Lab 4

James Beck, Colby Carter, Andrew Lam

12/9/2017

The goal of this lab is to build a time series model and conduct a monthly, 11-month ahead forecast of the series in 2015. In the report below, we will describe the following steps:

- Data Preparation
- Exploratory Data Analysis (EDA)
- Model Building
- Model Evaluation
- Forecast Generation

Data Preparation

We begin by converting the data into an *xts* object with a monthly time index.

```
# read in csv as dataframe
df <- read.csv("Lab4-series2.csv", header = TRUE, stringsAsFactors = FALSE)
# set start and end date
d.start <- as.Date("1990/1/1")
d.end <- as.Date("2015/11/1")
# create a sequence of months from start to end
months.seq = seq(d.start, d.end, "months")
# convert to xts
df_xts <- xts(df$x, order.by = months.seq)
```

We then split the data into a training set for observations prior to 2015 and a test set from January through November of 2015, over which we will compare our model's forecast.

```
xts_train <- df_xts["1990-01-01/2014-12-01"]
xts_test <- df_xts["2015-01-01/2015-11-01"]
```

Exploratory Data Analysis (EDA)

We explore our dataset by looking at sample observations at the beginning and end of the series, along with sample statistics. While the units of the series is unclear, it ranges from 3.76 to 9.59 with a mean of 6.3.

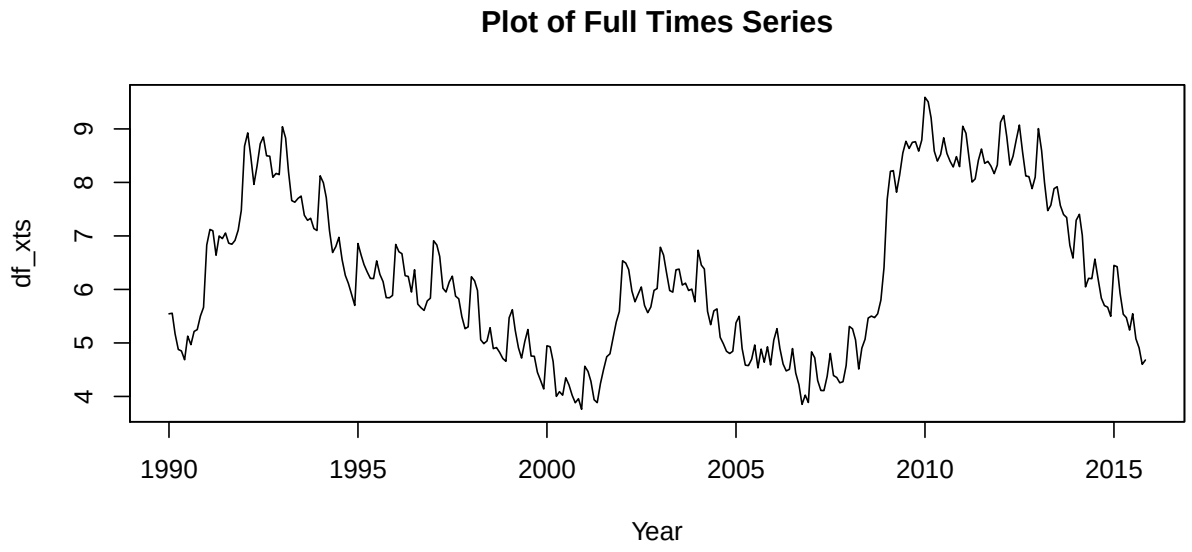
```
head(df_xts)
```

```
##             [,1]
## 1990-01-01 5.544
## 1990-02-01 5.555
## 1990-03-01 5.172
## 1990-04-01 4.878
## 1990-05-01 4.851
## 1990-06-01 4.686
```

```
describe(df_xts)
```

```
##    vars   n mean   sd median trimmed  mad  min  max range skew kurtosis
## X1    1 311 6.27 1.49   6.01    6.21 1.64 3.76 9.59  5.83 0.36    -1.01
##      se
## X1 0.08
```

```
plot.zoo(df_xts, main = "Plot of Full Times Series", xlab = "Year")
```

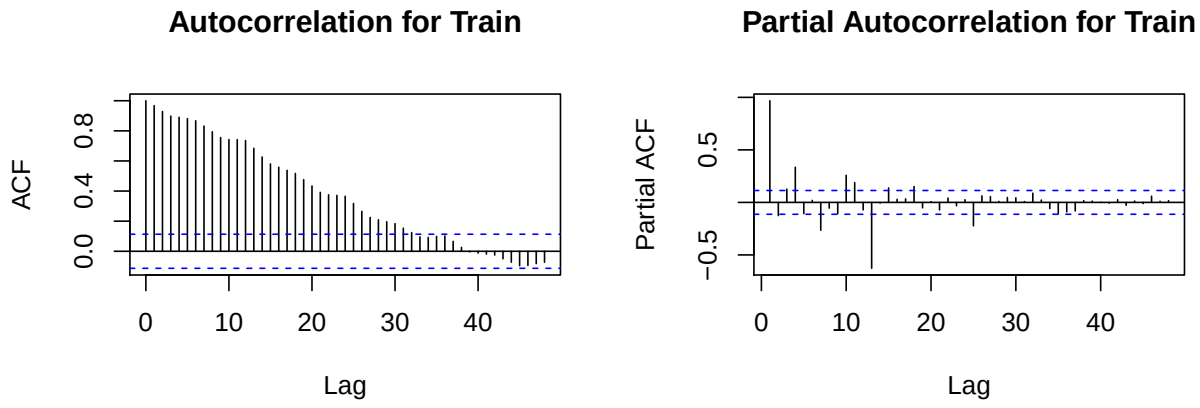


From the plot of the training data, we can see that there is downward trend from 1992-2001, 2002-2007, and 2010-2015. In addition, there also appears to be high volatility at the monthly cadence, which we revisit shortly. Given the volatility of the time series, we see a fairly wide frequency distribution with the bulk of the dataset falling between 3 and 7.

```
hist(xts_train, main = "Histogram of Training Data", xlab = "Training Data")
```

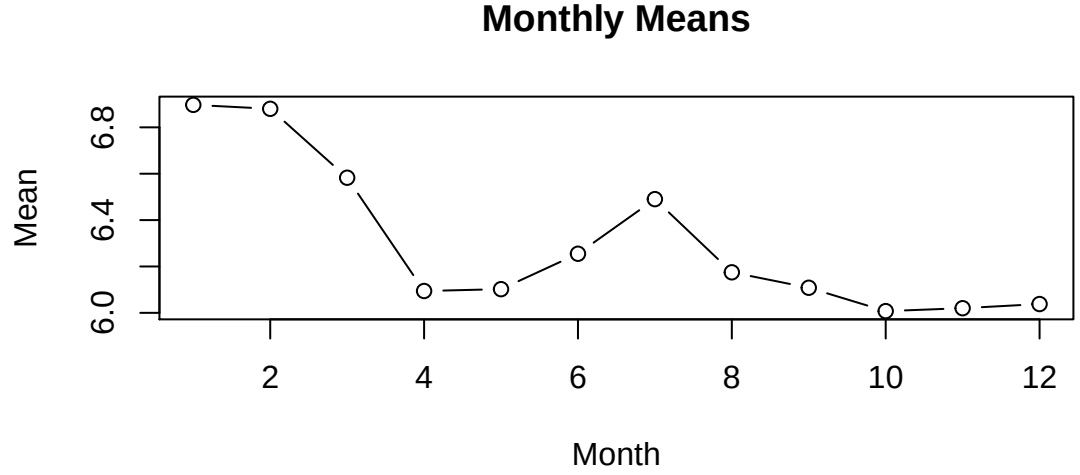
Looking at autocorrelation functions across all lags in the training sample, we confirm that there is likely autocorrelation and seasonal components:

```r
par(mfrow = c(1, 2))
acf(xts_train, 48, main = "Autocorrelation for Train")
pacf(xts_train, 48, main = "Partial Autocorrelation for Train")
```



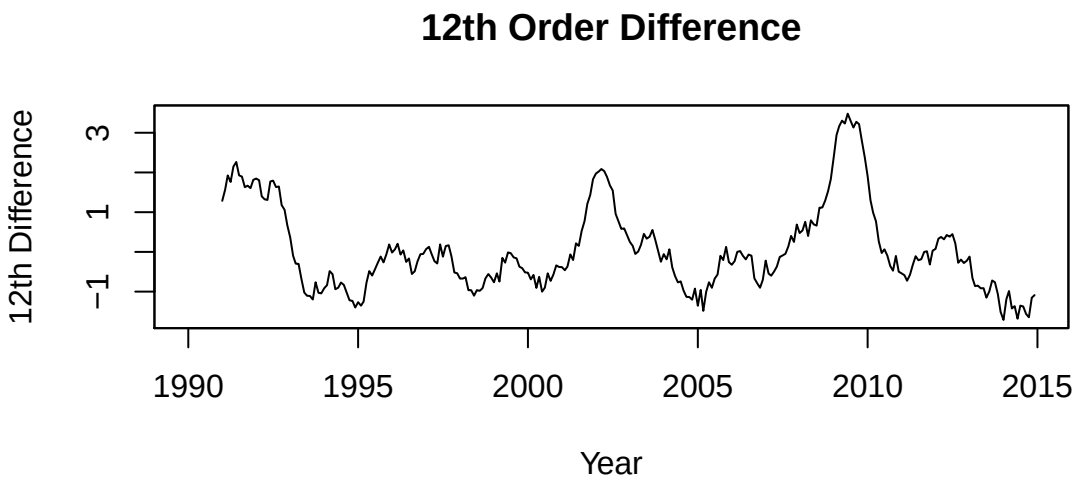
The ACF for the training data tails off slowly and is significant through lag 32, while the PACF appears to be oscillating and somewhat tailing off. There are also descending spikes at lags 12 and 24, suggesting a seasonal trend. We look more closely at any relationship between the months of the year:

```r
monthly <- matrix(xts_train, ncol = 12, byrow = TRUE)
monthly_means <- apply(monthly, 2, mean)
plot(monthly_means, type = "b", main = "Monthly Means", xlab = "Month",
    ylab = "Mean")
```



A likely seasonal component is confirmed by the monthly mean plot where we see that there appear to be cycles of different monthly levels (e.g. higher values at months 1, 2, 3, and 7).

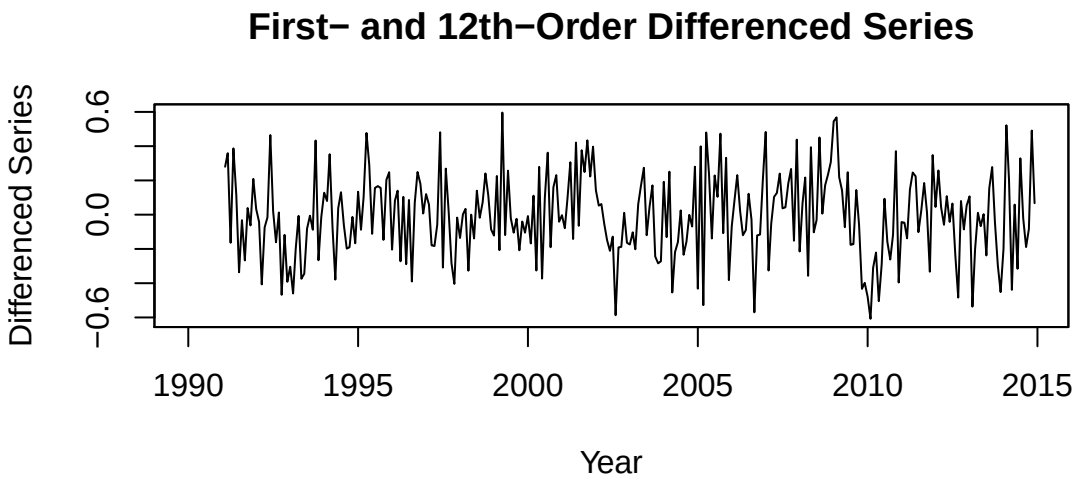
```r
# Calculate 12th difference
xts_train_d12 <- diff(xts_train, lag = 12)
plot.zoo(xts_train_d12, main = "12th Order Difference", xlab = "Year",
    ylab = "12th Difference")
```



So we opt to take a 12th order difference in order to remove the identified seasonality. The resulting series still does not appear to be stationary in the mean and variance, with spikes in the early part of the series and in 2002 and 2009, so taking a non-seasonal difference may be necessary.

We then move forward by taking our 12th order differenced series that helped mitigate the seasonality from earlier and additionally take a first order difference. The resulting twice-differenced series is shown below.

```
xts_train_d1_12 = diff(xts_train_d12, lag = 1)
plot.zoo(xts_train_d1_12, main = "First- and 12th-Order Differenced Series",
    xlab = "Year", ylab = "Differenced Series")
```



After taking the 12th order and 1st order difference, the series appears to be stationary in the mean and variance. Based on visual inspection, the plotted series more closely resembles a white noise series, and the Augmented Dickey-Fuller test rejects the null hypothesis, showing strong evidence that new series is stationary.

```
adf.test(xts_train_d1_12["1991-02-01/2014-12-01"])
```

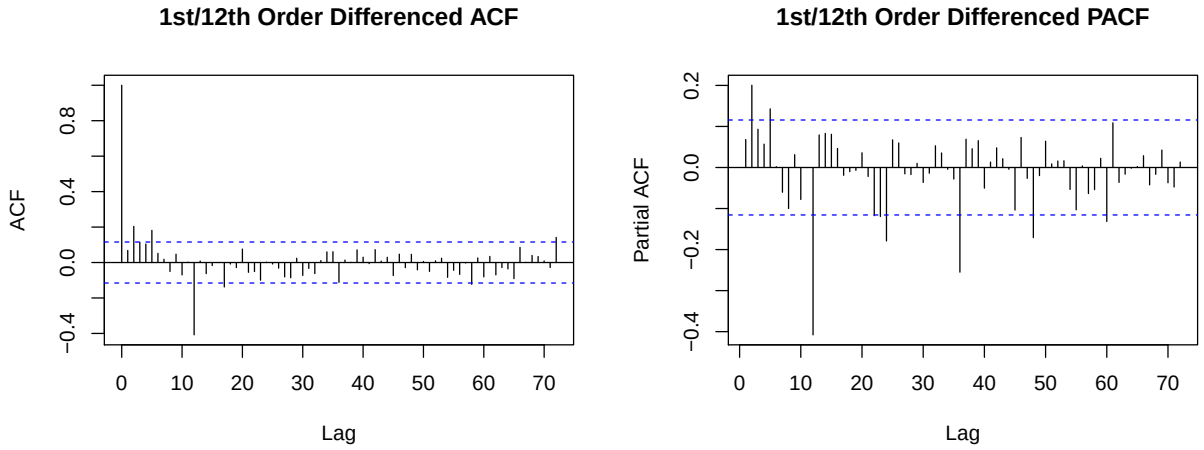
```
## Warning in adf.test(xts_train_d1_12["1991-02-01/2014-12-01"]): p-value
```

```
## smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  xts_train_d1_12["1991-02-01/2014-12-01"]
## Dickey-Fuller = -4.8865, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

With our new series, we turn back to the ACF and PACF to identify any seasonal or non-seasonal autoregressive and moving average processes.

```
par(mfrow = c(1, 2))
acf(xts_train_d1_12["1991-02-01/2014-12-01"], 72, main = "1st/12th Order Differenced ACF")
pacf(xts_train_d1_12["1991-02-01/2014-12-01"], 72, main = "1st/12th Order Differenced PACF")
```



Seasonal Behavior: Looking at lags 12, 24, and 36, we notice that there is a significant spike at lag 12 in the ACF, but nothing else after that, which may be indicative of a seasonal MA(1) component. On the other hand, in the PACF, we can see significant correlations at lags 12, 24, and 36, but in such away that may be descending; this does not lead us to believe that there is a seasonal autoregressive process.

Non-Seasonal Behavior: We can see what could either be spikes (or cliffs) in significant autocorrelations at early lags, which could also be oscillation or tapering off in later lags. So we consider testing both non-seasonal AR and MA processes for lags zero through six, which hover around the significant level in both ACF charts.

Model Building

Our data exploration has given us some indication that there is some 1st order seasonlity on a period of 12 months as well as a 1st order non-seasonal difference required to transform the series into something that is stationary in mean and variance. We have strong inclinations to choose Q=1 and P=0 (i.e. Seasonal MA(1) process) given our observations from the previously displayed ACF and PACF charts. We have some suspicions about the remaining parameters that would fully describe the model so we will move forward with exploring a finite set of possbilities based on these findings.

We choose to build a baseline SARIMA model with D=1 (12th order difference), m=12, d=1 (1st order difference), Q=1 (seasonal MA), and P=0 (seasonal AR). We will build models with varying values of p (non-seasonal AR) and q (non-seasonal MA) and then compare in-sample and out-of-sample fit to aid in the process of selecting a final model.

SARIMA: Arima(p,d=1,q) x (P=0,D=1,Q=1)[m=12]

```
ar_orders <- c(0, 1, 2, 3, 4, 5, 6)
ma_orders <- c(0, 1, 2, 3, 4, 5, 6)
AICs <- matrix(, nrow = length(ar_orders), ncol = length(ma_orders))
dimnames(AICs) <- list(ar_orders, ma_orders)
MAPEs <- matrix(, nrow = length(ar_orders), ncol = length(ma_orders))
dimnames(MAPEs) <- list(ar_orders, ma_orders)
for (p in ar_orders) {
    for (q in ma_orders) {
        sarima <- Arima(xts_train, order = c(p, 1, q), seasonal = list(order = c(0,
            1, 1), period = 12))
        AICs[p + 1, q + 1] <- round(sarima$aic, 2)
        fcast <- forecast(sarima, h = 11)
        compare <- cbind(fcast$mean, xts_test)
        MAPE <- mean(abs((compare[, 2] - compare[, 1])/compare[,
            2]) * 100)
        MAPEs[p + 1, q + 1] <- round(MAPE, 3)
    }
}
AICs
```

```
##         0       1       2       3       4       5       6
## 0  -96.70  -97.78 -108.77 -109.62 -109.61 -120.65 -120.24
## 1  -99.18 -122.71 -125.81 -123.83 -122.54 -122.11 -122.91
## 2 -113.62 -125.86 -123.86 -122.75 -121.07 -122.67 -120.98
## 3 -118.41 -123.86 -123.46 -120.50 -125.68 -123.75 -121.78
## 4 -117.94 -121.92 -120.54 -125.27 -123.77 -123.05 -121.16
## 5 -126.44 -124.51 -121.52 -125.44 -124.23 -119.06 -121.59
## 6 -124.48 -122.65 -121.19 -119.22 -117.21 -121.86 -118.16
```

```
MAPEs
```

```
##       0     1     2     3     4     5     6
## 0 7.376 7.192 6.861 6.711 6.462 5.883 5.655
## 1 7.067 3.319 3.847 3.814 4.044 4.682 3.743
## 2 6.353 3.770 3.788 3.883 3.733 4.344 4.006
## 3 5.814 3.777 3.845 3.530 3.591 3.664 3.712
## 4 5.387 3.796 4.186 3.445 3.700 3.659 3.563
## 5 4.137 4.055 6.061 4.225 4.988 4.044 5.303
## 6 4.092 3.830 3.866 3.982 3.955 4.157 4.166
```

The first matrix above shows the Akaike information criterion (AIC) for each combination of non-seasonal AR process (rows) and MA process (column), while the second matrix similarly shows the mean absolute percentage error (MAPE): the former is a measure of in-sample fit, including a penalty for adding too many parameters, while the latter is the error associated with the model's forecast compared to the values in our test dataset.

Beginning with the latter out-of-sample fit, our model with AR(1) and MA(1) components performs best on the holdout months, with a MAPE of 3.319. Then looking at AIC, the same model achieves a level of -122.7, and while not the lowest AIC of our models, it is still relatively strong while being most parsimonious. So for these reasons, we select the model with the ARMA(1,1) process:

SARIMA(p=1,d=1,q=1)x(P=0,D=1,Q=1),[m=12]

```
sarima_final <- Arima(xts_train, order = c(1, 1, 1), seasonal = list(order = c(0,
    1, 1), period = 12))
summary(sarima_final)
```

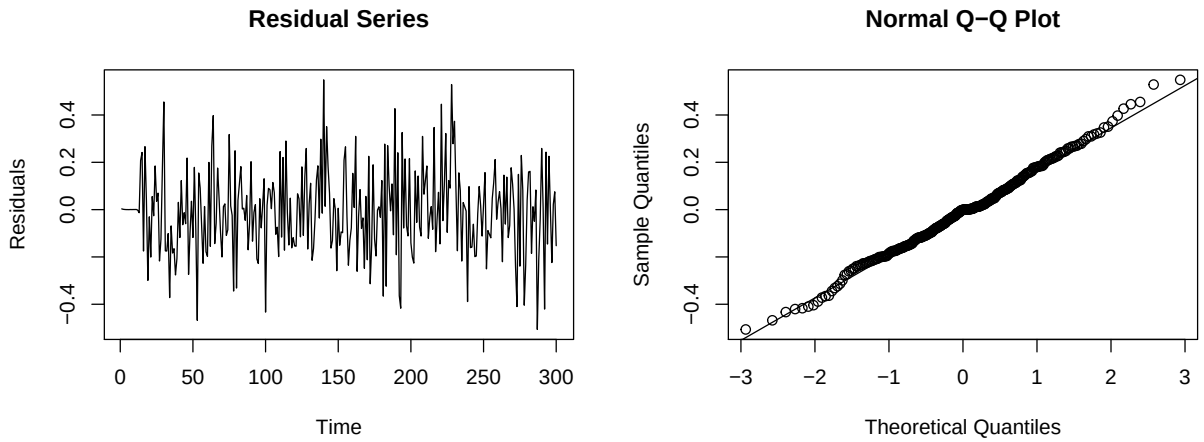
```
## Series: xts_train
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.9311  -0.8047  -0.8909
## s.e.  0.0388   0.0560   0.0520
##
## sigma^2 estimated as 0.03519:  log likelihood=65.36
## AIC=-122.71   AICc=-122.57   BIC=-108.08
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE     MAPE
## Training set -0.009339336 0.1825311 0.1427566 -0.1059365 2.389875
##                    MASE       ACF1
## Training set 0.5071314 -0.1050292
```

Looking more closely at this chosen model we make a few observations. The standardized residuals plot below appears to be mostly similar to white noise. Additionally all of the lags in the ACF are roughly within the 95% confidence bounds. Also, the Q-Q plot of the standardized residuals also looks fairly normal, with perhaps some deviation from the right tail of the normal distribution. The p-value for the Ljung-Box statistic is somewhat borderline, but is not sufficient at the 0.05 significance level to reject the null hypothesis that the residuals are independent. For a more visual assessment of the model we will plot the modeled values against our training data.

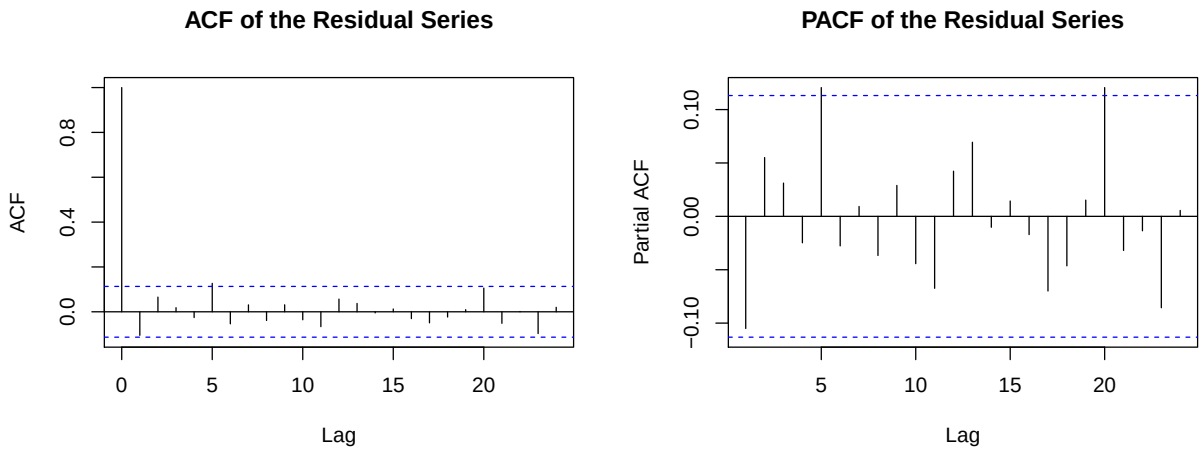
```
resids <- sarima_final$resid
# null hypothesis: independence of the residuals
Box.test(resids, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resids
## X-squared = 3.3425, df = 1, p-value = 0.06751
```

```
par(mfrow = c(1, 2))
plot.ts(resids, main = "Residual Series", ylab = "Residuals")
qqnorm(resids)
qqline(resids)
```



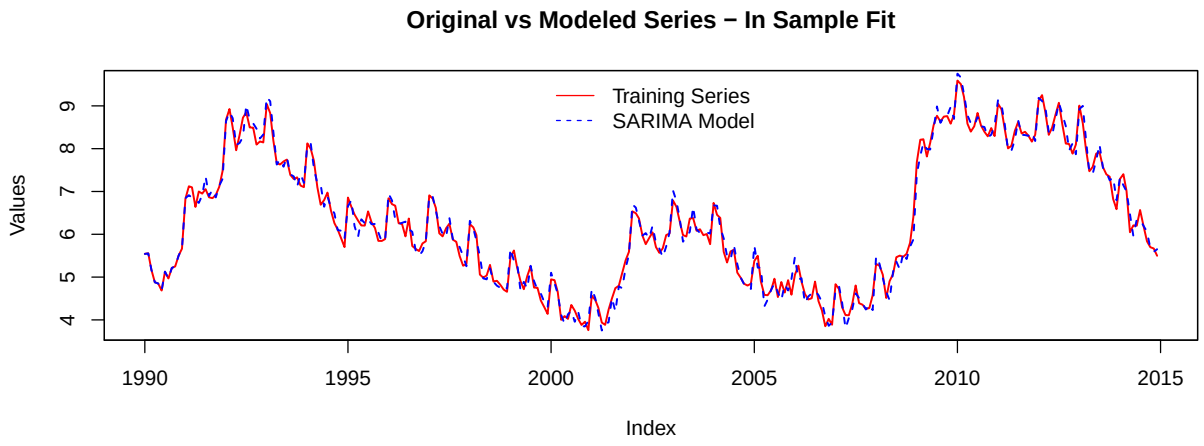
```r
acf(resids, main = "ACF of the Residual Series")
pacf(resids, main = "PACF of the Residual Series")
```



Model Evaluation

Plotting the input training data series against our modeled series yields strong results. It appears that our model closely overlaps the input data very closely with no major deviations anywhere in the series.

```r
par(mfrow = c(1, 1))
plot(zoo(xts_train, index(xts_train)), col = "red", main = "Original vs Modeled Series - In Sample Fit"
    ylab = "Values", lwd = 1.5)
lines(zoo(fitted(sarima_final), index(xts_train)), col = "blue",
    lwd = 1.5, lty = 2)
leg.txt <- c("Training Series", "SARIMA Model")
legend("top", legend = leg.txt, lty = c(1, 2), col = c("red",
    "blue"), bty = "n", cex = 1)
```



Forecast Generation

As a final test of our chosen model we evaluate its performance by forecasting beyond our input training series into the time range that we originally put aside as a test set at the start of the exercise. We compare the resulting forecast to the test set to evaluate the effectiveness of our modeling.

```
sarima_final.fcast <- forecast(sarima_final, 11, level = c(75,
    95))
compare <- cbind(xts_test, sarima_final.fcast$mean, sarima_final.fcast$lower[,
    2], sarima_final.fcast$upper[, 2])
colnames(compare) <- c("Test Data", "Forecast", "Low 95%", "High 95%")
round(compare, 3)
```

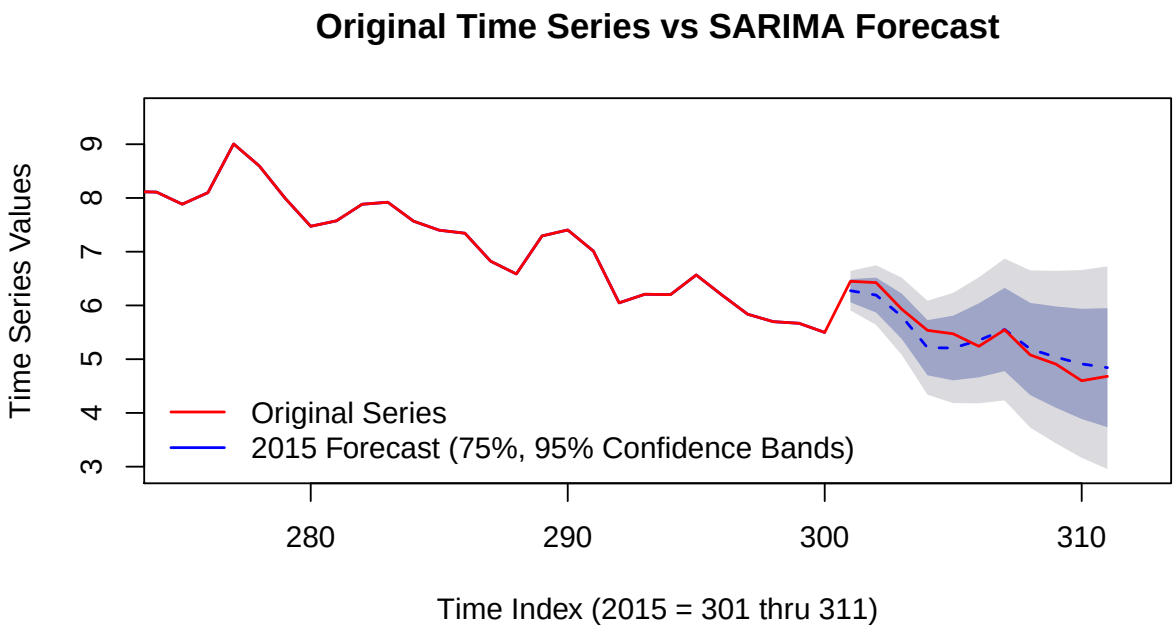
```
##       Test Data Forecast Low 95% High 95%
##  [1,]     6.449    6.274   5.907    6.642
##  [2,]     6.425    6.192   5.638    6.747
##  [3,]     5.929    5.798   5.079    6.516
##  [4,]     5.536    5.213   4.339    6.087
##  [5,]     5.472    5.208   4.183    6.233
##  [6,]     5.240    5.351   4.178    6.524
##  [7,]     5.546    5.552   4.233    6.871
##  [8,]     5.078    5.189   3.725    6.652
##  [9,]     4.907    5.038   3.433    6.644
## [10,]     4.599    4.912   3.165    6.658
## [11,]     4.681    4.842   2.956    6.728
```

```
mean(abs((compare[, 1] - compare[, 2])/compare[1]) * 100)
```

```
## [1] 2.759396
```

On our holdout dataset, our forecast has a mean absolute percentage error of 2.759%, which tracks the true values pretty closely, so we can be confident in our model's ability to forecast.

```
plot(sarima_final.fcast, main = "Original Time Series vs SARIMA Forecast",
    xlab = "Time Index (2015 = 301 thru 311)", ylab = "Time Series Values",
    xlim = c(275, 312), col = "blue", lwd = 1.5, flty = 2, flwd = 1.5)
lines(ts(as.numeric(df_xts), start = 1), col = "red", lwd = 1.5)
leg.txt <- c("Original Series", "2015 Forecast (75%, 95% Confidence Bands)")
legend("bottomleft", legend = leg.txt, lty = c(1, 1, 2), lwd = c(1.5,
    1.5, 1.5), col = c("red", "blue"), bty = "n", cex = 1)
```



By plotting the forecasted values against the test data we set aside at the start of this exercise, we obtain further confidence in our choice of model. There are small deviations from the test series, but overall the SARIMA forecast follows the test data quite closely. In fact, all of the test data points fall within the forecast's 75% confidence interval. It is expected that forecasting well beyond this 11-step range would start to yield larger deviations from expected values (given the widening confidence intervals, relying on fewer true observations), but for the range of interest in this exercise, a SARIMA(p=1,d=1,q=1)x(P=0,D=1,Q=1),[m=12] model appears to fit the provided data well.