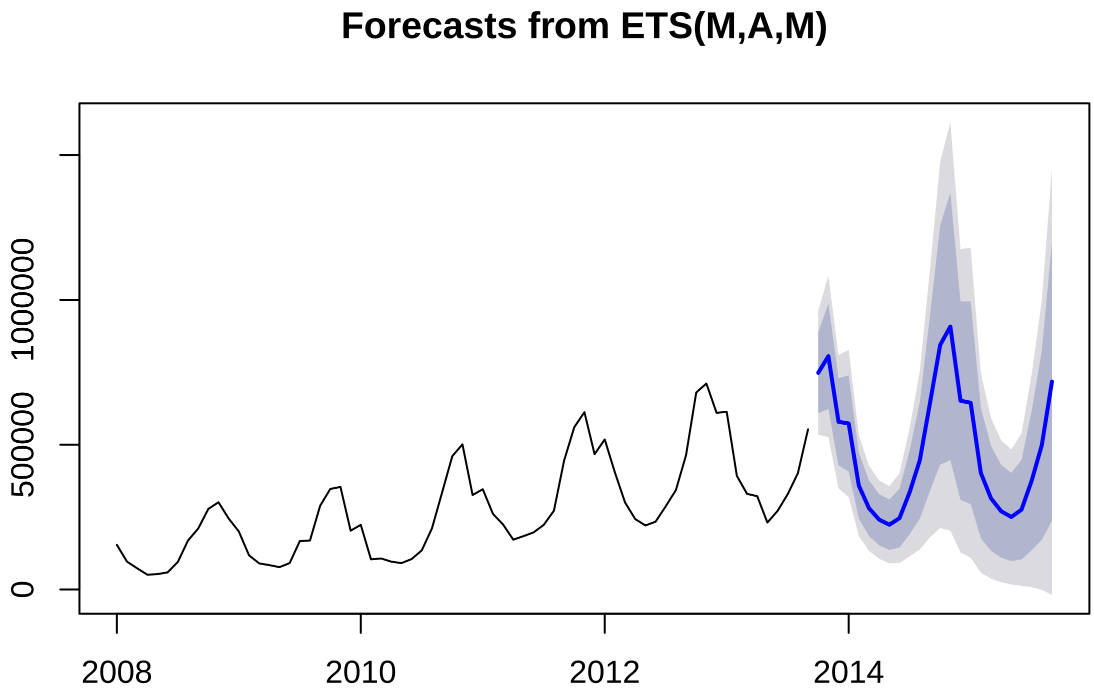
Choose Best Model

The Holt-Winters Seasonal model yields the highest accuracy.

Build ARIMA Model

Build the Holt-Winters Seasonal Model with all data.

```
# Holt-Winters Seasonal Model with all data
fit_bm <- ets(ts_full, model='MAM')
plot(forecast(fit_bm))
```



Predict the next 4 periods.

```
# Holt-Winters Seasonal Model with all data
forecast(fit_bm, 4)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
## Oct 2013       747868.1 608137.7 887598.4 534168.9  961567.2
## Nov 2013       805244.4 623012.4 987476.3 526544.7 1083944.1
## Dec 2013       578735.9 428216.8 729255.0 348536.8  808935.0
## Jan 2014       573014.4 406742.6 739286.3 318723.6  827305.3
```

The forecasts for the next 4 periods are: - Oct 2013: $747,868 - Nov 2013: $805,244 - Dec 2013: $578,736 - Jan 2014: $573,014 It is odd that December and January forecasts are lower than October and November forecasts, but upon checking the data, this is consistent with previous patterns.

Conclusions

Step 1: Plan Your Analysis

Time Series Criteria

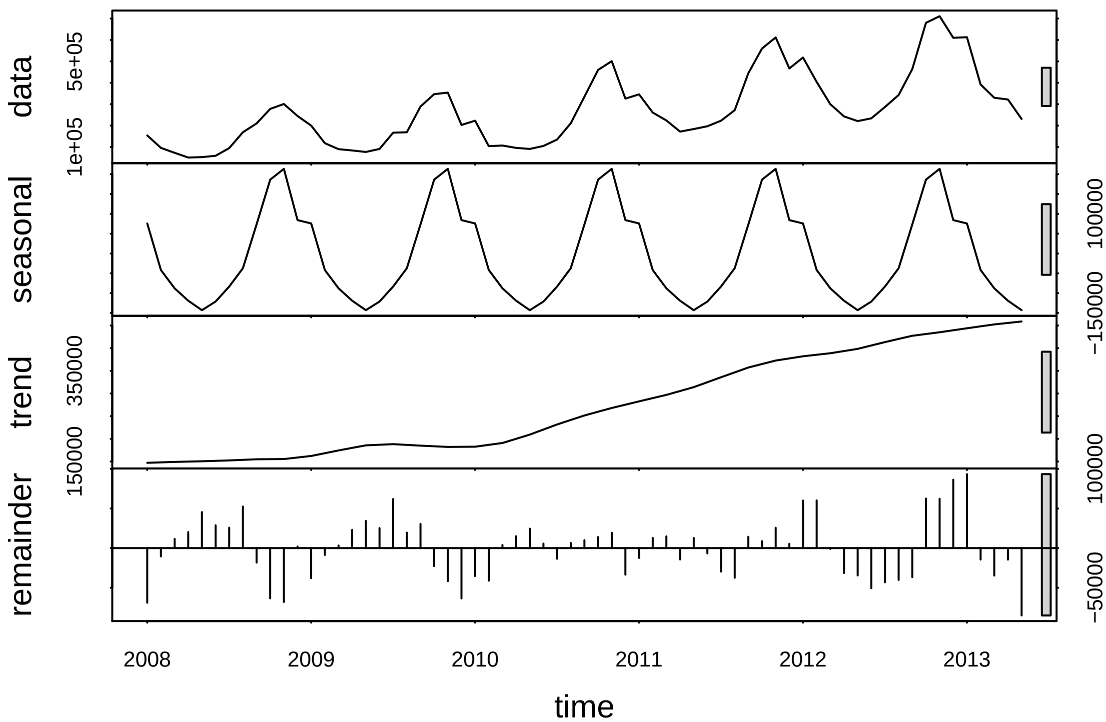
The dataset meets the time series dataset criteria because: - the data is continuous, with monthly sales values from January 2008 to September 2013 - the values are ordered - the values are equally spaced a month apart - there is only one value per each month ### Holdout Sample Because we are attempting to predict four months in the future, we should use monthly sales for the most recent four months as the holdout sample.

Step 2: Determine Trend, Seasonal, and Error Components

Per the time series decomposition graph below:

- Error: Multiplicative
- Trend: Additive

- Seasonality: Multiplicative



Step 3: Build Your Models

ETS Model

Model Terms

Per the time series decomposition graph, the model terms for ETS are additive error, additive trend, and additive seasonality: - ETS(M,A,M) #### In-Sample Error Per the table of errors below: - Root Mean Square Error (RMSE) is 31,474.37 - Mean Absolute Scaled Error (MASE) is 0.3528697 - Mean Absolute Percentage Error (MAPE) is 10.3052

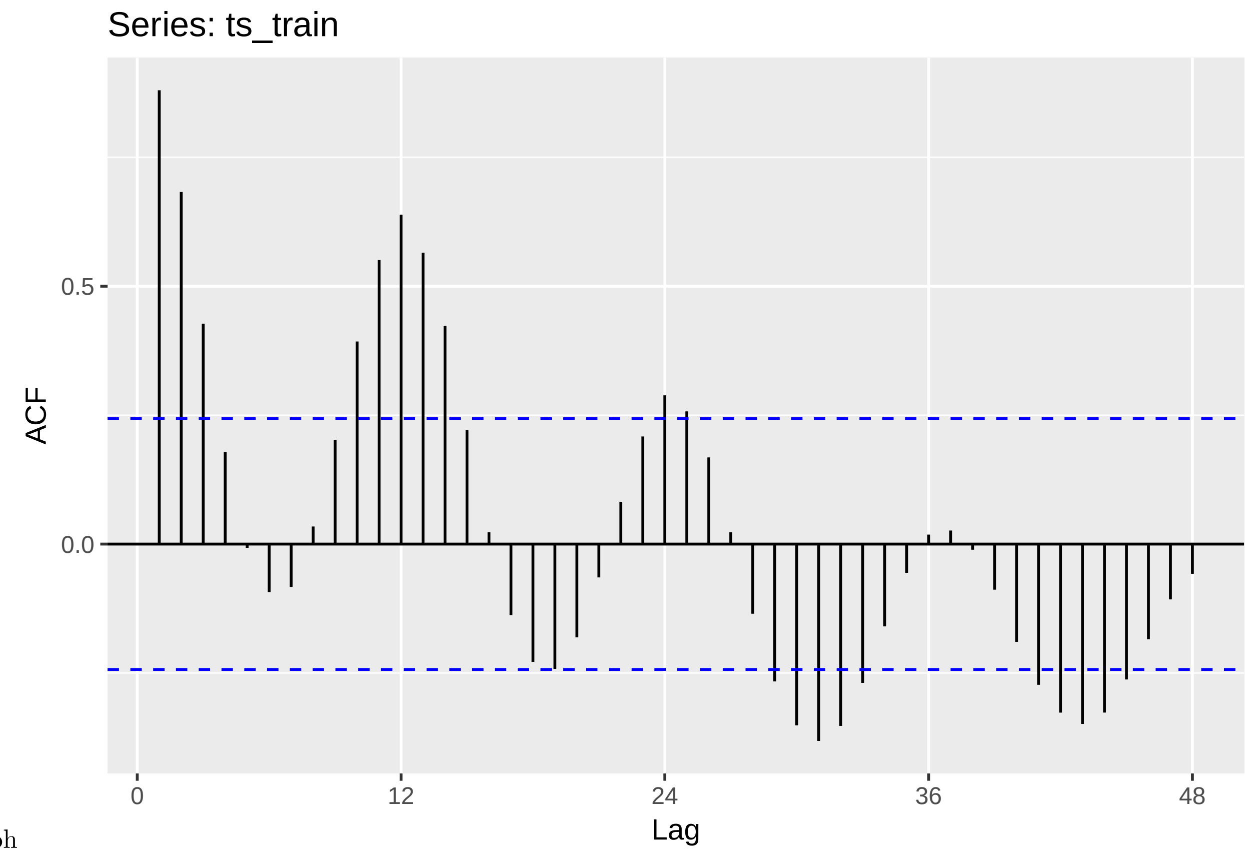
```
##                    ME     RMSE      MAE       MPE    MAPE      MASE
## Training set 3243.47 31474.37 24188.22 -0.572395 10.3052 0.3528697
##                     ACF1
## Training set 0.008740233
```

ARIMA Model

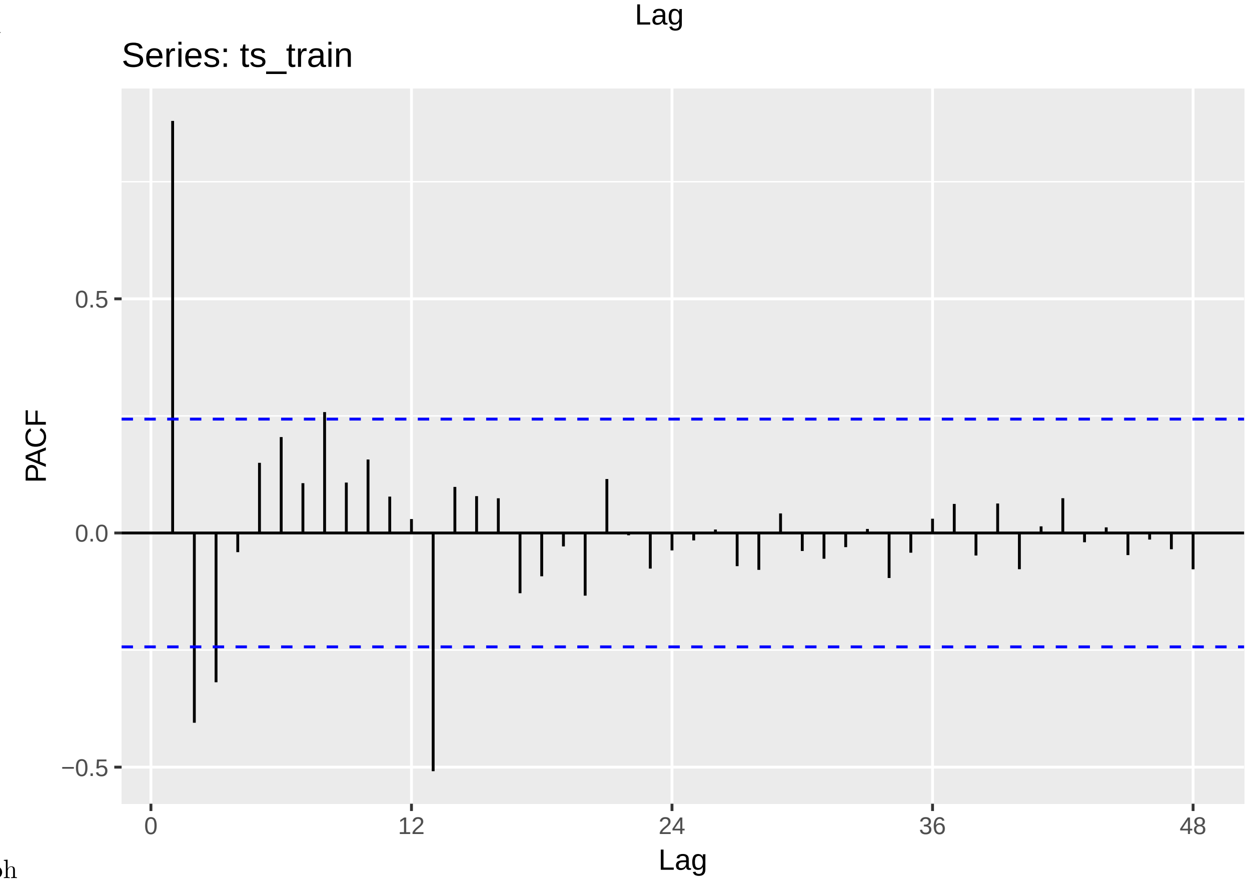
Model Terms

Differencing the Dataset

Per the ACF and PACF graphs of time series data below, the large auto-correlations and partial auto-correlations suggest the data must be differenced both seasonally and non-seasonally, signified as d=1 for non-seasonal differencing and D=1 for seasonal differencing.



ACF Graph

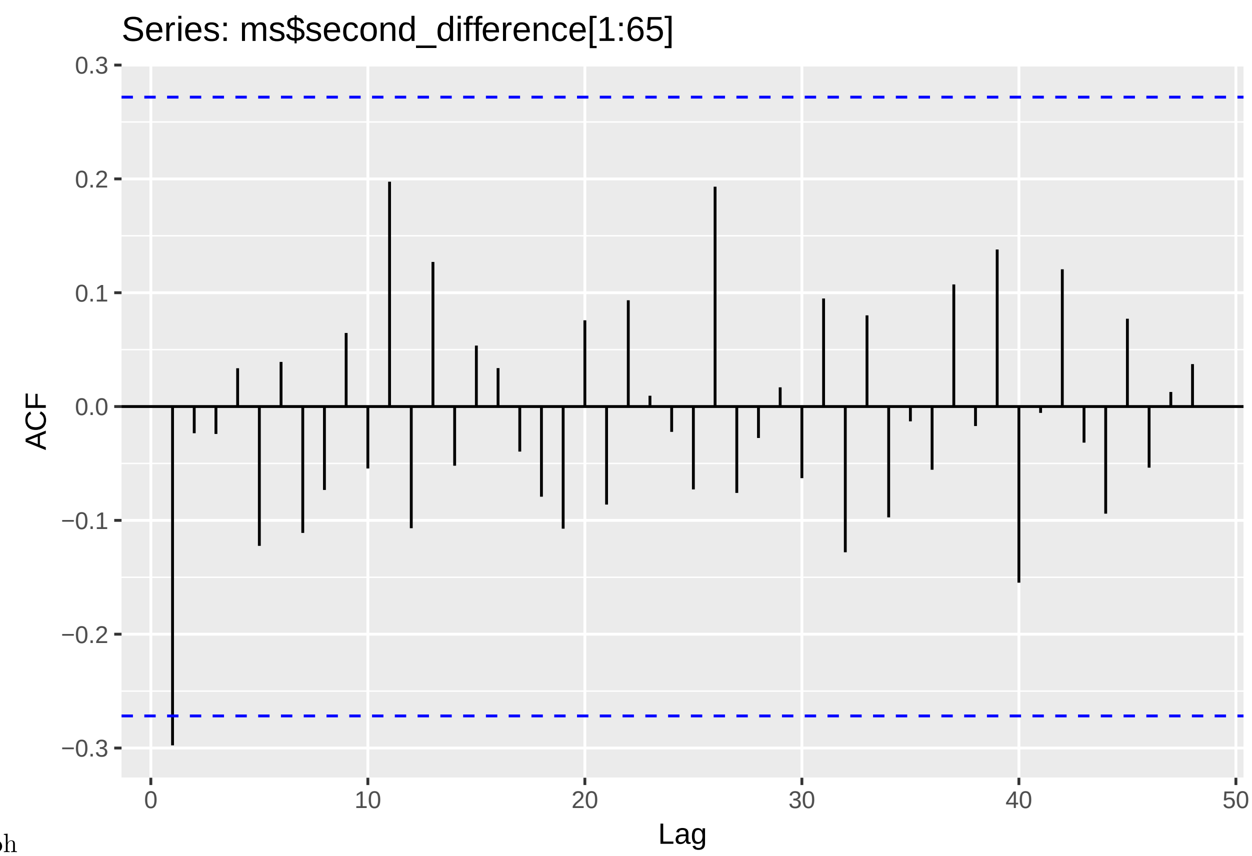


PACF Graph

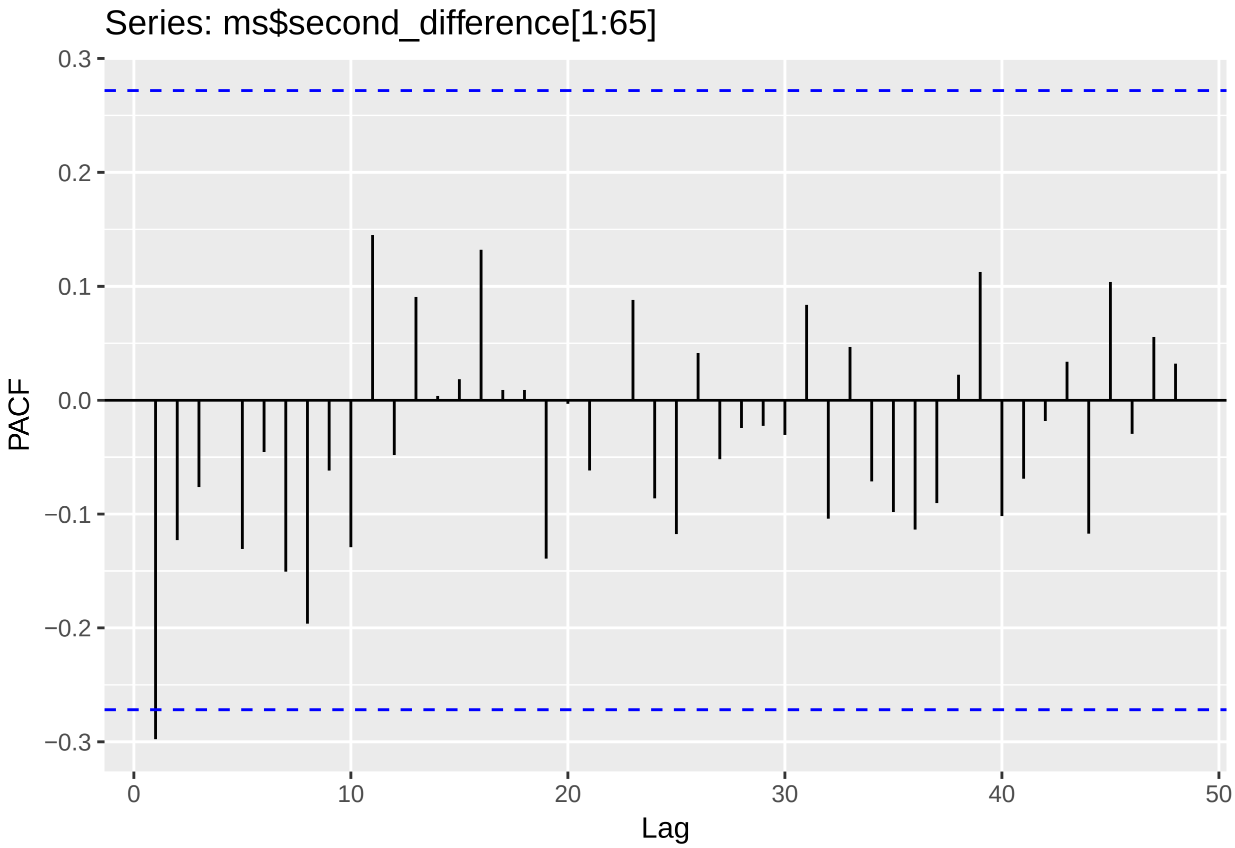
AR and MA Terms

Per the ACF and PACF graphs of stationary data below:

- The ACF and PACF have negative values at lag 1, suggesting a non-seasonal MA Term, signified as q=1.
- The ACF and PACF show little AC and PAC at the first seasonal lag, lag 12, suggesting no seasonal AR or MA Terms, signified as Q=0.



ACF Graph



PACF Graph

Thus, the model structure is:

- ARIMA(0,1,1)(0,1,0)[12]

In-Sample Error

Per the table of errors below:

- Root Mean Square Error (RMSE) is 36,761.53
- Mean Absolute Scaled Error (MASE) is 0.3646109
- Mean Absolute Percentage Error (MAPE) is 9.824411

```
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -356.2665 36761.53 24993.04 -1.802137 9.824411 0.3646109
##                    ACF1
## Training set 0.01641446
```

Step 4:

Choose Best Model

Per the table's of ETS and ARIMA Holdout Sample Error below, the ARIMA Model displays a lower MAPE and MASE. As such, the ARIMA Model was used to forecast the next four months of video game sales.

ETS Holdout Sample Error

```
##                     ME     RMSE      MAE       MPE      MAPE      MASE
## Training set   3243.47 31474.37 24188.22 -0.572395 10.305204 0.4019854
```

```
## Test set     -33469.61 53828.48 41542.76 -6.347585  9.326605 0.6904015
##                     ACF1
## Training set 0.008740233
## Test set              NA
```

ARIMA Holdout Sample Error

```
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set  -356.2665 36761.53 24993.04 -1.802137 9.824411 0.4153609
## Test set     27271.5199 33999.79 27271.52  6.183294 6.183294 0.4532270
##                    ACF1
## Training set 0.01641446
## Test set             NA
```

Forecast Results

Per the table below, the forecasted monthly video game sales for the next four months are:

- Oct 2013: $747,868
- Nov 2013: $805,244
- Dec 2013: $578,736
- Jan 2014: $573,014

```
##          Point Forecast    Lo 80    Hi 80    Lo 95     Hi 95
## Oct 2013       747868.1 608137.7 887598.4 534168.9  961567.2
## Nov 2013       805244.4 623012.4 987476.3 526544.7 1083944.1
## Dec 2013       578735.9 428216.8 729255.0 348536.8  808935.0
## Jan 2014       573014.4 406742.6 739286.3 318723.6  827305.3
```