Time Series Analysis of Australian Minimum Daily Temperatures (1981-1990)

Benjamin Drabeck

Department of Probability and Statistics, University of California Santa Barbara

PSTAT 174: Time Series Analysis

Dr. Raya Feldman

December 9th, 2023

Abstract

This report presents a comprehensive analysis of forecasting the minimum daily temperature in Melbourne, Australia, using a dataset from 1981-1990. This project aims to reveal patterns, trends, and seasonal variations in temperature, crucial for climate analysis and planning. The data, sourced from the Australian Bureau of Meteorology, was initially collected daily but aggregated to monthly averages for clarity. This study primarily employs Seasonal Autoregressive Integrated Moving Average (SARIMA) modeling to forecast the next 12 months, focusing on the dataset's strong seasonality. The process involves data transformation, differencing, and comprehensive model selection based on AICc values. Various SARIMA models are evaluated for their fit, stationarity, and forecasting accuracy, culminating in a robust approach to predicting temperature trends.

Introduction

Unlike financial markets, which are often unpredictable due to outliers and black swan events that can trigger recessions and affect daily derivative prices, weather tends to be more predictable, even considering factors like climate change. In this project, we explore the possibilities of forecasting the minimum daily temperature (in Celsius) in Melbourne, Australia in a dataset spanning from 1981-1990 utilizing R Studio. The significance of this dataset lies in its potential to reveal patterns, trends, and seasonal variations in temperature, which are crucial for climate analysis, agricultural planning, and environmental forecasting. This data is particularly fascinating, as Australia experiences seasons in "reverse" compared to the United States in the northern hemisphere. The source of the data is credited as the Australian Bureau of Meteorology and I initially found it on machinelearningmastery.com. While the data was initially collected daily, I aggregated it to a monthly average to reduce noise. This diminished the size of the dataset from 3650 daily observations to 120 months, reducing the periods and allowing clearer monthly cycles. The first nine years of the data set was used for a training dataset with the last year reserved for the testing dataset. Applying differencing to address strong seasonality, I was able to use SARIMA modeling to forecast the next 12 months of data and hypothesize the accuracy of the model. Forecasting with the SARIMA models highlighted the importance of data preprocessing and model selection in time series analysis, despite limitations in capturing short term trends and balance between model accuracy and simplicity.

Part I: Initial Data Analysis

After importing the raw data as a csv into R-Studio, it's important to ensure the data is in a format that we can utilize. Using the as.Date function, I can ensure R doesn't interpret the recorded dates as a string type. I then plotted the data as an "xts" object as a default time series object was not providing an adequate graph. See Figure 1 Below

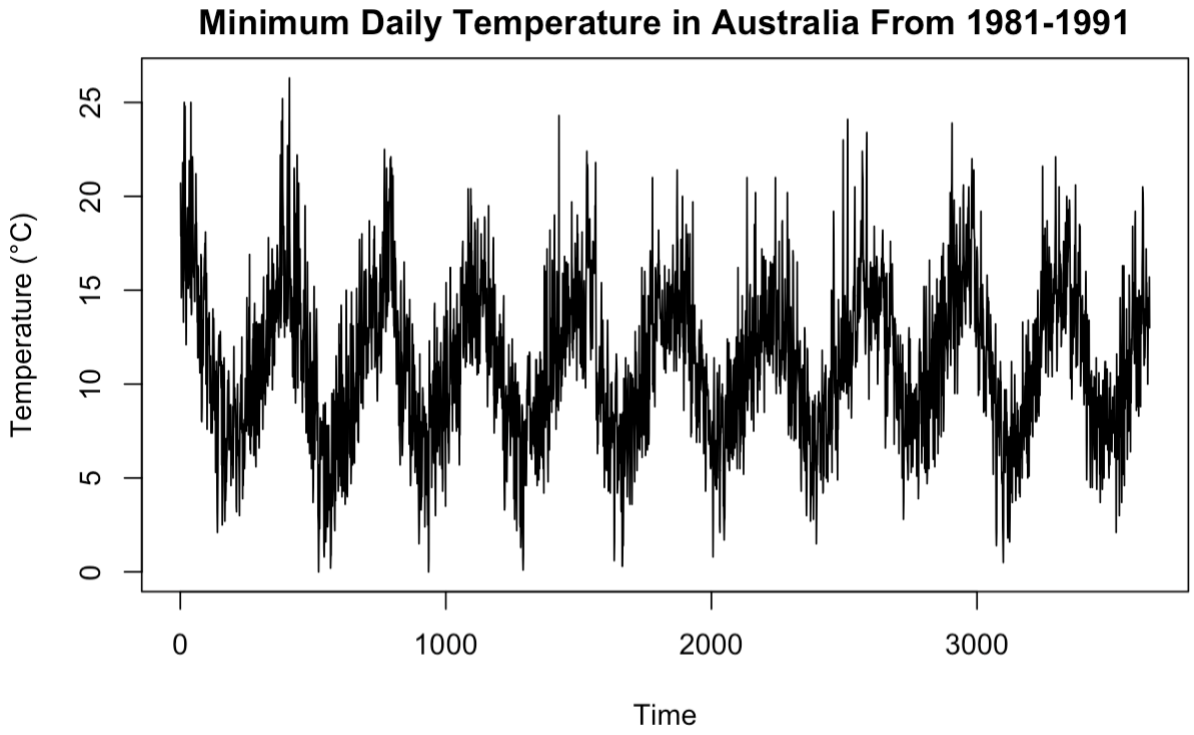


Figure 1: Time Series Plot of the initial Minimum Daily Temperature in Melbourne, Australia Dataset from January 1st, 1981 to December 31st, 1991

Concluding from the plot above, daily observations are too frequent, providing excess noise and making monthly observations unclear. Utilizing the dataset as an "xts" object, I aggregated the data monthly to smooth the 3650 observation dataset into 120 monthly averages. I then split the first 108 months into a training dataset, and used the remaining 12 months for a

testing set. The "xts" object was then converted to a time series object to ensure future functions would work as intended. The plot of the monthly averaged minimum temperature dataset is shown below via Figure 2.

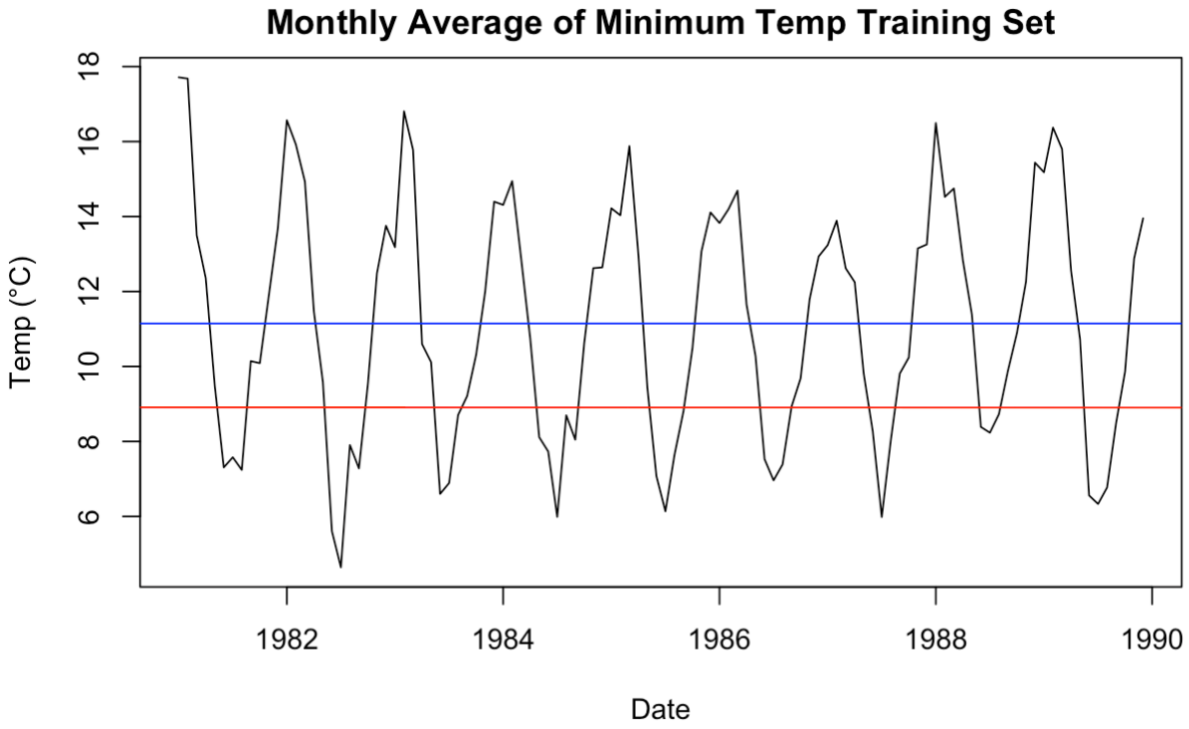


Figure 2: Time Series Plot of the Monthly Averaged Minimum Daily Temperature Training Set

The data now appears to be much smoother, and at an initial glance, suggests a constant variance. The blue mean line is centered at around 11° Celsius with a horizontal regression line at 9° Celsius indicating no linear trend. The data exhibits clear seasonal patterns, with peaks indicating summer months and lows inferring a winter season. These summer peaks are at the start of each year, demonstrating Australia's "reverse" seasons compared to the Northern Hemisphere.

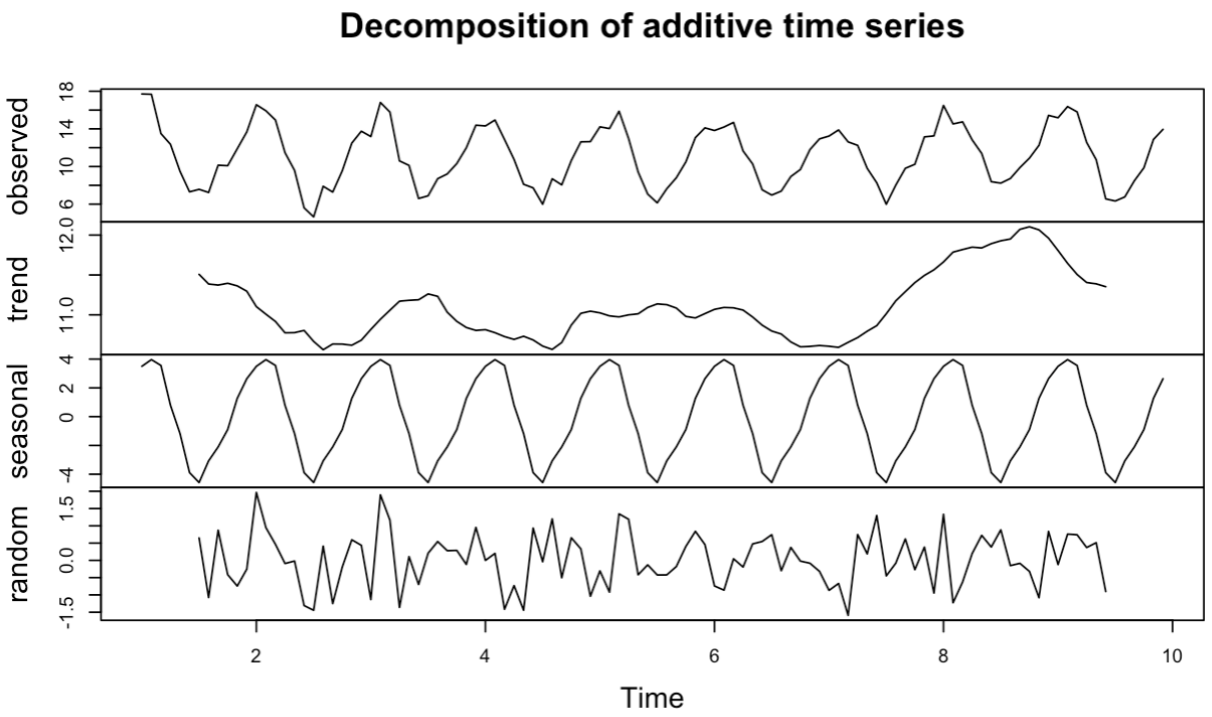


Figure 3: Decomposition of Training Data

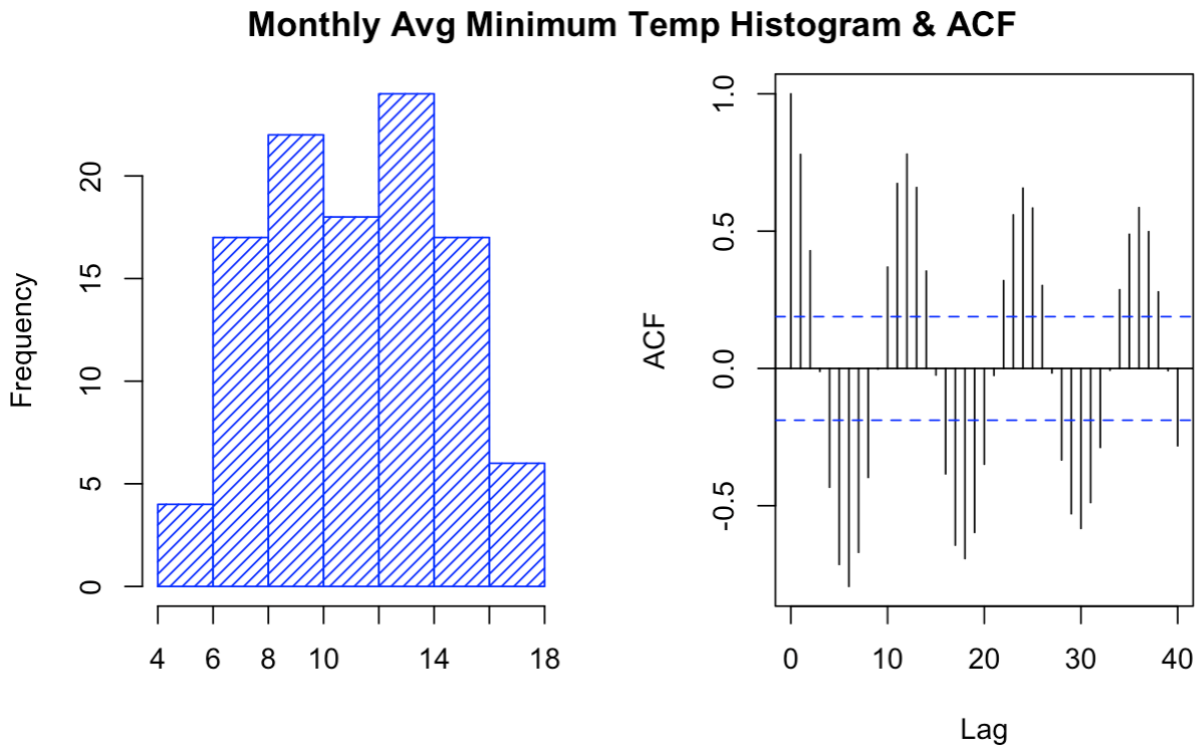


Figure 4: Histogram and Auto Correlation Function Graph of Training Data

Figure 3 illustrates the decomposition of the training data into three components: observed, trend, and seasonal. The observed component captures the actual recorded data points, displaying the raw time series data with all its inherent fluctuations. The trend component reveals a more smoothed curve, reflecting the underlying direction or pattern in the data over time, with no seasonal effects or random noise. The trend shows some upward and downward movements, indicating possible cyclic behavior but no clear long-term upward or downward trajectory. The seasonal component shows a clear and consistent pattern, repeating at regular intervals, which implies seasonality. This seasonality appears to be stable over time, suggesting a predictable seasonal influence on the temperatures. Lastly, the random component, which represents the residual or unexplained noise after the trend and seasonal effects have been removed, appears to be fairly irregular and lacks any pattern, indicating white noise.

Figure 4 presents a histogram alongside an Autocorrelation Function (ACF) graph for the monthly average minimum temperature training data. The histogram displays the frequency distribution of the temperature data, indicating how often each temperature range occurs within the dataset. The shape of the histogram suggests the distribution of temperatures, which is roughly symmetric, hinting at a normal distribution of monthly average temperatures. This is a great indication that the data will not need transformations as it is inferring a bell shaped curve. We can use parametric statistical methods that assume a normal distribution and reduce the complexity of modeling. The ACF graph displays spikes at several lags that extend beyond the significance bounds (indicated by the blue dashed lines), which suggests that there are specific lags at which the data points in the time series are correlated with each other. These spikes are periodically significant at every 12 lags, confirming that the data exhibits strong seasonality. A Seasonal ARIMA (SARIMA) model could be utilized to capture this seasonality.

Part II: Data Transformation and Differencing

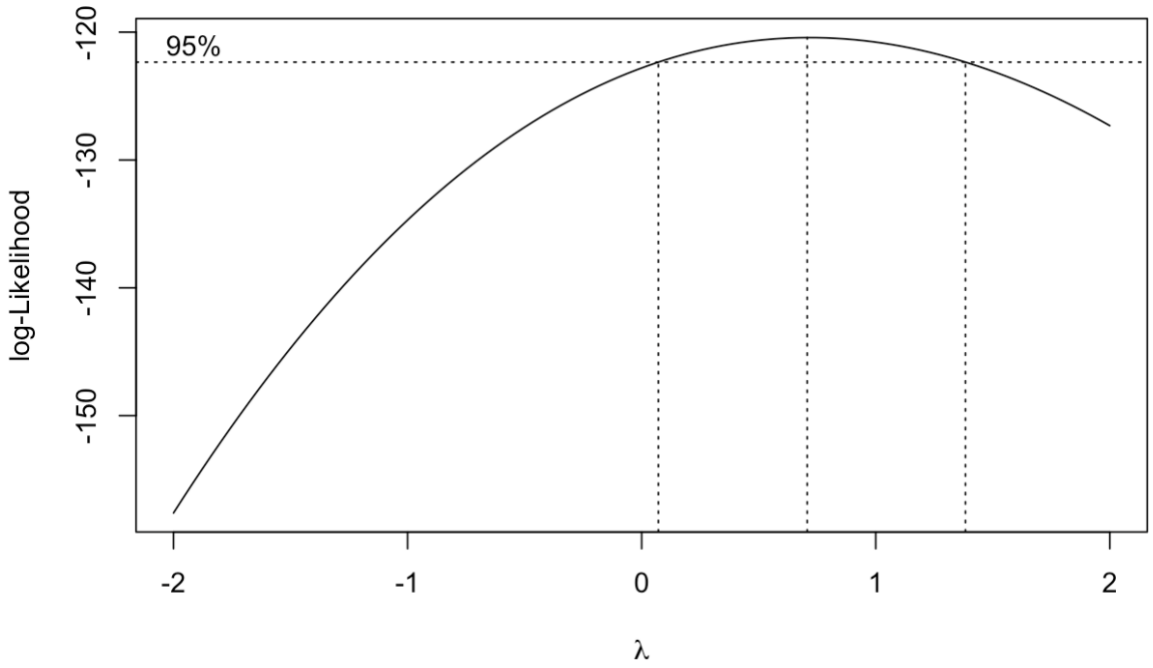


Figure 5: Box-Cox Transformation with Confidence Interval and λ = 0.7070707

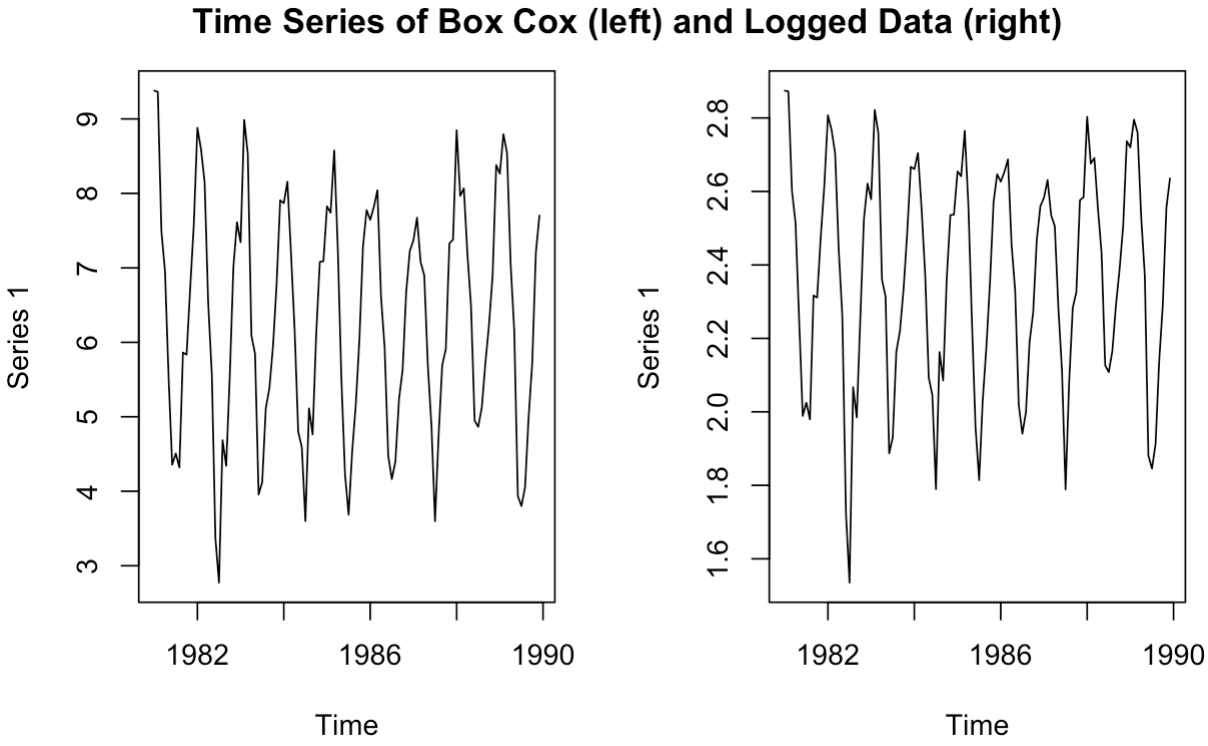


Figure 6: Time Series of Box Cox and Logged Transformed Data

Transformations like Box-Cox and logarithmic are common techniques applied to normalize data, since many statistical techniques that we will use for modeling assume normality. As seen in Figure 5, after applying a box cox transformation, we are left with a lambda value close to one. One is inside the confidence interval for lambda, so a box cox is unnecessary as the data is already close to normal distribution as predicted earlier. We only transform the data if necessary to ensure the simplicity and interpretability of the predictive model. Figure 6 represents the Box-Cox transformation (left) and the logged data (right). The Box-Cox plot shows variation in the data's spread over time, while the log transformation plot shows a more consistent spread, indicating reduced variability. Logging the data reduces the impact of outliers by compressing the range, especially for larger values, leading to a more uniform data distribution. We can further explore the normalization of these transformed datasets by using histograms to visualize a potential bell curve.

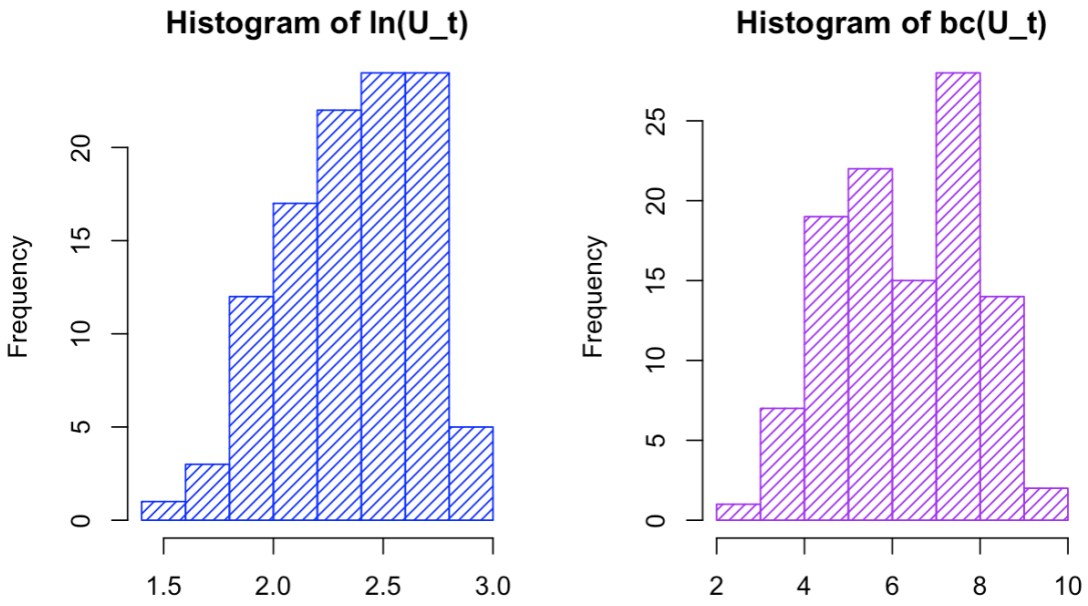


Figure 7: Histograms of Logged and Box Cox Transformed Training Data

The histograms in Figure 7 display the frequency distributions for the logged and Box Cox transformed training data. The histogram of the logged data, $\ln(U_t)$, shows a skew to the right, lacking the bell-shaped curve that characterizes a normal distribution. The Box Cox transformation, $\text{bc}(U_t)$, results in a more symmetric histogram but with bimodality (exhibiting two peaks instead of the single peak of a normal distribution). Neither transformation appears to achieve perfect normality, and due to the lambda value lying inside the confidence interval we will not use a box cox transformation. We will likely continue without transformations, but we can do a final decomposition on the logged data to rule it out.

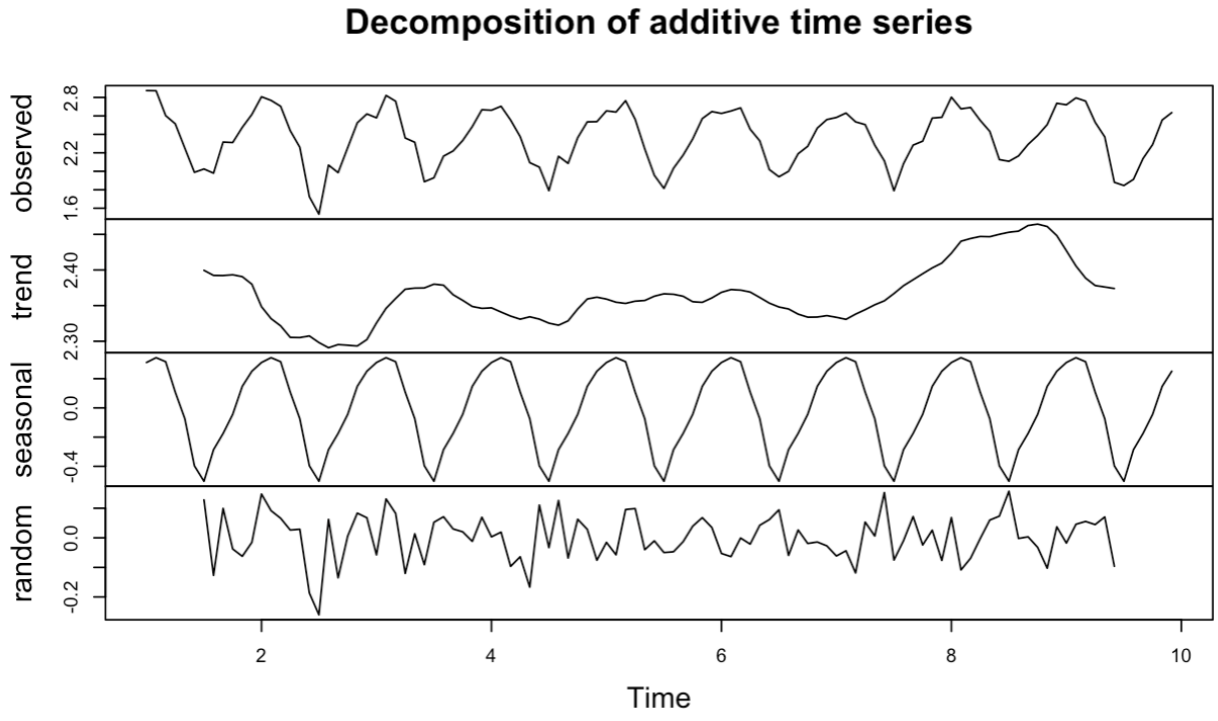


Figure 8: Decomposition of Logged Training Data

The decomposition of thelogged training data as seen in Figure 8 is very similar to the decomposition of the original training data in Figure 3. Given the observed stability, linear trend, and constant seasonal pattern, a log transformation does not seem necessary as variance appears to be relatively stable. The random component resembles white noise with a mean centered around 0 and constant variance. We can address the strong seasonal component next by differencing the original training data.

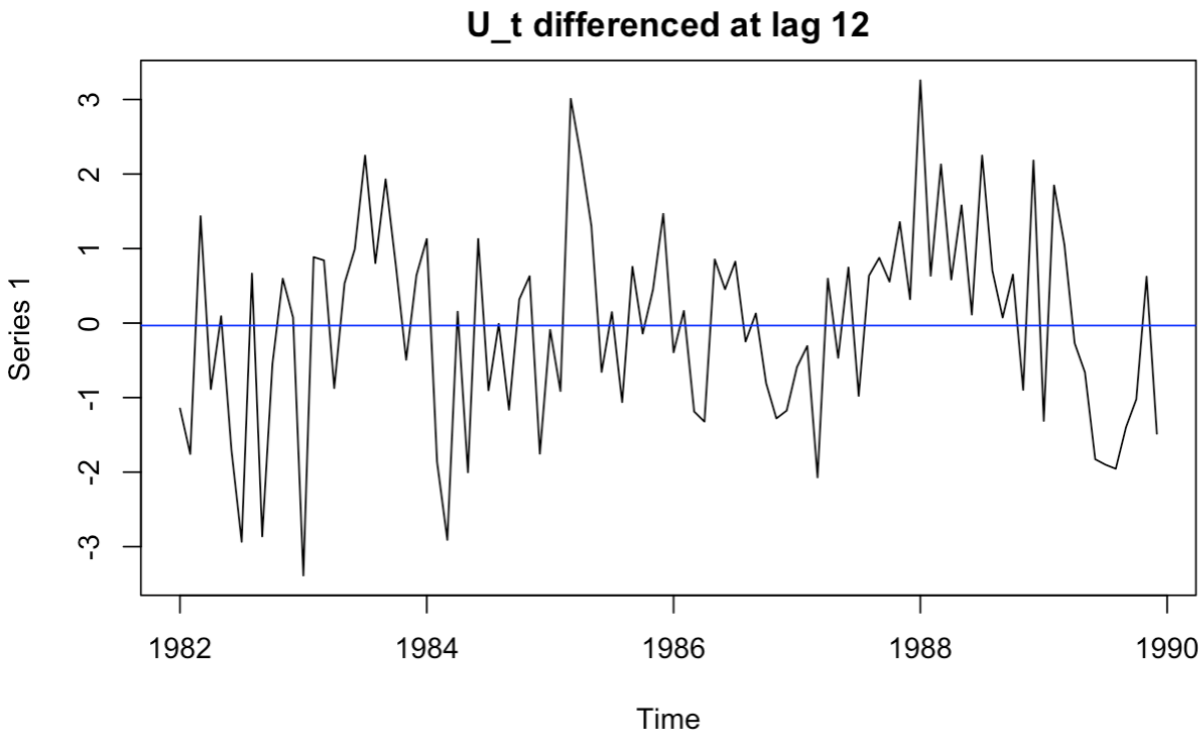


Figure 9: Differenced Training Data at Lag 12 with Mean (blue)

As previously analyzed in Figure 4, the data exhibits periodic autocorrelation spikes every 12 lags. This is common for weather data, as this demonstrates the recurring pattern of seasons on an annual basis of 12 months. I differenced the data at lag 12 to address this seasonality and the resulting time series is plotted above in Figure 9. Differencing subtracts the

current value of the series from the value at the next seasonal period, eliminating repeating patterns associated with seasonality. As indicated by the blue line in Figure 9, the mean is now about zero, which is a strong indication that the data is approaching stationarity. The trend line was having trouble being visualized on the plot so a summary statistic is provided below in figure 10. The model does not have statistically significant coefficients, as indicated by the p-values being greater than 0.05. The coefficient for index is very close to zero (0.004990), with a standard error (0.004941) almost as large, which suggests that the trend line would be nearly horizontal and very close to the intercept of -0.275066. This concludes that there is little to no trend in the data so detrending would not be required for further analysis.We can further review the ACF and PACF plots after differencing to verify reduced autocorrelation and to choose ARIMA model parameters. We can also revisit the normal distribution of the training data and compare it to the differenced data at lag 12.

```
Call:
lm(formula = mat.lag_12 ~ index)

Residuals:
    Min      1Q  Median      3Q     Max
-3.1769 -1.0133  0.1994  0.8362  3.1688

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.275066   0.275976  -0.997    0.321
index        0.004990   0.004941   1.010    0.315

Residual standard error: 1.341 on 94 degrees of freedom
Multiple R-squared:  0.01074,   Adjusted R-squared:  0.0002123
F-statistic:  1.02 on 1 and 94 DF,  p-value: 0.3151
```

Figure 10: Summary Statistic for Trend Regression Line on Differenced Training Data at Lag 12

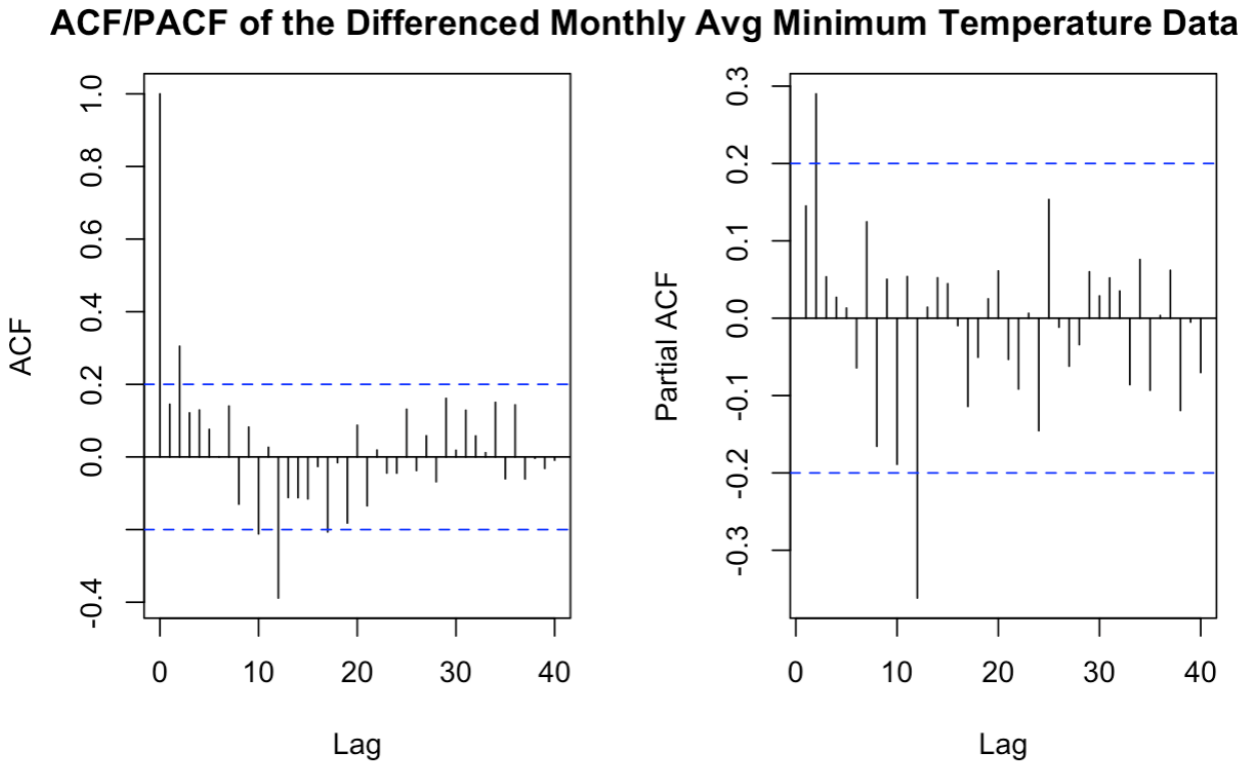


Figure 11: ACF & PACF of Differenced Training Data at Lag 12

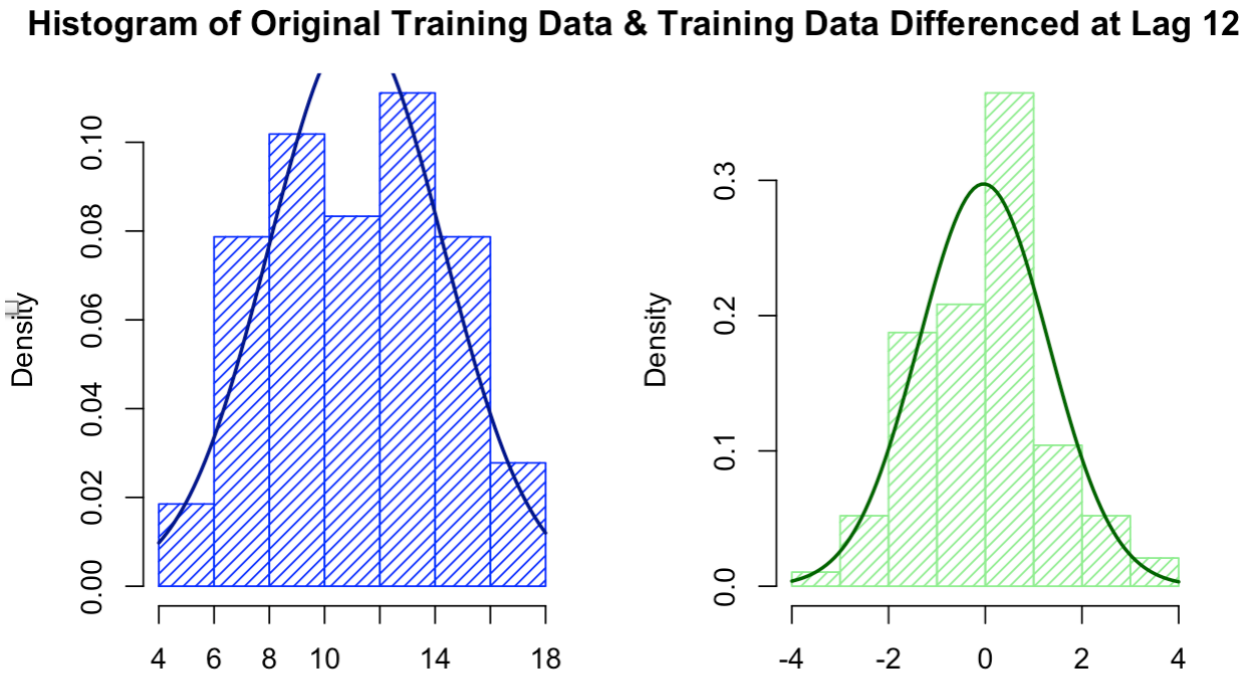


Figure 12: Histogram of Original & Differenced Training Data at Lag 12

In the ACF plot, the rapid decline in the correlation with increasing lags shows that the differencing at lag 12 has removed seasonality, as there are no significant spikes at multiples of 12. The PACF plot, which isolates the correlation at each lag after removing the effects of earlier lags, shows that most correlations are within the confidence interval, inferring no significant autocorrelation at individual lags. This concludes the data does not require additional AR or MA terms for this particular lag. In Figure 12, the histogram of the differenced data at lag 12, illustrated with green bars, shows a distribution that is more symmetric and closely aligned with the normal distribution curve, indicating that the differencing process has also normalized the data. This is contrasted with the histogram of the original training data, shown with blue bars, where the distribution is more skewed and does not follow the bell-shaped curve of a normal distribution as closely. The improved symmetry and alignment with the normal distribution in the differenced data meets the assumptions required for the modeling techniques. These plots support the conclusion that the data is now stationary and can be modeled using ARIMA without the need for further seasonal differencing or detrending.

PART III: Model Selection and Fitting

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model is chosen for its ability to model and forecast time series data that exhibits seasonality. We can effectively handle the seasonality that was indicated by the autocorrelation spikes at lag 12. As previously notated in Figure 11, we can use the ACF and PACF values outside the confidence interval to create the model, ensuring that the model accounts for any remaining patterns in the data. Specifically, the ACF indicates significant autocorrelations at lags 2, 9, 11, and 16, while the

PACF shows significant partial autocorrelations at lags 1 and 11. This created the initial choice of SARIMA(1,1,1)(1,0,0)[12], which integrates a non-seasonal AR and MA component and a seasonal AR component. We can consider s=12 and d=0 due to the lack of detrending and seasonality at lag 12. The AR1 term in the initial model was found to be statistically insignificant upon running the model, leading me to fix it at zero. Other model candidates were explored to attempt to fit the data characteristics. For instance, the model SARIMA(1,1,2)(1,1,1)[12] was considered to accommodate additional lag correlations suggested by the ACF. The potential simpler model with SARIMA(1,1,1)(0,0,1)[12] had an AR1 term set to zero due to insignificance similar to the first candidate model. Attempts to increase complexity, like SARIMA(2,1,1)(2,1,1)[12], aimed to capture additional lags but might have introduced overfitting. Non-seasonal alternatives such as ARIMA(0,1,0) and the more complex ARIMA(2,1,2) were attempted to understand the non-seasonal patterns. The SARIMA(1,1,1)(0,1,0)[12] model was a simpler seasonal approach, as it still took into account the differencing at lag 12 (D = 1). The ARIMA(2,1,0) model took out MA terms, focusing on the autoregressive components. The models and their provided AICc values are shown in Figure 13.

Model:	AICc:
1. SARIMA(1,1,1) x $(1,0,0)_{12}$:	311.37
2. SARIMA(1,1,2) x $(1,1,1)_{12}$:	319.6
3. SARIMA(1,1,1) x $(0,0,1)_{12}$:	299.42
4. SARIMA(2,1,1) x $(2,1,1)_{12}$:	312.13
5. ARIMA(0,1,0):	377.29
6. ARIMA(2,1,2):	327.94

7. SARIMA(1,1,1) x $(0,1,0)_{12}$: 375.43

8. ARIMA(2,1,0): 335.17

Figure 13: AICc values of Candidate Models

*(Models 1 & 3 AICc Values are Calculated After Fixing AR1 to 0)

AICc is a statistical measure that combines model fit and complexity, providing a way to assess the trade-off between how well a model fits the data and how many parameters it uses. We use AICc to help us select models that balance goodness of fit and model simplicity, and the lower AICc value, the better indication that the model will be useful. After comparing the AICc values of each model, we will choose models 1 and 3 due to the lowest AIC value and inclusion of seasonal components. We can write the models out mathematically as shown below in figure 14. However, we will have to analyze the models further, as AIC alone is not enough evidence to conclude an accurate or useful model. We have to check if it is stationary and invertible.

Model A: SARIMA(1,1,1) x $(1,0,0)_{12}$:

$$(1 + 0.4653_{(0.0964)}B^{12})X_t = (1 - 0.7915_{(0.0832)}B)Z_t$$

Model B: SARIMA(1,1,1) x $(0,0,1)_{12}$:

$$(1 - 0.7678_{(0.0897)}B)(1 - 0.7848_{(0.1498)}B^{12})Z_t$$

Figure 14: Mathematical Equations of Chosen Models A & B

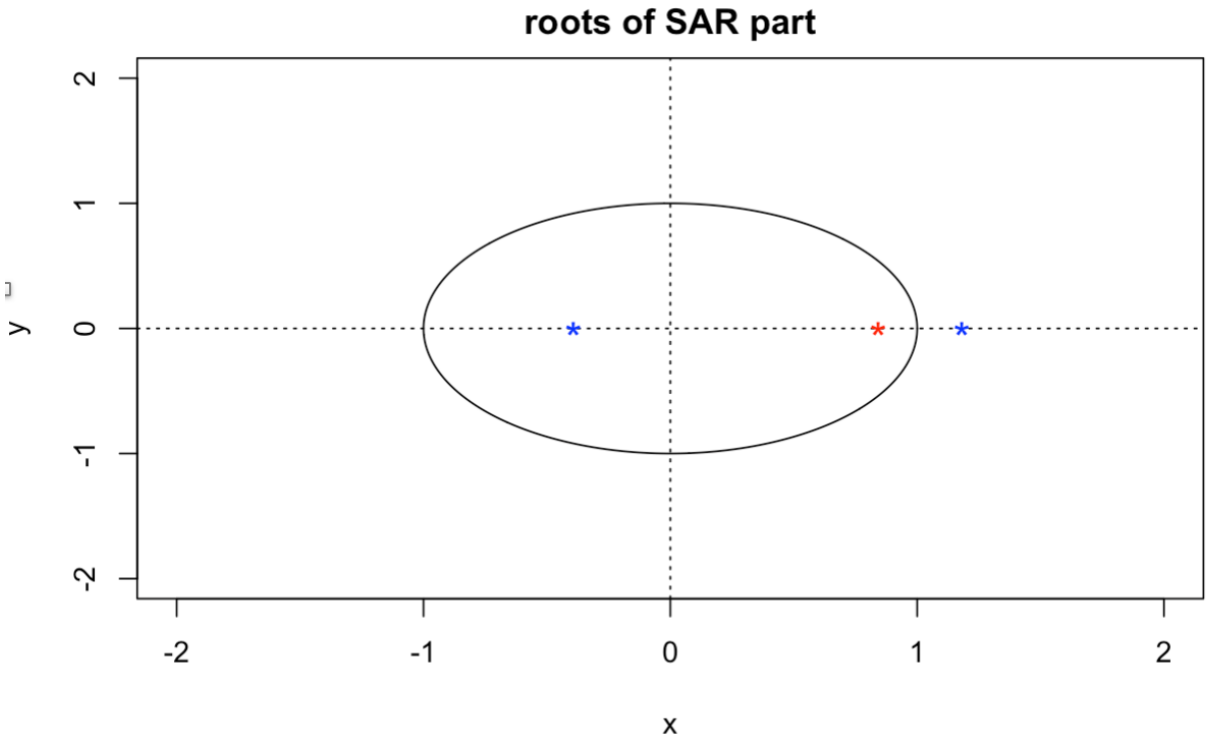


Figure 15: Model A Roots of SAR

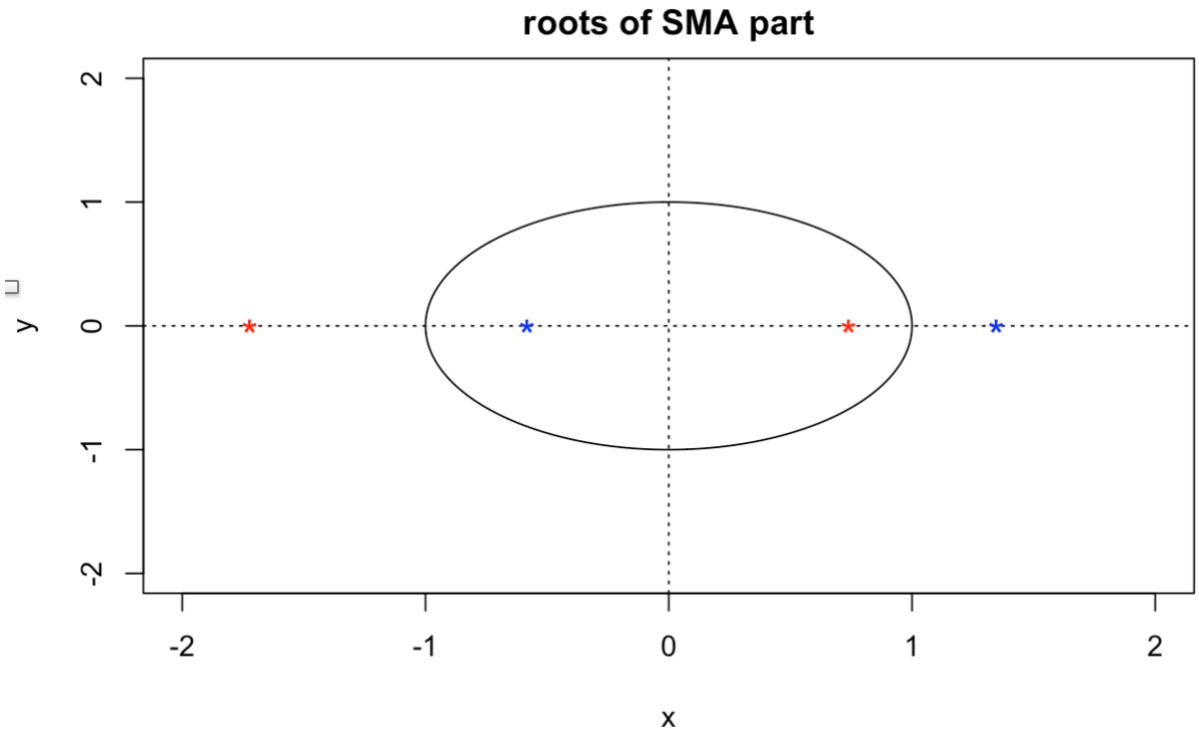


Figure 16: Model B Roots of SMA

For Model A in Figure 15, the SAR part has a root at 0.4653, which is inside the unit circle. This indicates that the SAR part of the model is stationary. The MA part has a root at 0.7915, which is also inside the unit circle, suggesting that the MA part is invertible.Model B, in Figure 16, does not have a SAR part, implying there's no seasonality to be considered in the stationarity analysis. The SMA part has a root at 0.7648, which is inside the unit circle, indicating that the SMA part is stationary. The MA part has a root at 0.7678, also within the unit circle, which confirms the invertibility of the MA component of Model B. For ARIMA and SARIMA models to be stationary and invertible, the roots of the model from the autoregressive (AR) and moving average (MA) parts must be inside the unit circle.. This requirement is necessary because it ensures that the impact of shocks on the time series decreases over time, otherwise it could exponentially increase overtime which would negatively affect the forecasting.

Part IV: Diagnostic Checking & Forecasting

Now that we have potential models that have passed stationarity and invertibility checks, we can do a final diagnostic analysis to ensure assumptions like normal distribution. This diagnostic analysis includes a Shapiro-Wilk Normality Test, Histograms, Time Series plots, and ACF/PACFs of the residuals, QQplots, Box-Pierce & Ljung Box tests, and the Yule Walker method.

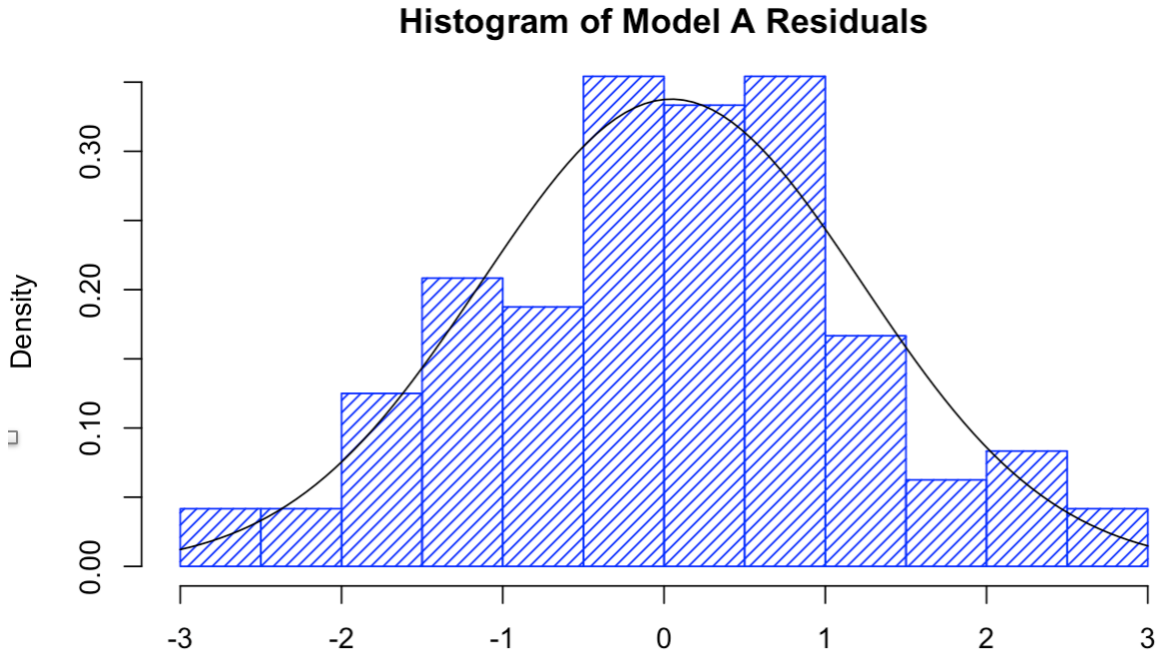


Figure 17: Histogram of Model A Residuals

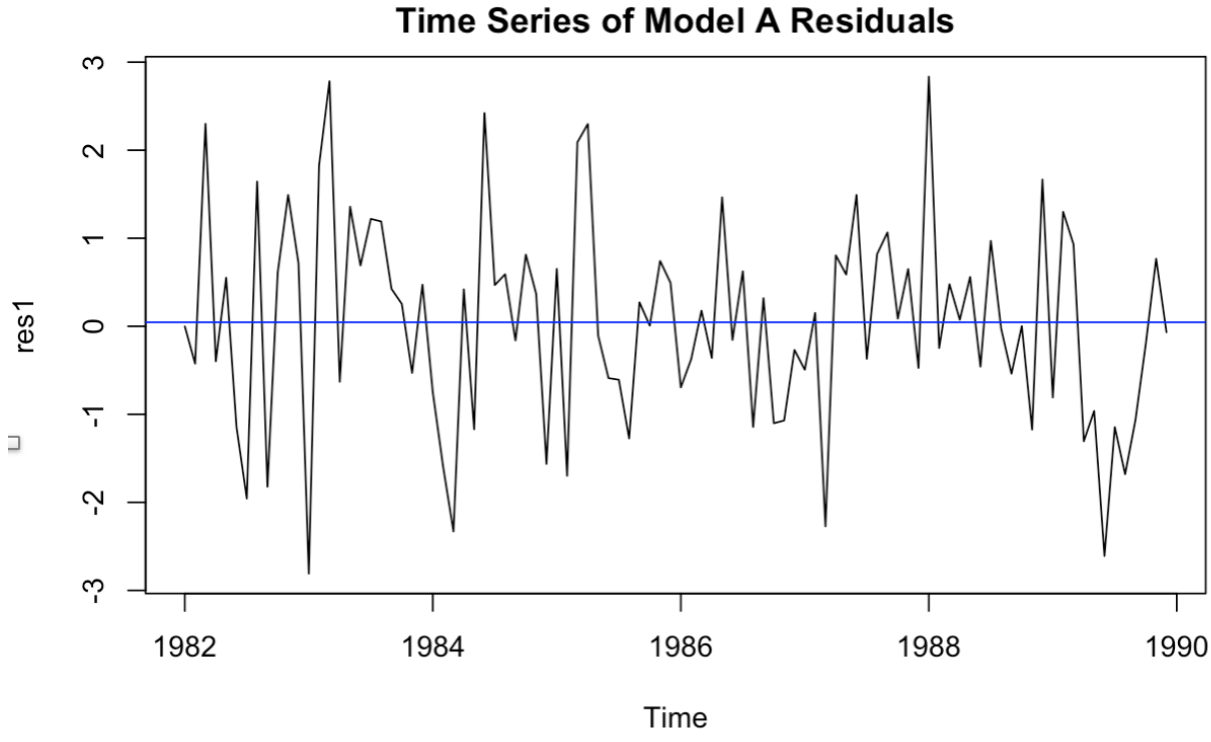


Figure 18: Time Series of Model A Residuals

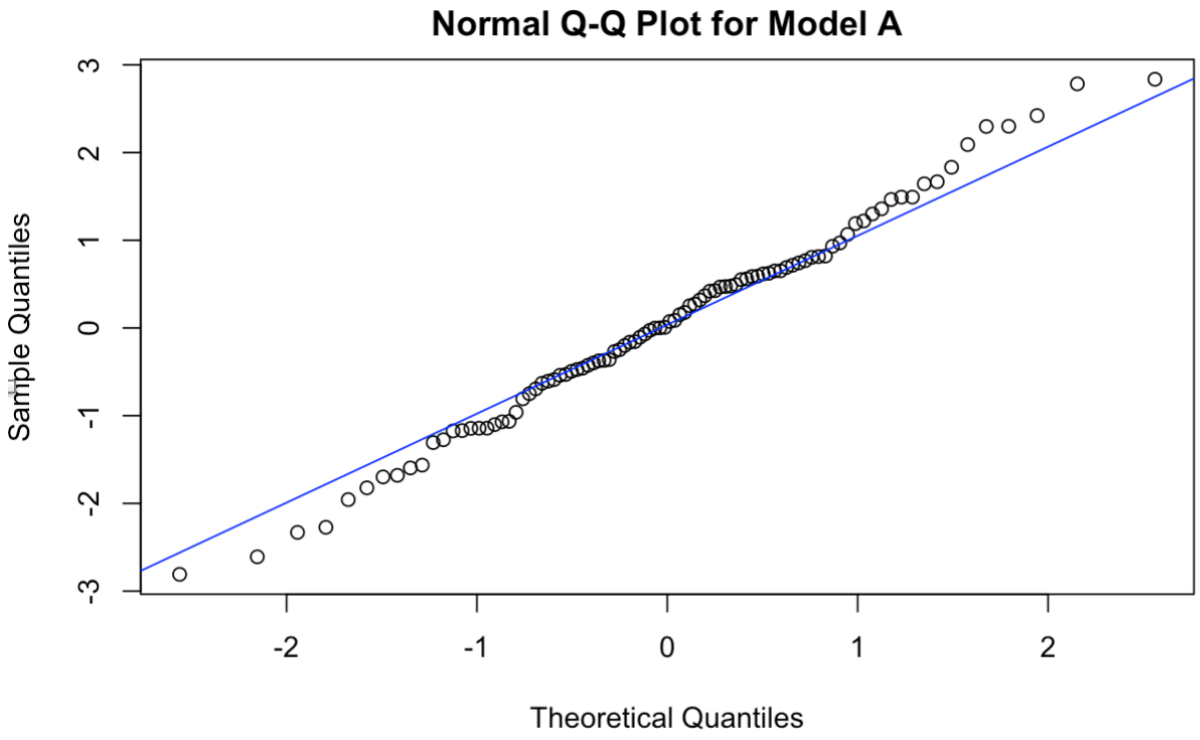


Figure 19: Normal QQ Plot for Model A Residuals

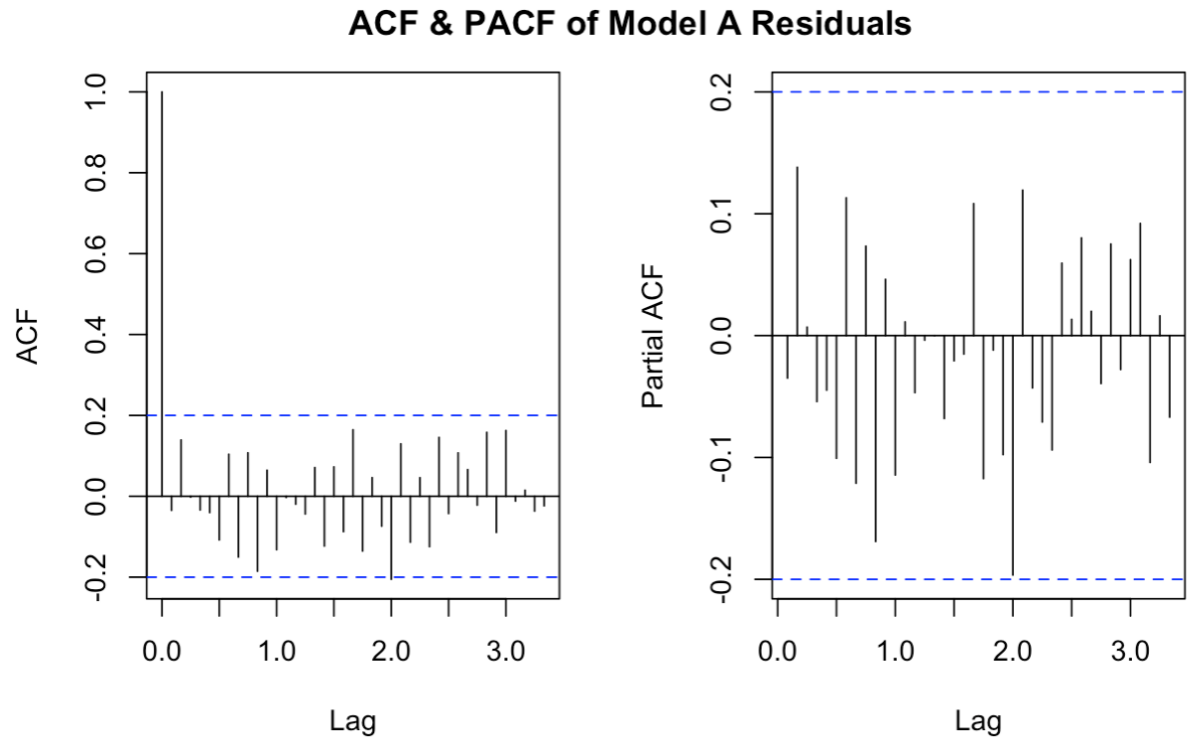


Figure 20: ACF & PACF of Model A Residuals

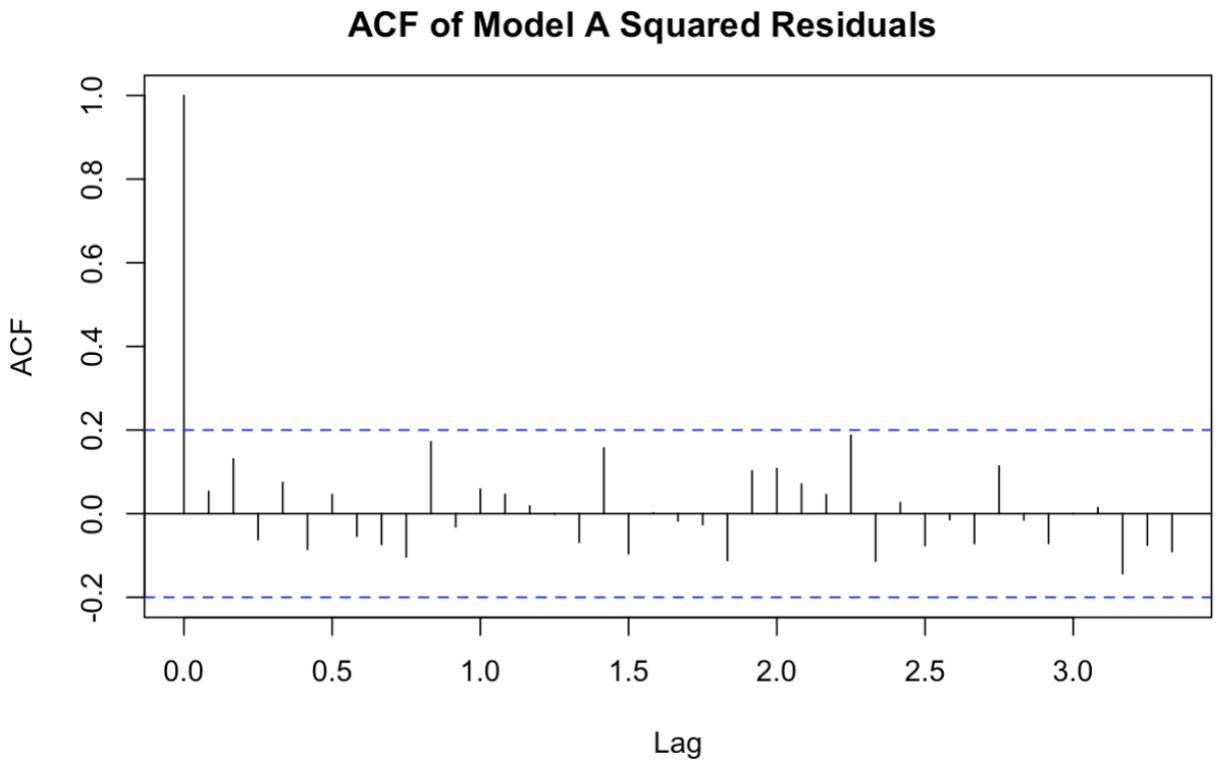


Figure 21: ACF of Model A Squared Residuals

The Shapiro-Wilk test results in a high p-value of 0.8889, indicating that we do not reject the null hypothesis of normality; hence, the residuals can be considered normally distributed. The histograms and QQ plots further support this in Figure 17 and 19 by showing the residuals are well approximated by a normal distribution, with the histogram displaying a bell-shaped curve and the QQ plot closely following the theoretical line. Time series plots of the residuals in Figure 18 show no apparent patterns or trends, suggesting that the residuals are randomly distributed over time, which is a sign that the model is capturing the underlying process well. The ACF and PACF of the residuals further confirm this, as most autocorrelations are within the confidence bounds, suggesting no significant autocorrelation. Box-Pierce and Ljung-Box tests, which are used to detect overall autocorrelation in the residuals at multiple lag lengths, both return high p-values (0.2201 and 0.149), implying that there is no significant autocorrelation in

the residuals at the lags tested. Furthermore, applying these tests to the squared residuals, also yields a high p-value (0.6417), indicating no significant autocorrelations in the squared residuals. These residuals are squared to possibly detect nonlinear dependencies. Finally, the Yule-Walker method selects an AR model of order 0, suggesting that no additional AR terms are needed and that the model does not exhibit serial correlation. Serial correlation is the relationship between a given variable and a lagged version of itself. These diagnostics indicate that the residuals of the model satisfy the assumptions of independence, homoscedasticity, and normality. This is a strong indication that model A is appropriate for the data and can be used for forecasting without concerns of violating the underlying assumptions of the model.

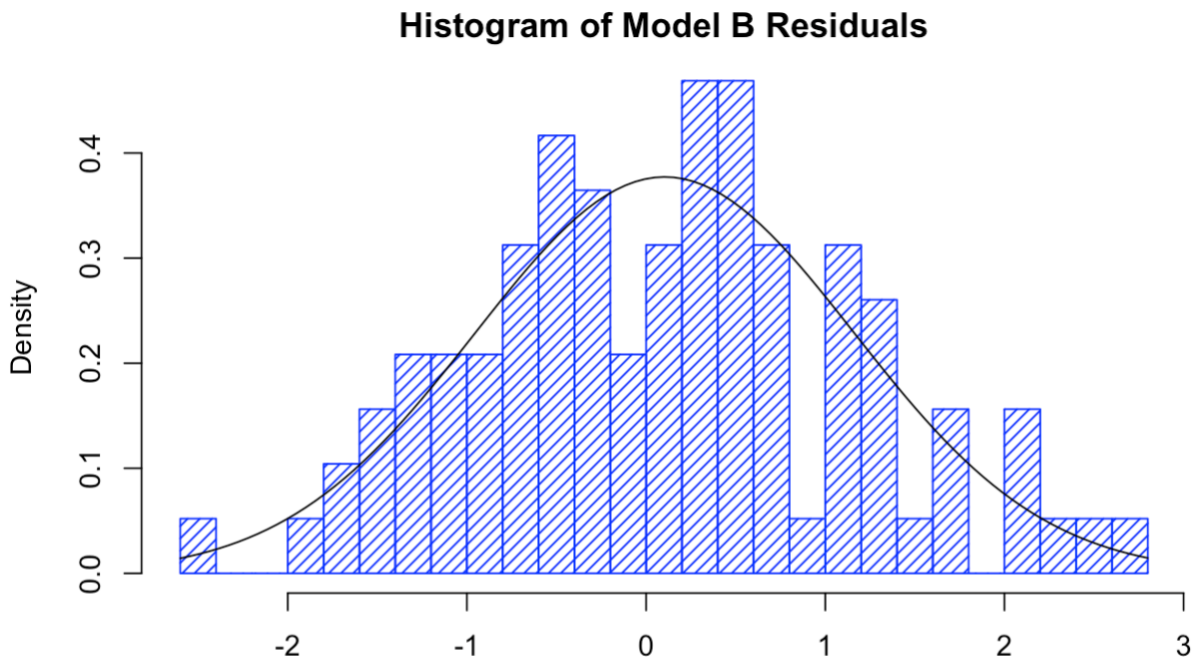


Figure 22: Histogram of Model B Residuals

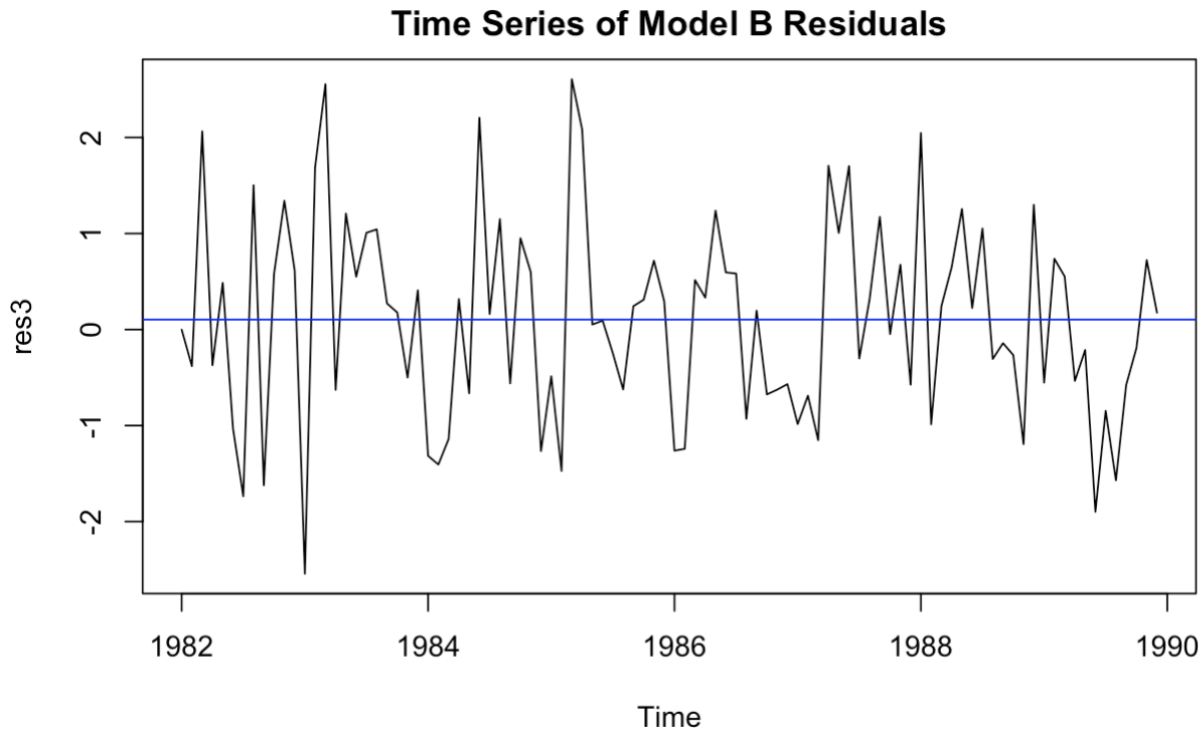


Figure 23: Time Series of Model B Residuals

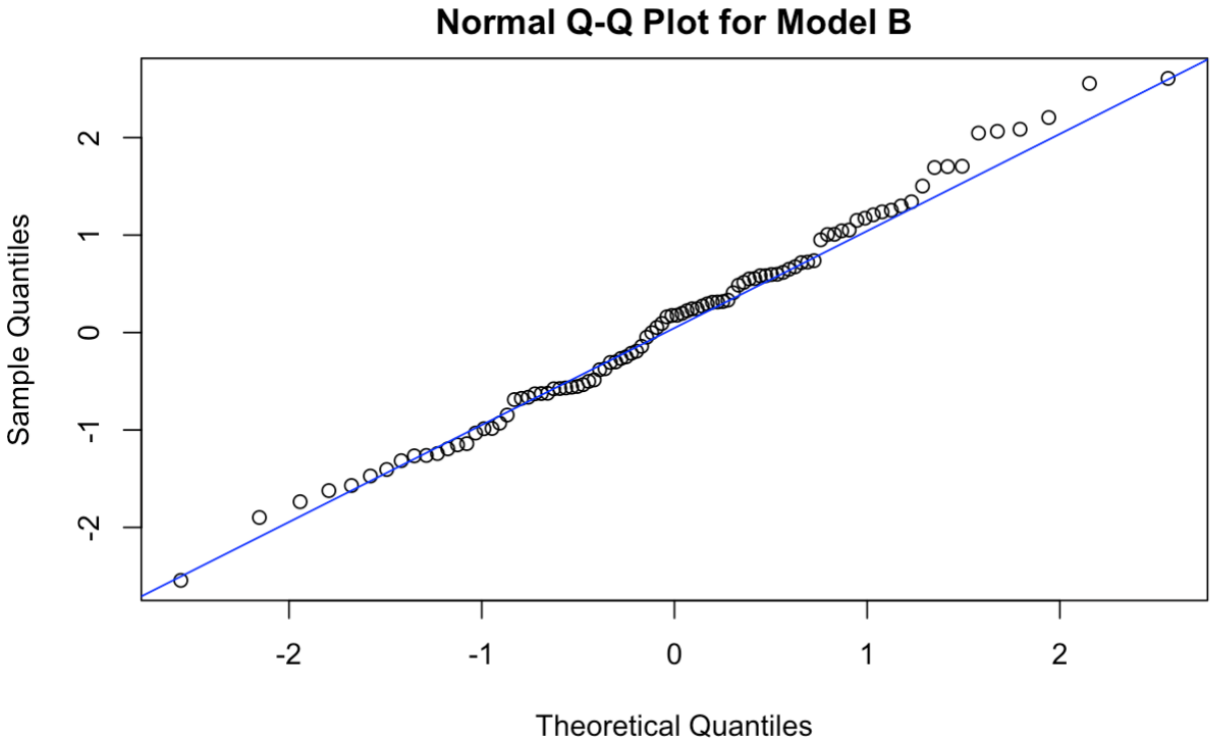


Figure 24: Normal QQ Plot for Model B Residuals

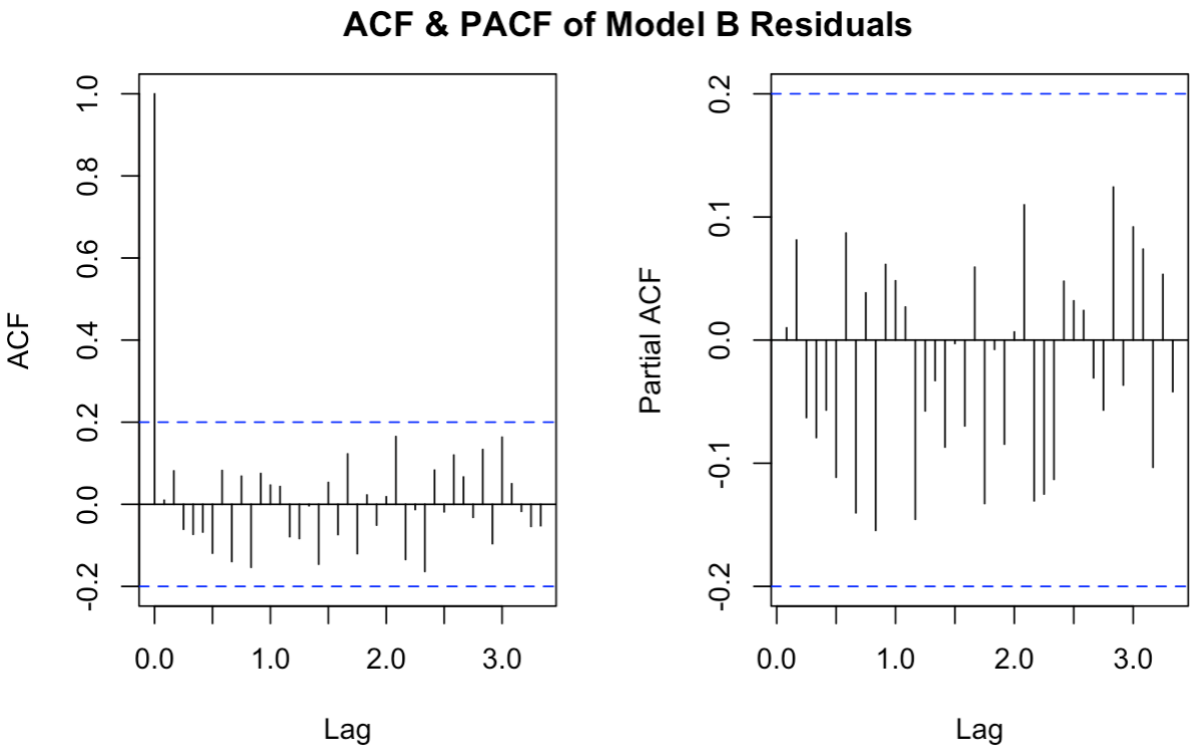


Figure 25: ACF & PACF of Model B Residuals

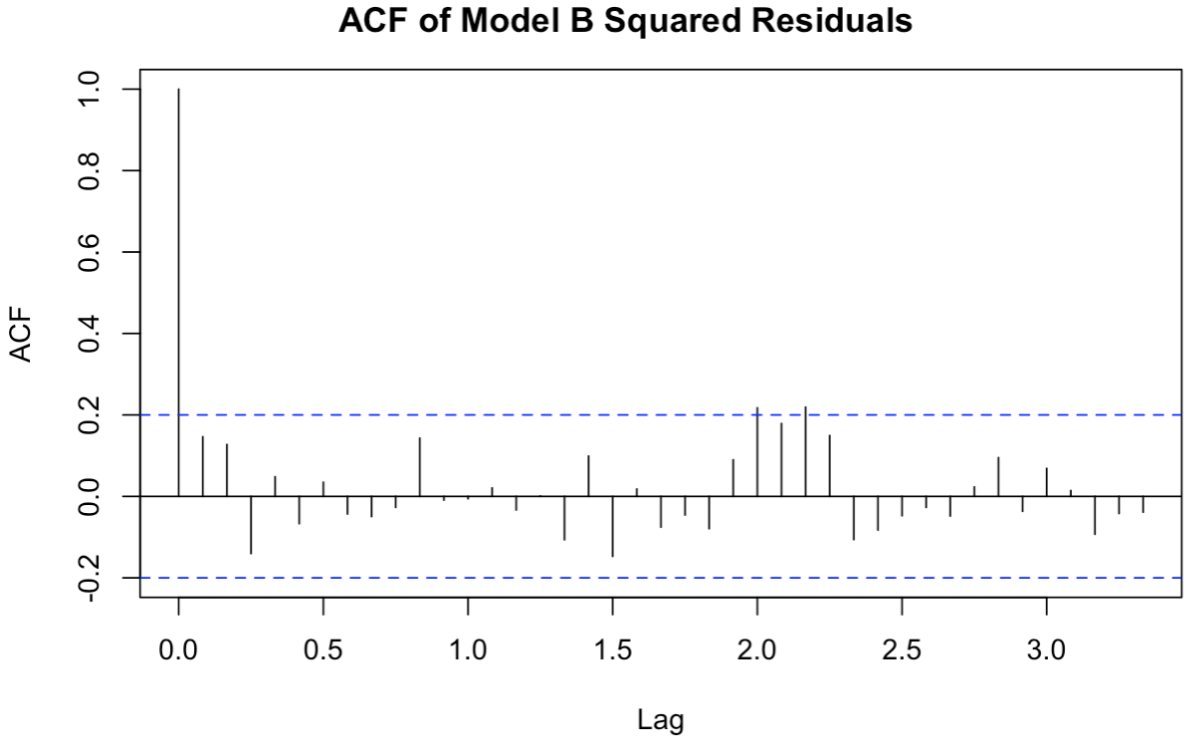


Figure 26: ACF of Model B Squared Residuals

Doing similar diagnostic checking for Model B, we can see even stronger normal distributions. The Box-Pierce test for the residuals res3 has a p-value of 0.5025, and the Ljung-Box test has a p-value of 0.4129, both indicating a lack of significant autocorrelation at various lag lengths. When these tests are applied to the squared residuals, the Ljung-Box test yields a p-value of 0.6698, suggesting that there is no substantial autocorrelation. The histogram of Model B residuals suggests a fairly normal distribution, with small skewness. The QQ plot supports this observation as the points largely follow the theoretical line, with some deviations at the tails which are fairly common. The time series plot of Model B residuals shows no apparent trends or patterns, indicating good model fit. The ACF and PACF of the residuals show spikes within the confidence bounds, which is consistent with white noise behavior. The ACF of the squared residuals also shows there are no significant spikes that would indicate periods of increased variance in the residuals.The application of the Yule-Walker method suggests an AR model of order 0 is appropriate, with a sigma squared estimate of 1.118, indicating no further AR terms are required.The residuals appear to be random, normally distributed without significant autocorrelation or volatility clustering, and the model does not require additional AR terms. The conclusion from these diagnostics is that Model B is even more normal than Model A. We will start forecasting with model B.

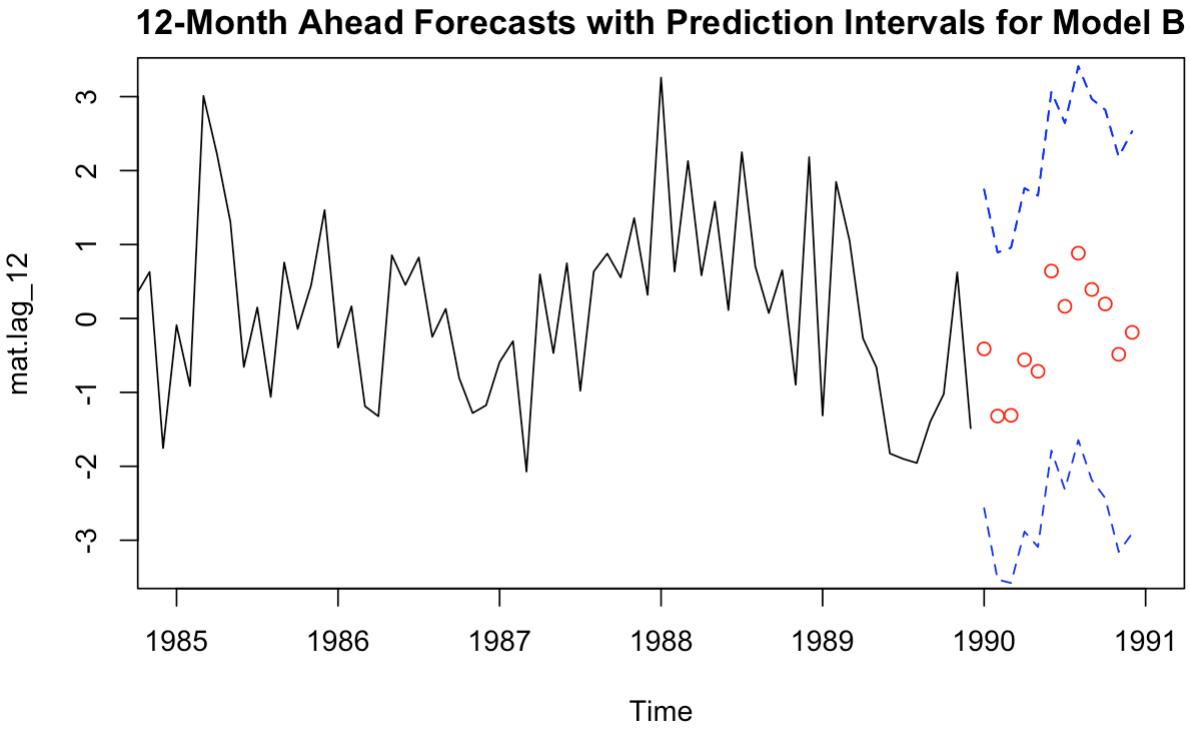


Figure 27: 12-Month Forecast with Prediction Intervals for Model B On Differenced Training Data

A 12 month forecast using model B is depicted above in Figure 27. The time series suggests that the forecasting model for Model B captures the general trend of the data fairly well at first. The actual values stay within the confidence intervals for most of the observed period, indicating that the model's predictions are consistent with the historical data. However, towards the end of the time series, the actual values deviate from the predicted intervals, which may signal potential issues with the model's ability to capture more recent dynamics or may simply reflect natural variability in the data that any model may struggle to predict. The fact that the actual values towards the end of the series fall outside the prediction intervals suggests that while the model may be adequate for a certain period, it may not fully capture the underlying process

of the time series or may be missing some recent changes in the pattern. This could be a sign of model underfitting or overfitting. The model could be too simplistic and fail to capture all of the patterns of the data. The parameters could also be too tuned to the training data, and cannot forecast new data accurately. We will attempt forecasting with Model A.

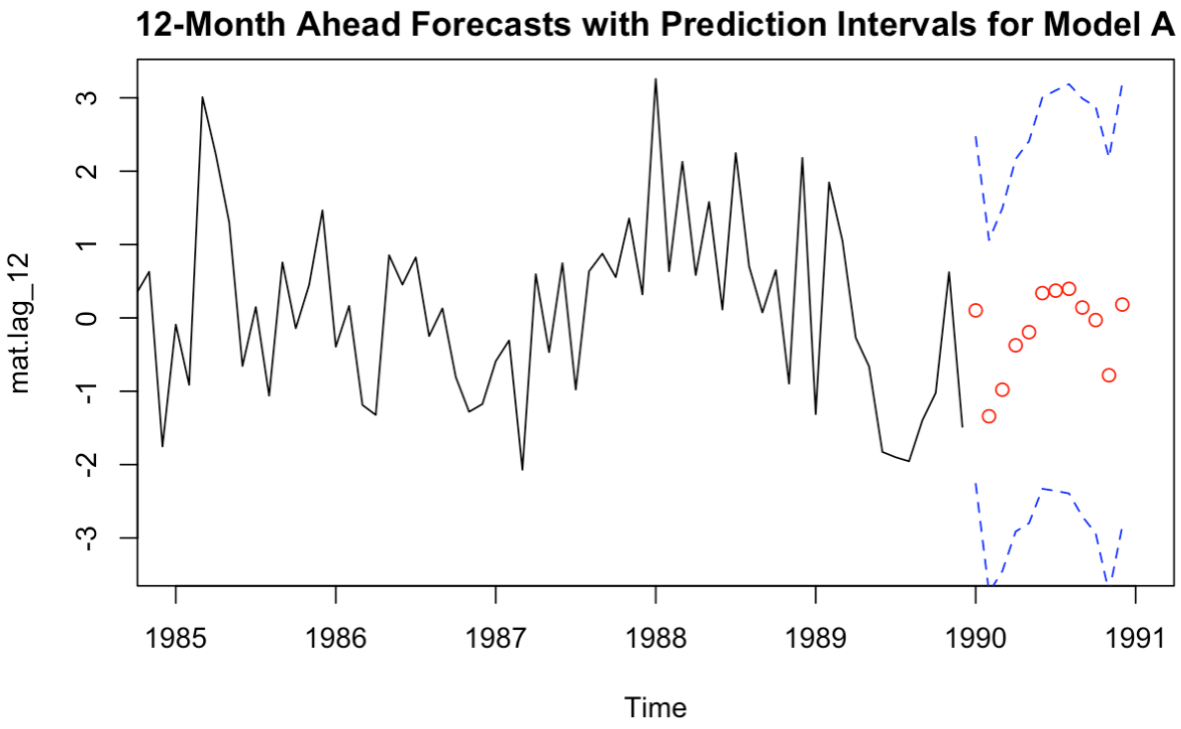


Figure 28: 12-Month Forecast with Prediction Intervals for Model A on Differenced Training Data

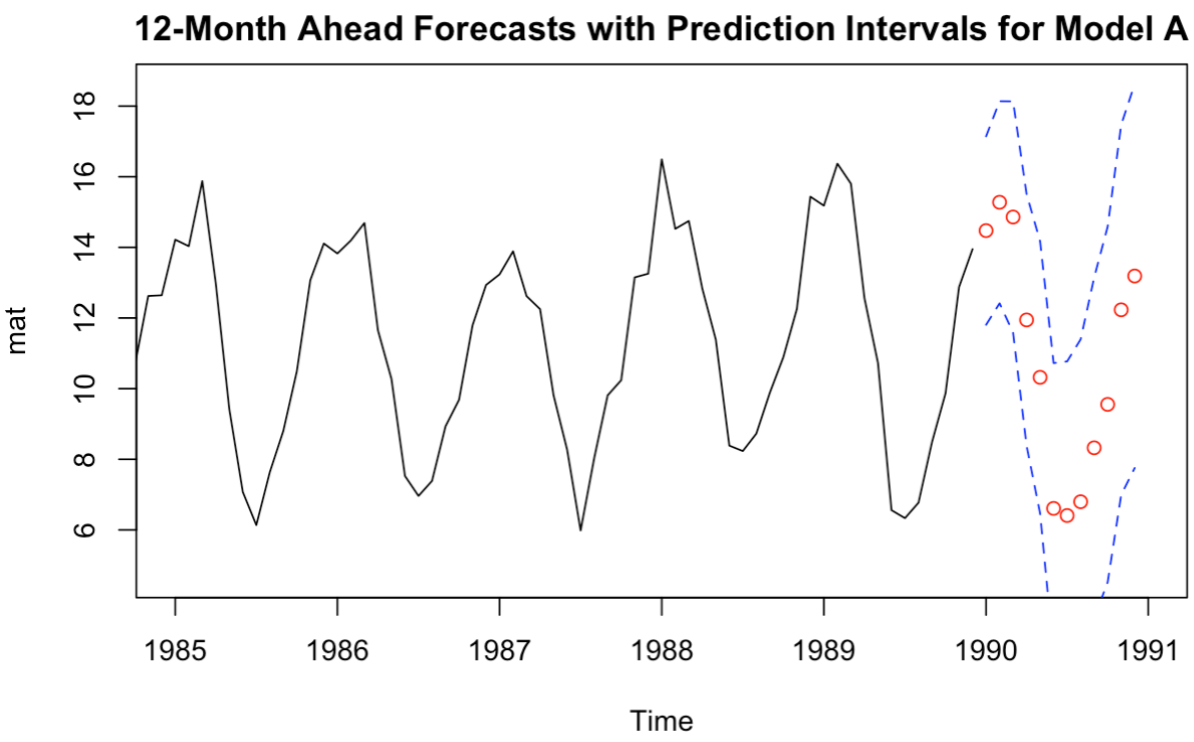


Figure 29: 12-Month Forecast with Prediction Intervals for Model A on Training Data

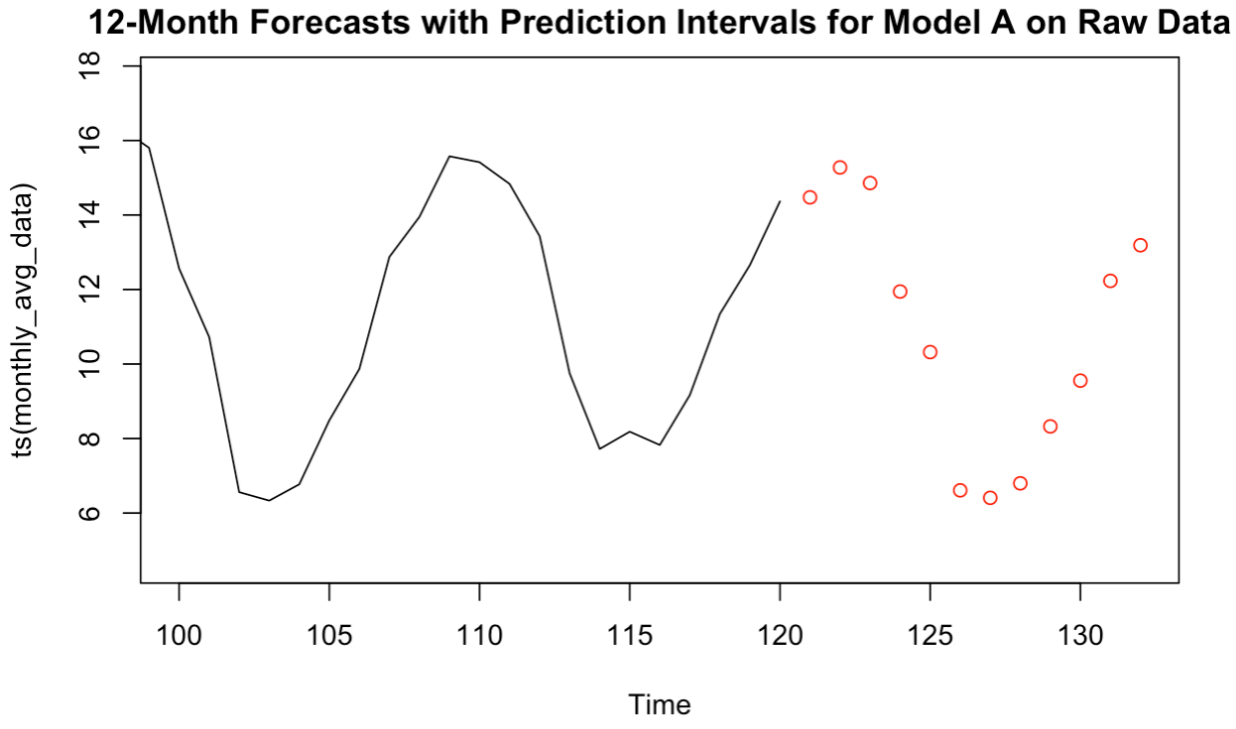


Figure 30: 12-Month Forecast with Prediction Intervals for Model A on Raw Data

The Time Series graphs represented by Figure 28, 29, and 30 indicate that model A was successful at forecasting future values. In Figure 28, the (blue dashed lines) confidence interval margins are superior to model B's error margins. The prediction intervals for Model A are tighter and the forecasted points (red circles) are consistently within these intervals. Model A's predictions are not only closer to the actual values but also the model is more certain about its forecasts. Figures 29 and 30 showcase the forecasting performance of Model A. The forecasted values (red circles) mostly fall within the prediction intervals (blue dashed lines), which suggests that Model A has captured the central tendency and variability of the historical data. The model's ability to forecast within these intervals indicates a strong balance between model complexity and data fitting. The charts also reveal the challenge of data aggregation from daily to monthly entries. While aggregation reduced noise, it also led to a loss of detail. This makes it difficult for the model to capture all of the fine details and patterns of the series. Outliers can distort patterns and trends when aggregated to a monthly average. The variability introduced by outliers eventually affects the model's performance. These outliers could be extreme weather events or storms that are not fully accounted for when forecasting the aggregated data. The impact of this shows the consequences of certain data prepossessing choices done at the early stages of the project. Future modeling efforts may benefit from an initial focus on monthly data to mitigate the noise that daily fluctuations introduce. However, it will be essential to weigh the advantages of noise reduction against the loss of information that daily collected data would provide. These observations support the idea that “All models are wrong, some are useful.”, the famous quote written by British statistician George Box.

Conclusion

This project successfully demonstrates the application of SARIMA modeling in forecasting the minimum daily temperature (in Celsius) in Melbourne, Australia in a dataset spanning from 1981-1990. The initial aggregation of daily data into monthly averages and the use of transformations and differencing addressed the dataset's seasonality and noise. The selection of SARIMA models was first guided by AICc values, ensuring a balance between model simplicity and fit. Diagnostic checks, including normality tests and residual analysis, confirmed the assumptions needed for forecasting. The final forecasting results on 12 months using model SARIMA(1,1,1) x $(1,0,0)_{12}$, while generally consistent with historical data, revealed some limitations in capturing recent trends and the impact of data aggregation choices in data preprocessing. This highlights the trade-off in model accuracy versus simplicity and the consequences of data preprocessing decisions. While this model cannot predict extreme weather events, it can give us insight on seasonal weather patterns and conditions. Overall, the report explores the potential of SARIMA models in weather forecasting and explains the importance of data preprocessing and model selection in time series analysis.

References

Brownlee, J. (2020, December 31). *7 Time Series Datasets for Machine Learning*. MachineLearningMastery.com. https://machinelearningmastery.com/time-series-datasets-for-machine-learning/

Stack Overflow - where developers learn, share, & build careers. (n.d.). Stack Overflow. https://stackoverflow.com/

Appendix

PSTAT 174 Final Project Benjamin Drabeck

Ben Drabeck

2023-12-04

plot.root function definition

```
plot.roots <- function(ar.roots=NULL, ma.roots=NULL, size=2, angles=FALSE, special=NULL, sqecial=NULL,my
{xylims <- c(-size,size)
      omegas <- seq(0,2*pi,pi/500)
      temp <- exp(complex(real=rep(0,length(omegas)),imag=omegas))
      plot(Re(temp),Im(temp),typ="l",xlab="x",ylab="y",xlim=xylims,ylim=xylims,main=main)
      abline(v=0,lty="dotted")
      abline(h=0,lty="dotted")
      if(!is.null(ar.roots))
        {
          points(Re(1/ar.roots),Im(1/ar.roots),col=first.col,pch=my.pch)
          points(Re(ar.roots),Im(ar.roots),col=second.col,pch=my.pch)
        }
      if(!is.null(ma.roots))
        {
          points(Re(1/ma.roots),Im(1/ma.roots),pch="*",cex=1.5,col=first.col)
          points(Re(ma.roots),Im(ma.roots),pch="*",cex=1.5,col=second.col)
        }
      if(angles)
        {
          if(!is.null(ar.roots))
            {
              abline(a=0,b=Im(ar.roots[1])/Re(ar.roots[1]),lty="dotted")
              abline(a=0,b=Im(ar.roots[2])/Re(ar.roots[2]),lty="dotted")
            }
          if(!is.null(ma.roots))
            {
              sapply(1:length(ma.roots), function(j) abline(a=0,b=Im(ma.roots[j])/Re(ma.roots[j]),lty="
            }
        }
      if(!is.null(special))
        {
          lines(Re(special),Im(special),lwd=2)
        }
      if(!is.null(sqecial))
        {
          lines(Re(sqecial),Im(sqecial),lwd=2)
        }
        }
```

```r
#Loading necessary libraries
library(readr)
library(xts)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

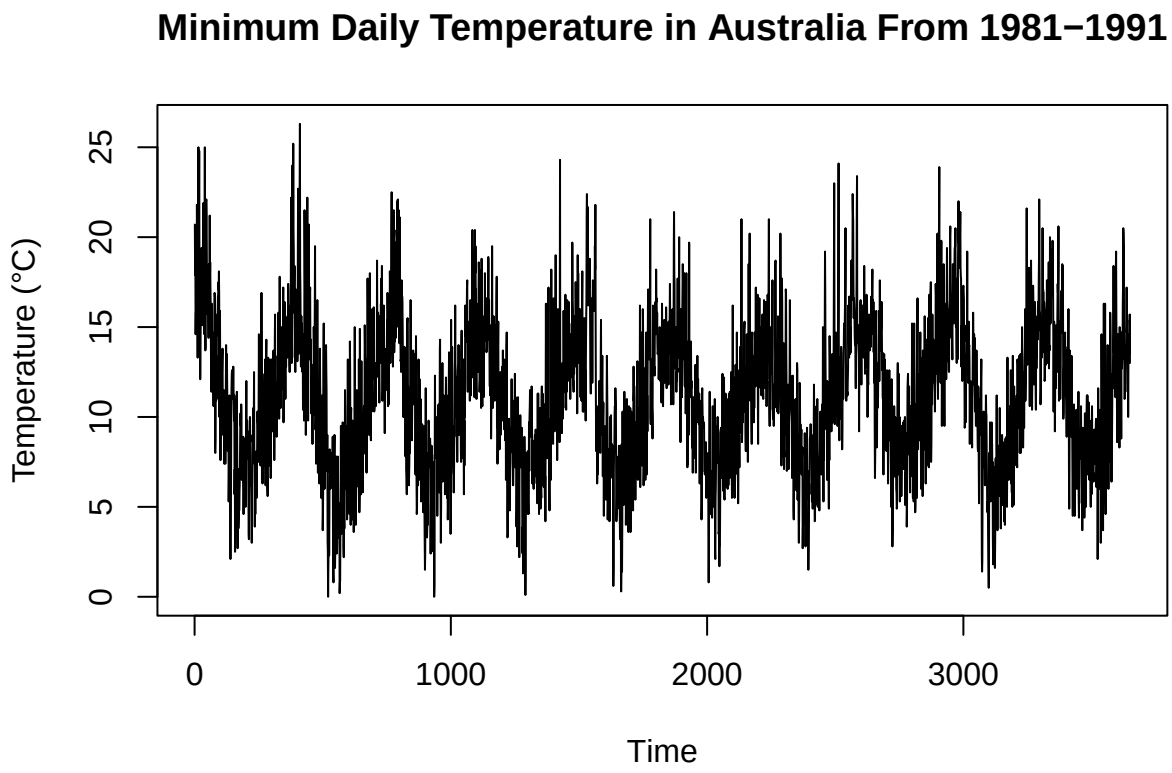
```r
library(MASS)
library(ggplot2)
library(ggfortify)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method                 from
##   autoplot.Arima         ggfortify
##   autoplot.acf           ggfortify
##   autoplot.ar            ggfortify
##   autoplot.bats          ggfortify
##   autoplot.decomposed.ts ggfortify
##   autoplot.ets           ggfortify
##   autoplot.forecast      ggfortify
##   autoplot.stl           ggfortify
##   autoplot.ts            ggfortify
##   fitted.ar              ggfortify
##   fortify.ts             ggfortify
##   residuals.ar           ggfortify
```

```r
library(MuMIn)

#Importing dataset and plotting original time series data
daily_temperatures <- read_csv("https://raw.githubusercontent.com/jbrownlee/Datasets/master/daily-min-t
```

```
## Rows: 3650 Columns: 2

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): Temp
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

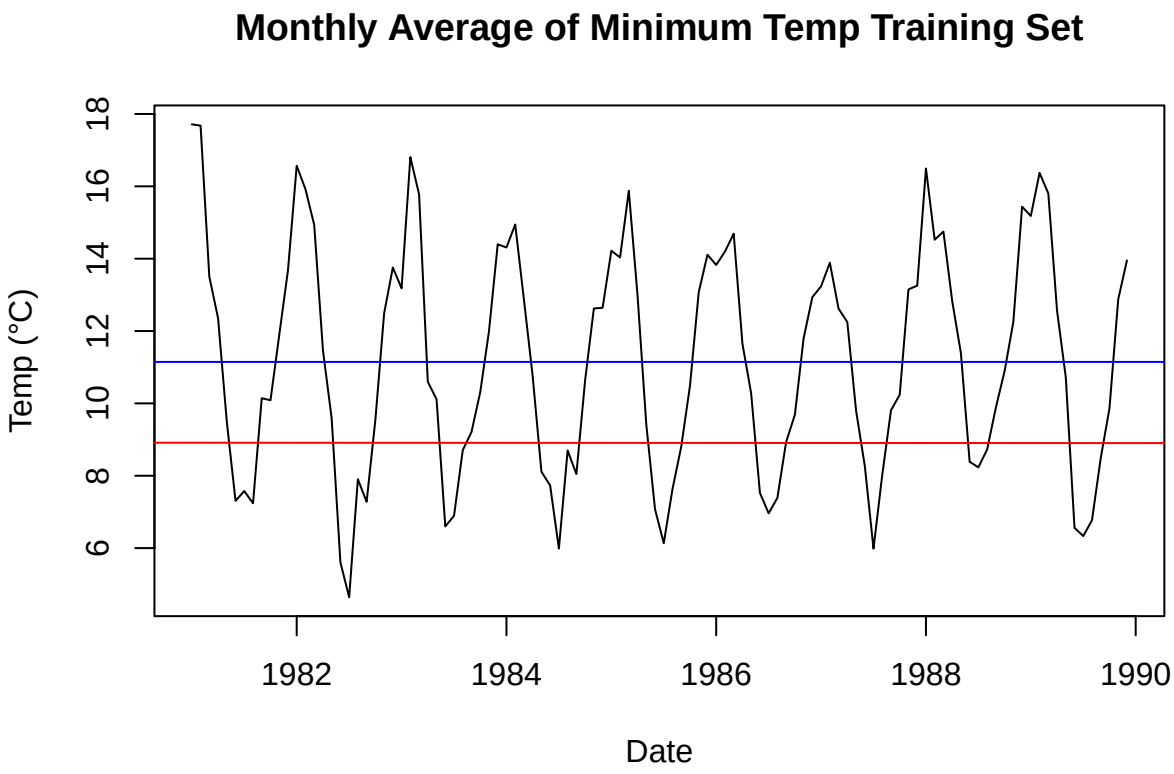
```r
daily_temperatures$Date <- as.Date(daily_temperatures$Date)
temperature_xts <- xts(daily_temperatures$Temp, order.by = daily_temperatures$Date)
plot.ts(temperature_xts, main = "Minimum Daily Temperature in Australia From 1981-1991", ylab = "Tempera
```




```r
#Aggregating by month to reduce noise
monthly_avg_data <- apply.monthly(temperature_xts, FUN = mean)
length(monthly_avg_data)
```

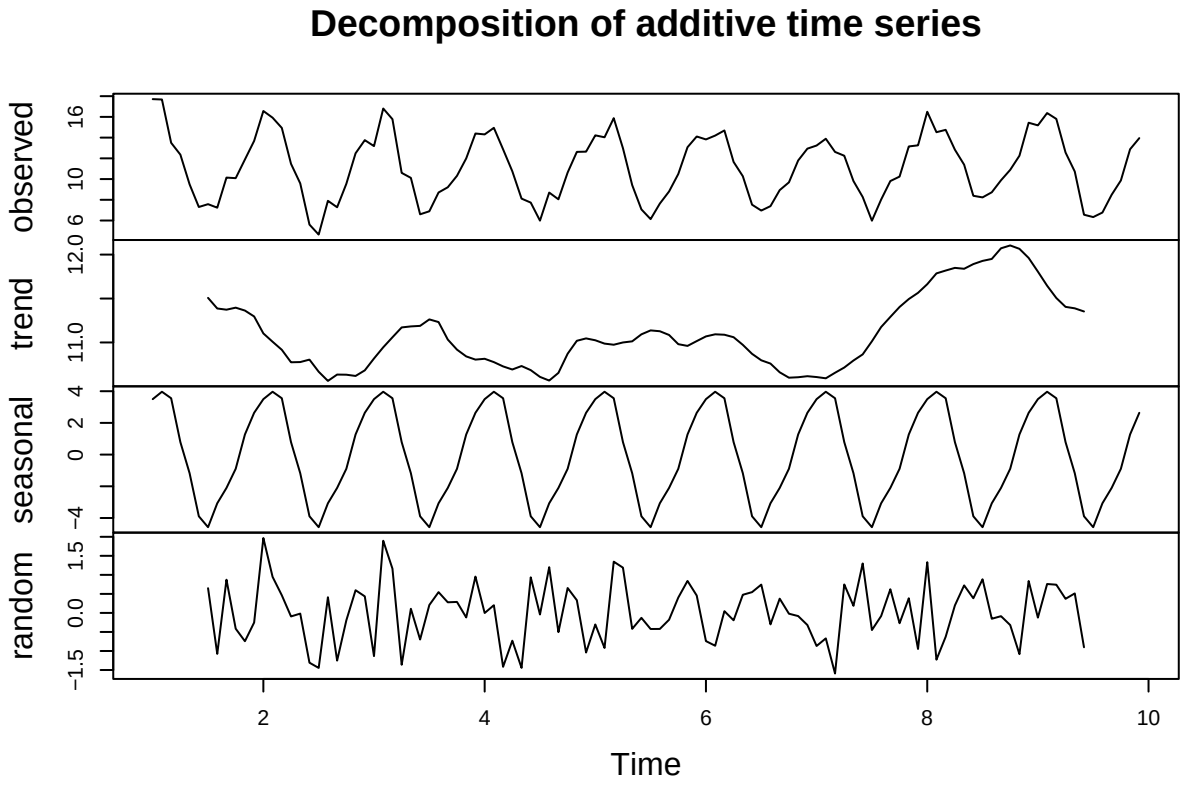
```
## [1] 120
```

```r
#Splitting into training and testing datasets, will use last year for forecasting
mat_xts <- monthly_avg_data[c(1:108)]
mate_xts <- monthly_avg_data[c(109:120)]

#Converting back to ts object for simplicity
start_year <- format(index(monthly_avg_data)[1], "%Y")
start_month <- format(index(monthly_avg_data)[1], "%m")
mat <- ts(coredata(mat_xts), start = c(as.numeric(start_year), as.numeric(start_month)), frequency = 12
mate <- ts(coredata(mate_xts), start = c(as.numeric(start_year) + 9, as.numeric(start_month)), frequency

#Plotting aggregated data
plot(mat, xlab = "Date", ylab = "Temp (°C)", main = "Monthly Average of Minimum Temp Training Set", type
fit_mat <- lm(mat ~ as.numeric(1:length(mat))); abline(fit_mat, col="red")
abline(h=mean(mat, na.rm = TRUE), col="blue")
```



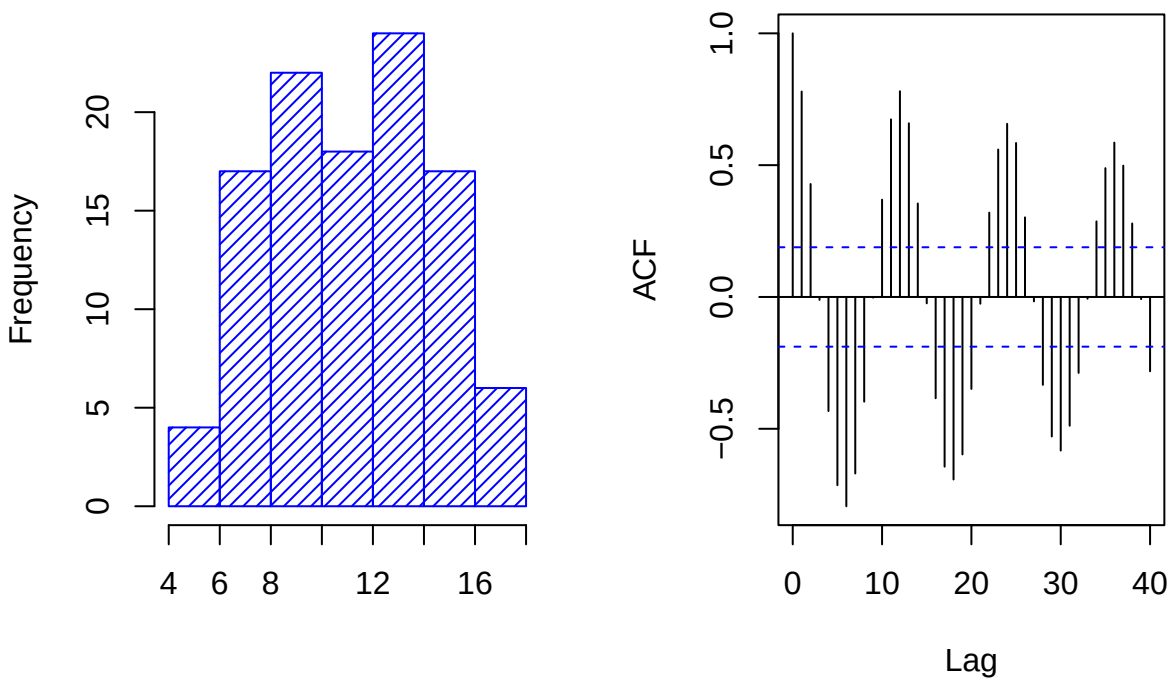
Decomposition

```
x<-ts(as.ts(mat), frequency=12)
decomp<-decompose(x)
plot(decomp)
```



Histogram and ACF Analysis

```
op = par(mfrow = c(1,2))
#slight left skew
hist(mat, density = 20, col="blue", xlab = "",main = "")
acf(coredata(mat), lag.max=40, main = "")
title("Monthly Avg Minimum Temp Histogram & ACF", line = -1, outer = TRUE)
```



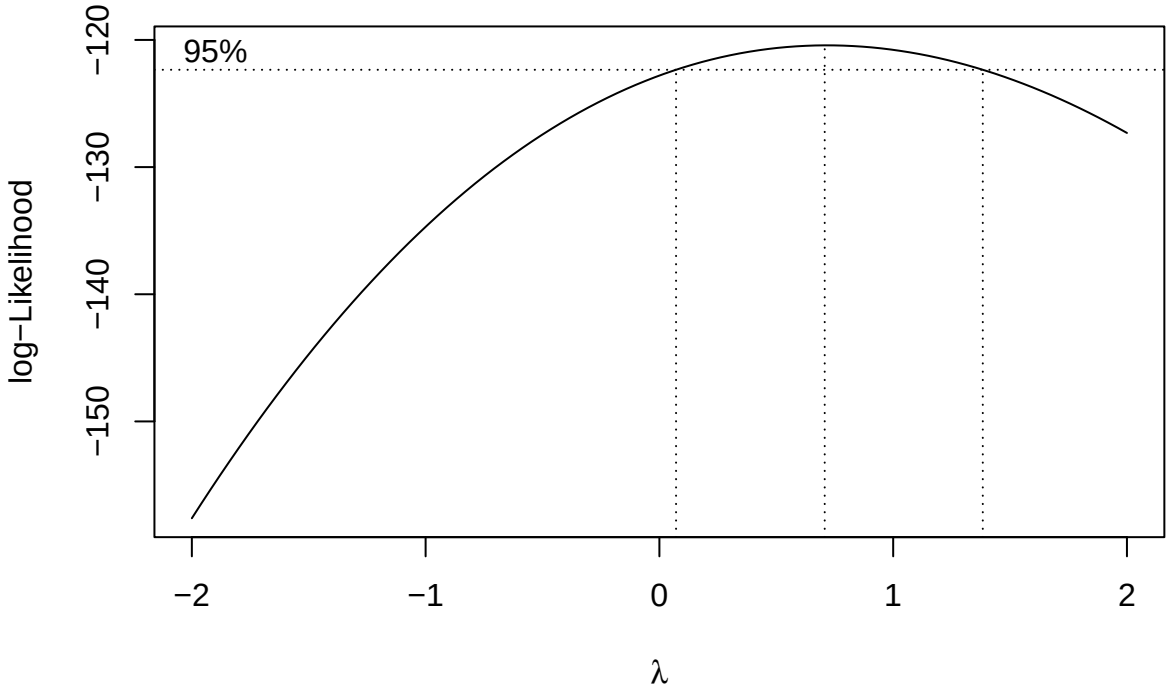

```
#9.97
var(mat)
```

```
##          Series 1
## Series 1 9.977766
```

```
#gradual decline, strong seasonality
```

Box Cox Comparison

```
#Transformation
bcTransform <- boxcox(mat~as.numeric(1:length(mat)))
```

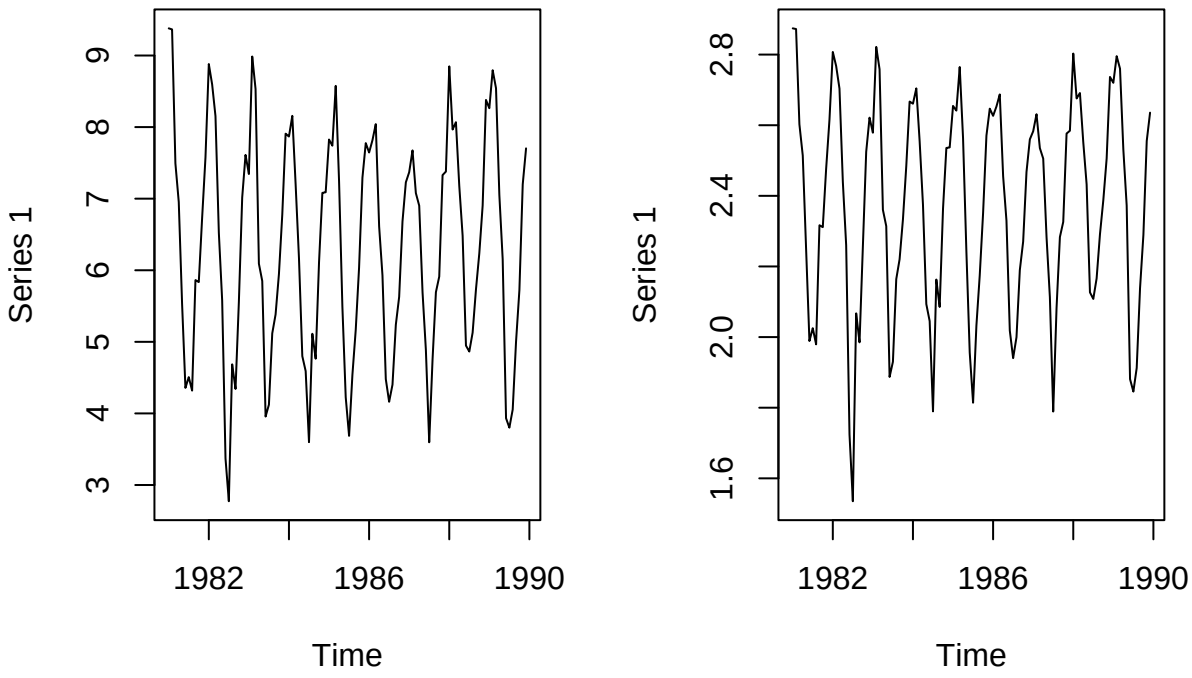



```
op = par(mfrow = c(1,2))

#Find lambda and assign variable
bcTransform$x[which(bcTransform$y==max(bcTransform$y))]
```

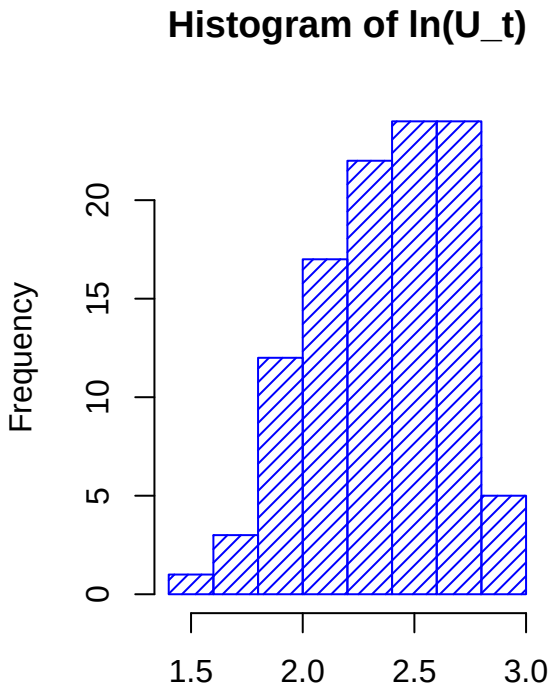
```
## [1] 0.7070707
```

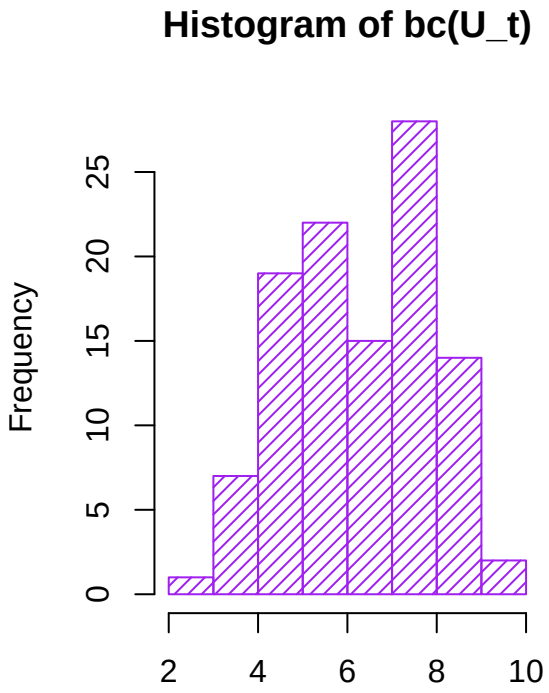
```
lambda<-bcTransform$x[which(bcTransform$y==max(bcTransform$y))]
mat.bc = (1/lambda)*(mat^lambda-1)
mat.log<-log(mat)

#Plotting for Analysis
plot.ts(mat.bc)
plot.ts(mat.log)
title("Time Series of Box Cox (left) and Logged Data (right)", line = -1, outer = TRUE)
```

Time Series of Box Cox (left) and Logged Data (right)




```
hist(mat.log, density = 20, col="blue", xlab="", main="Histogram of ln(U_t)")
hist(mat.bc, density = 20, col="purple", xlab="", main="Histogram of bc(U_t)")
```






```
var(mat.log)
```

```
##            Series 1
## Series 1 0.09084297
```

```
var(mat.bc)
```

```
##          Series 1
## Series 1 2.475097
```

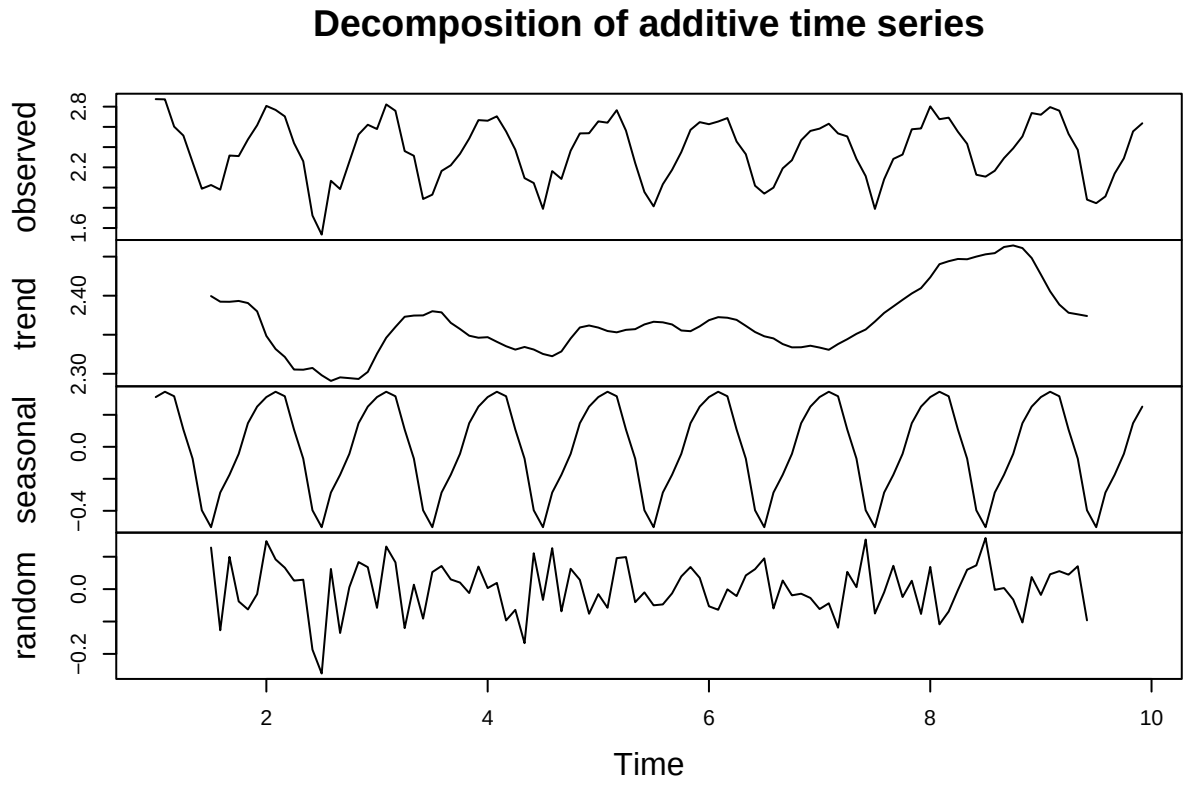
```
lambda
```

```
## [1] 0.7070707
```

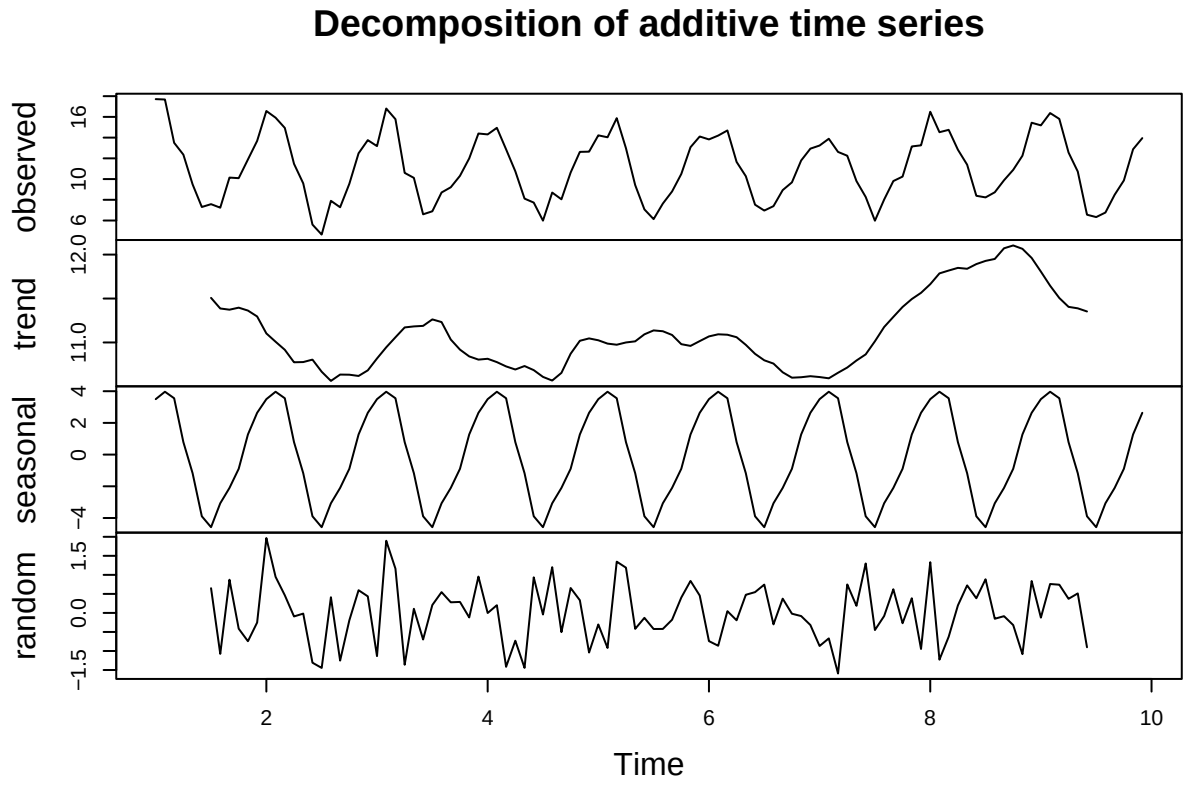
```
#histogram of ln is skewed to the right and does not have bell shaped curve
#histogram of box cox appears more symmetric but with two peaks and not bell shaped
#1 is inside our confidence interval so we will not used our transformed data.
```

Decomposition of logged & training data Comparison

```
#Decomp and plot log
y <- ts(as.ts(mat.log), frequency=12)
decompy <- decompose(y)
plot(decompy)
```



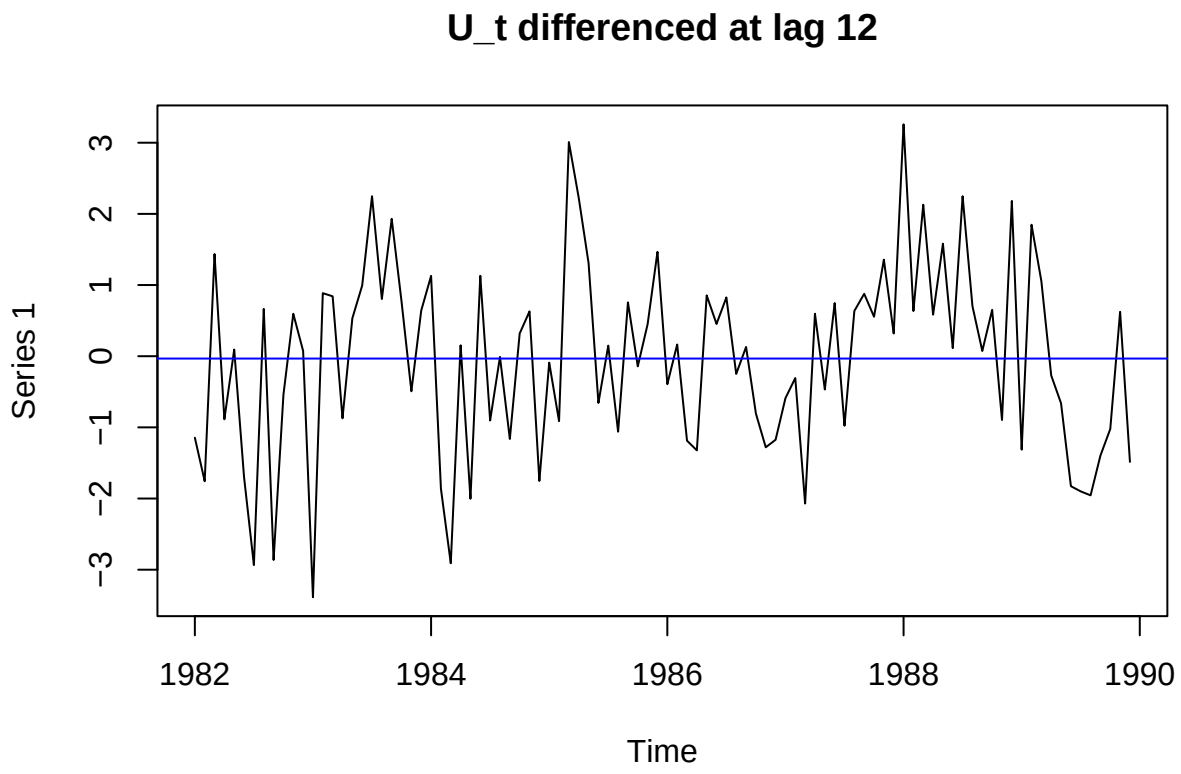

```
#Decomp and plot training data
x<-ts(as.ts(mat), frequency=12)
decompx<-decompose(x)
plot(decompx)
```




```
#Conclusion: Given the observed stability, linear trend, and constant seasonal pattern, a log transform
```

Difference at lag 12 to address seasonality

```
mat.lag_12 <- diff(mat,lag=12)
plot.ts(mat.lag_12, main="U_t differenced at lag 12")
fit_lag_12 <- lm(na.omit(mat.lag_12) ~ as.numeric(1:length(na.omit(mat.lag_12))))
abline(fit_lag_12, col="red", lwd=6)
abline(h=mean(mat.lag_12, na.rm = TRUE), col="blue")
```



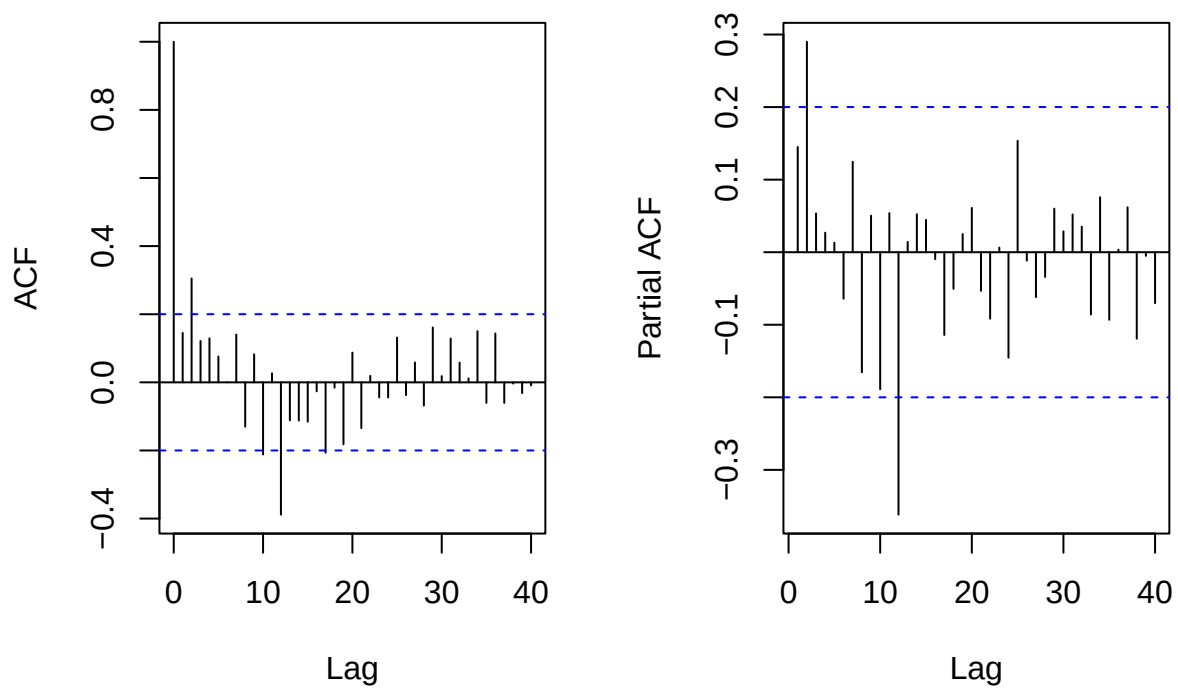

```
summary(fit_lag_12)
```

```
##
## Call:
## lm(formula = na.omit(mat.lag_12) ~ as.numeric(1:length(na.omit(mat.lag_12))))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1769 -1.0133  0.1994  0.8362  3.1688
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             -0.275066   0.275976  -0.997    0.321
## as.numeric(1:length(na.omit(mat.lag_12)))  0.004990   0.004941   1.010    0.315
##
## Residual standard error: 1.341 on 94 degrees of freedom
## Multiple R-squared:  0.01074,    Adjusted R-squared:  0.0002123
## F-statistic:  1.02 on 1 and 94 DF,  p-value: 0.3151
```

```
# mean at 0, trend not statistically significant
```

ACF/PACF Analysis

```
op = par(mfrow = c(1,2))
acf(coredata(mat.lag_12), lag.max=40, main = "")
pacf(coredata(mat.lag_12), lag.max=40, main = "")
title("ACF/PACF of the Differenced Monthly Avg Minimum Temperature Data", line = -1, outer = TRUE)
```

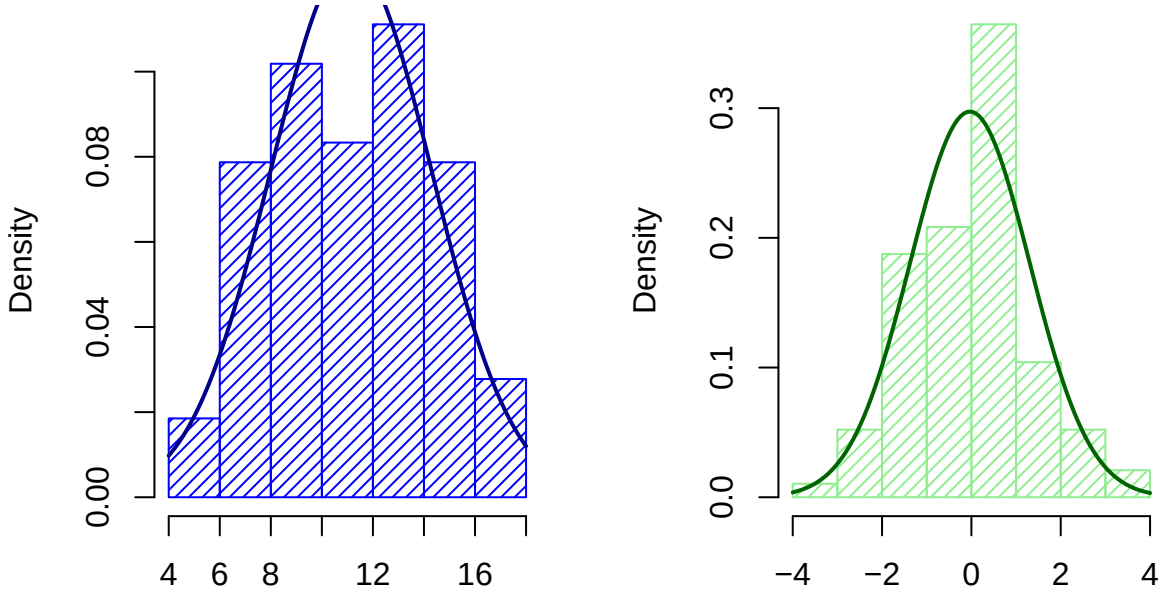



```
par(op)
```

Histogram Comparison

```
#Original data
op = par(mfrow = c(1,2))
hist(mat, density=20, col="blue", xlab="", main="", prob=TRUE)
curve(dnorm(x, mean=mean(mat, na.rm=TRUE), sd=sd(mat, na.rm=TRUE)), col="dark blue", lwd=2, add=TRUE)

#Differenced data
hist(mat.lag_12, density=20, col="light green", xlab="", main="", prob=TRUE)
curve(dnorm(x, mean=mean(mat.lag_12, na.rm=TRUE), sd=sd(mat.lag_12, na.rm=TRUE)), col="dark green", lwd

title("Histogram of Original Training Data & Training Data Differenced at Lag 12", line = -1, outer = T
```

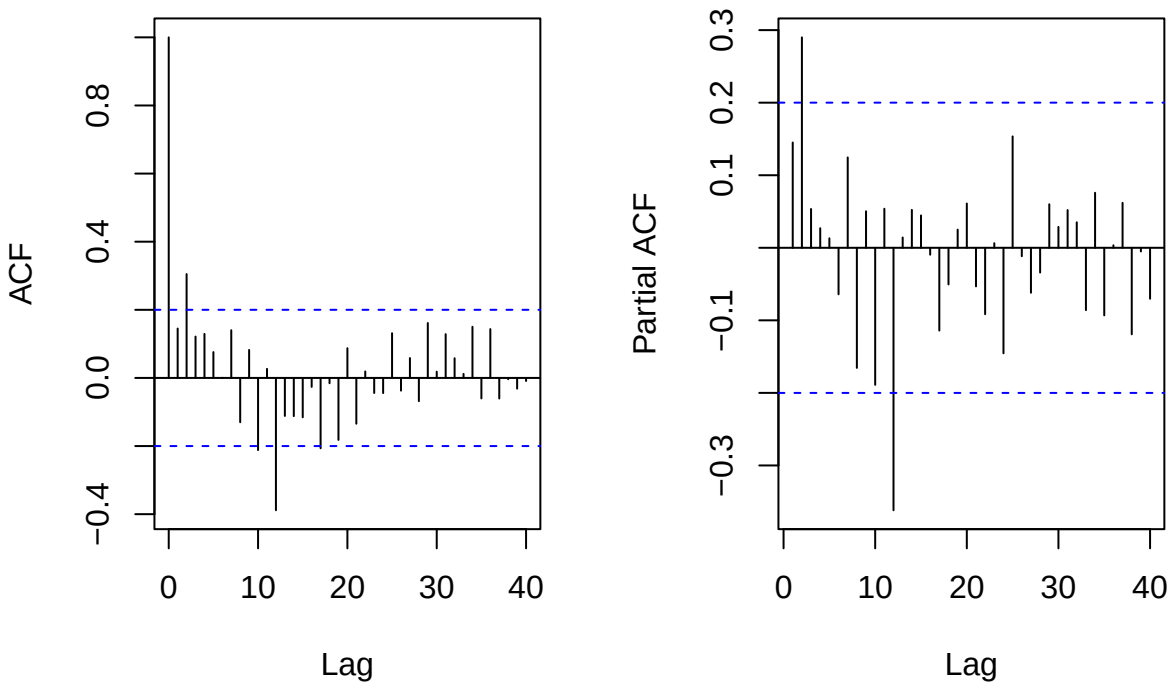
istogram of Original Training Data & Training Data Differenced at Lag 1



MODEL SELECTION

```
#Plotting ACF/PACF of the differenced data
op = par(mfrow = c(1,2))
acf(coredata(mat.lag_12), lag.max=40, main = "")
pacf(coredata(mat.lag_12), lag.max=40, main = "")
title("ACF/PACF of the Differenced Monthly Avg Minimum Temperature Data", line = -1, outer = TRUE)
```

ACF/PACF of the Differenced Monthly Avg Minimum Temperature Data




```
par(op)
```

ACF Outside Confidence Intervals: 2, 9,11,16 PACF Outside Confidence Intervals: 1,11 s = 12 D = 1 d = 1 Q = 1 q = 1 or 2 P = 1 p = 1

Candidate Models:

```
# Model with SARIMA(1,1,1)(1,0,0)[12]
fit1 <- Arima(mat.lag_12, order=c(1,1,1), seasonal=list(order=c(1,0,0), period=12), method = "ML")
fit1
```

```
## Series: mat.lag_12
## ARIMA(1,1,1)(1,0,0)[12]
##
## Coefficients:
##           ar1      ma1     sar1
##       -0.0887  -0.7338  -0.4570
## s.e.   0.1687   0.1451   0.0977
##
## sigma^2 = 1.443:  log likelihood = -152.55
## AIC=313.1   AICc=313.55   BIC=323.32
```

```
#Fixing ar1 to 0 due to insignificance
fit1_fixed <- Arima(mat.lag_12, order=c(1,1,1), seasonal=list(order=c(1,0,0), period=12), fixed = c(0, N
fit1_fixed
```

```
## Series: mat.lag_12
## ARIMA(1,1,1)(1,0,0)[12]
##
## Coefficients:
##       ar1      ma1     sar1
##         0  -0.7915  -0.4653
## s.e.    0   0.0832   0.0964
##
## sigma^2 = 1.428:  log likelihood = -152.69
## AIC=311.37   AICc=311.64   BIC=319.04
```

```
# Model with SARIMA(1,1,2)(1,1,1)[12]
fit2 <- Arima(mat.lag_12, order=c(1,1,2), seasonal=list(order=c(1,1,1), period=12), method = "ML")
fit2
```

```
## Series: mat.lag_12
## ARIMA(1,1,2)(1,1,1)[12]
##
## Coefficients:
##           ar1      ma1      ma2     sar1     sma1
##       -0.5188  -0.3176  -0.2845  -0.3626  -0.9997
## s.e.   0.3911   0.4047   0.3293   0.1122   0.3098
##
## sigma^2 = 1.683:  log likelihood = -153.8
## AIC=319.6   AICc=320.71   BIC=334.12
```

```
# Try a simpler model, SARIMA(1,1,1)(0,0,1)[12]
fit3 <- Arima(mat.lag_12, order=c(1,1,1), seasonal=list(order=c(0,0,1), period=12), method = "ML")
fit3
```

```
## Series: mat.lag_12
## ARIMA(1,1,1)(0,0,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.0713  -0.8104  -0.8112
## s.e.  0.1702   0.1261   0.1757
##
## sigma^2 = 1.147:  log likelihood = -146.62
## AIC=301.24   AICc=301.69   BIC=311.46
```

```
#Fixing ar1 to 0 due to insignificance
fit3_fixed <- Arima(mat.lag_12, order=c(1,1,1), seasonal=list(order=c(0,0,1), period=12), fixed = c(0, N
fit3_fixed
```

```
## Series: mat.lag_12
## ARIMA(1,1,1)(0,0,1)[12]
##
## Coefficients:
##       ar1      ma1     sma1
##         0  -0.7678  -0.7848
## s.e.    0   0.0897   0.1498
```

```
##
## sigma^2 = 1.153:  log likelihood = -146.71
## AIC=299.42   AICc=299.68   BIC=307.08
```

```
# Try increasing complexity, SARIMA(2,1,1)(2,1,1)[12]
fit4 <- Arima(mat.lag_12, order=c(2,1,1), seasonal=list(order=c(2,1,1), period=12), method = "ML")
fit4
```

```
## Series: mat.lag_12
## ARIMA(2,1,1)(2,1,1)[12]
##
## Coefficients:
##          ar1     ar2      ma1     sar1     sar2     sma1
##       0.1568  0.2078  -1.0000  -0.8123  -0.5187  -0.4601
## s.e.  0.1118  0.1130   0.1613   0.1615   0.1406   0.2214
##
## sigma^2 = 1.599:  log likelihood = -149.07
## AIC=312.13   AICc=313.63   BIC=329.06
```

```
# Simpler non-seasonal model, ARIMA(0,1,0)
fit5 <- Arima(mat.lag_12, order=c(0,1,0), method = "ML")
fit5
```

```
## Series: mat.lag_12
## ARIMA(0,1,0)
##
## sigma^2 = 3.042:  log likelihood = -187.65
## AIC=377.29   AICc=377.34   BIC=379.85
```

```
# More complex non-seasonal model, ARIMA(2,1,2)
fit6 <- Arima(mat.lag_12, order=c(2,1,2), method = "ML")
fit6
```

```
## Series: mat.lag_12
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1     ar2      ma1     ma2
##       0.3343  0.2726  -1.2374  0.2374
## s.e.  0.3397  0.1268   0.3541  0.3521
##
## sigma^2 = 1.673:  log likelihood = -158.97
## AIC=327.94   AICc=328.61   BIC=340.71
```

```
# Simpler seasonal model, SARIMA(1,1,1)(0,1,0)[12]
fit7 <- Arima(mat.lag_12, order=c(1,1,1), seasonal=list(order=c(0,1,0), period=12), method = "ML")
fit7
```

```
## Series: mat.lag_12
## ARIMA(1,1,1)(0,1,0)[12]
##
## Coefficients:
```

```
##           ar1      ma1
##       -0.2028  -0.6437
## s.e.   0.1732   0.1575
##
## sigma^2 = 5.091:  log likelihood = -184.71
## AIC=375.43   AICc=375.73   BIC=382.68
```

```
# Model without MA terms, ARIMA(2,1,0)
fit8 <- Arima(mat.lag_12, order=c(2,1,0), method = "ML")
fit8
```

```
## Series: mat.lag_12
## ARIMA(2,1,0)
##
## Coefficients:
##           ar1      ar2
##       -0.7276  -0.2236
## s.e.   0.1003   0.1020
##
## sigma^2 = 1.902:  log likelihood = -164.58
## AIC=335.17   AICc=335.43   BIC=342.83
```

AICc Comparison

```
AICc(fit1_fixed)
```

```
## [1] 311.638
```

```
AICc(fit2)
```

```
## [1] 320.7089
```

```
AICc(fit3_fixed)
```

```
## [1] 299.6797
```

```
AICc(fit4)
```

```
## [1] 313.6263
```

```
AICc(fit5)
```

```
## [1] 377.3358
```

```
AICc(fit6)
```

```
## [1] 328.6143
```

```
AICc(fit7)
```

```
## [1] 375.7304
```

```
AICc(fit8)
```

```
## [1] 335.4335
```

```
#Conclusion: We will use ARIMA(1,1,1)(1,0,0)[12] and ARIMA(1,1,1)(0,0,1)[12]  for forecasting
```

Series: mat.lag_12 ARIMA(1,1,1)(1,0,0)[12]

Coefficients: ar1 ma1 sar1 0 -0.7915 -0.4653 s.e. 0 0.0832 0.0964

sigma^2 = 1.428: log likelihood = -152.69 AIC=311.37 AICc=311.64 BIC=319.04

Series: mat.lag_12 ARIMA(1,1,1)(0,0,1)[12]

Coefficients: ar1 ma1 sma1 0 -0.7678 -0.7848 s.e. 0 0.0897 0.1498

sigma^2 = 1.153: log likelihood = -146.71 AIC=299.42 AICc=299.68 BIC=307.08

Model A (fit1_fixed): Sarima(1,1,1)(1,0,0)[12]

$$(1 + 0.4653_{(0.0964)}B^{12})X_t = (1 - 0.7915_{(0.0832)}B)Z_t$$

$$\hat{\sigma}_Z^2 = 1.428$$

Model B (fit3_fixed): Sarima(1,1,1)(0,0,1)[12]

$$(1 - 0.7678_{(0.0897)}B)(1 - 0.7848_{(0.1498)}B^{12})Z_t$$

$$\hat{\sigma}_Z^2 = 1.153$$

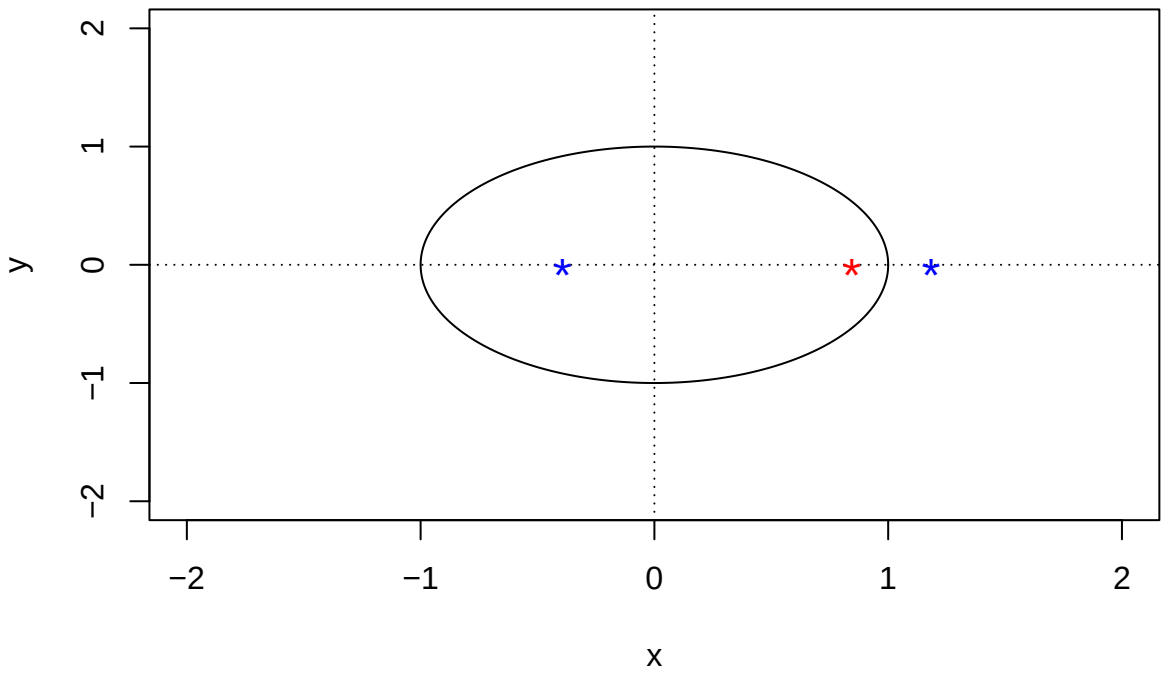
*Checking stationarity/invertibility:

Model A: • SAR part: 0.4653 is smaller than 1. So the roots lie outside the unit circle. • AR part: no AR part • SMA part: no SMA part • MA part: 0.7915 is smaller than 1. So the roots lie outside the unit circle

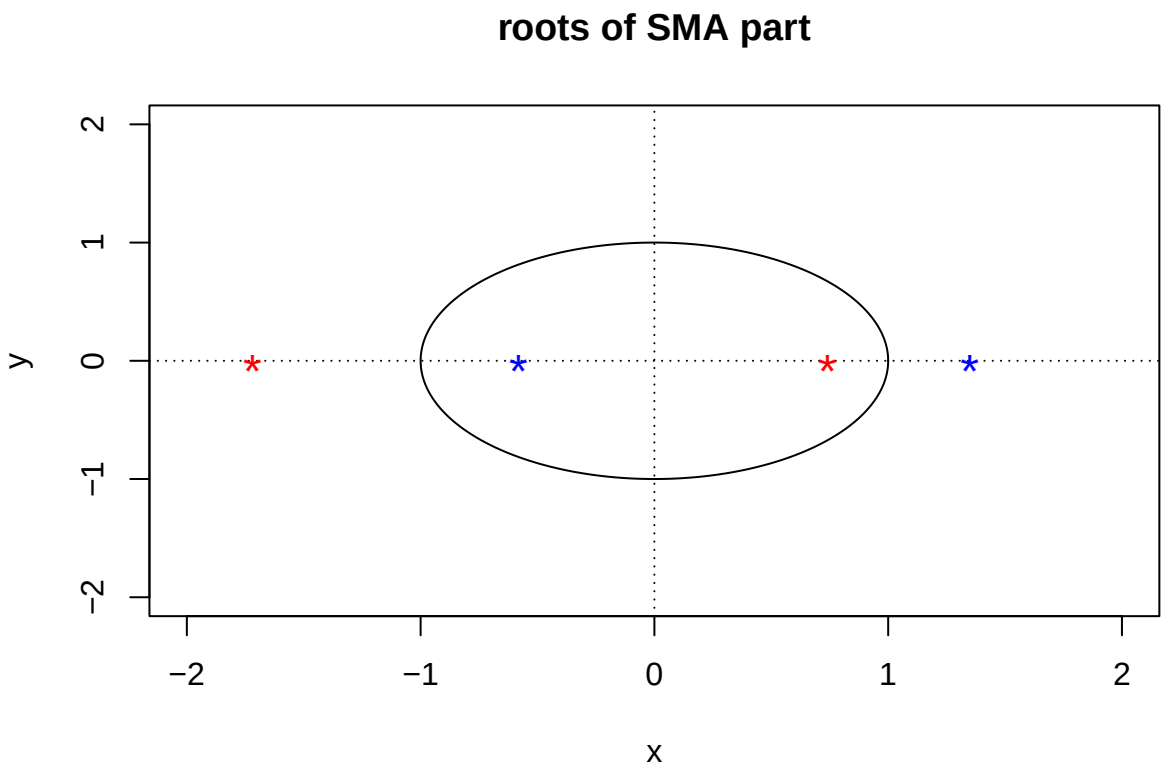
Model B: • SAR part: no SAR part • AR part: no AR part • SMA part: 0.7648 is smaller than 1. So the roots lie outside the unit circle • MA part: 0.7678 is smaller than 1. So the roots lie outside the unit circle

Poly Root Graphs

```
#Plotting Model A roots
library(forecast)
plot.roots(NULL, polyroot(c(1, -0.7915, -0.4653)), main = "roots of SAR part")
```




```
#Plotting Model B roots
plot.roots(NULL, polyroot(c(1, -0.7678, -0.7848)), main = "roots of SMA part")
```

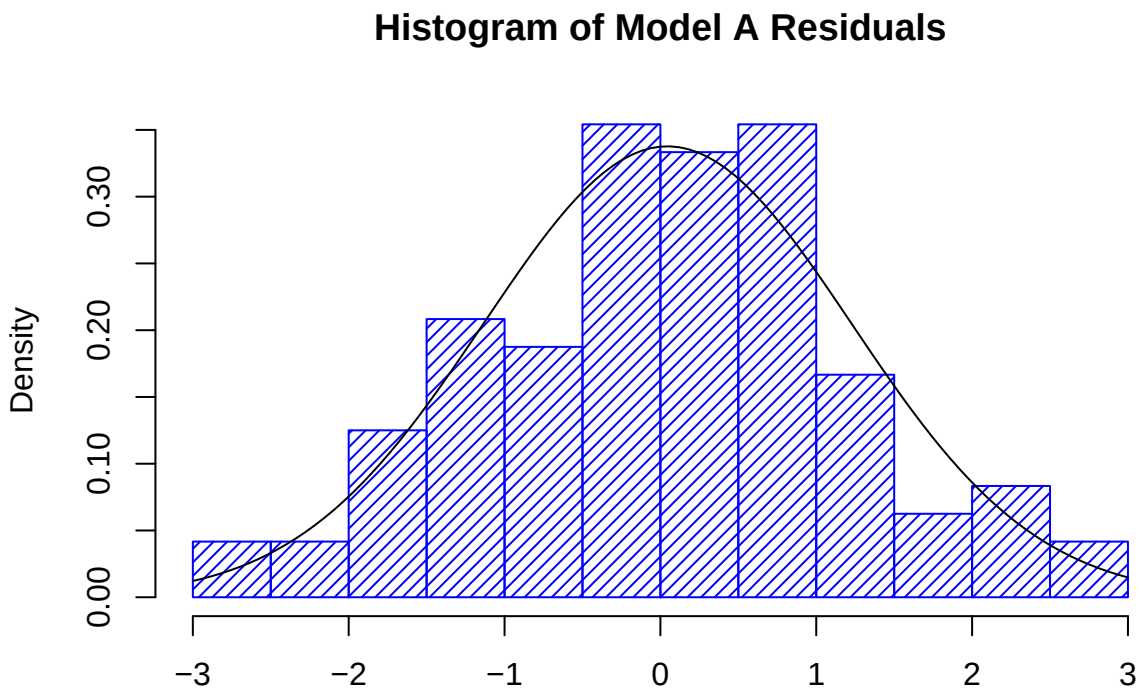


DIAGNOSTIC CHECKING FOR MODEL A

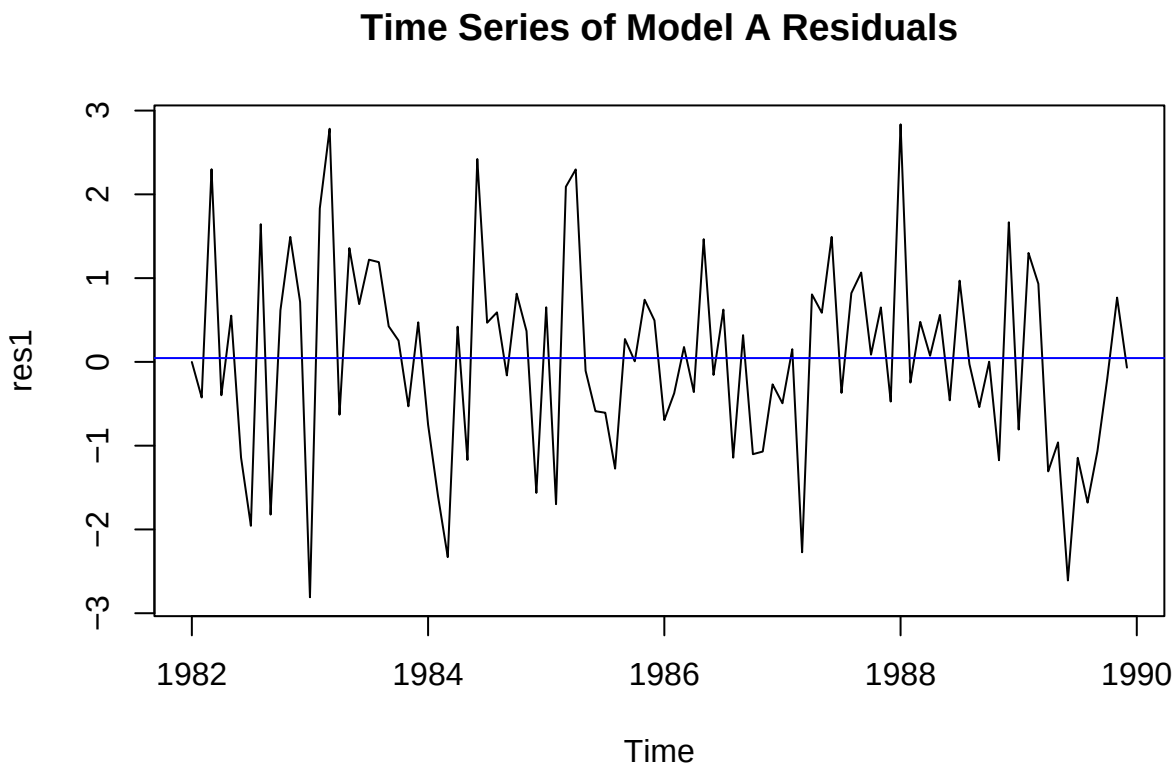
```
#Analyzing normal distribution of residuals
res1 <- residuals(fit1_fixed)
shapiro.test(res1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res1
## W = 0.99278, p-value = 0.8889
```

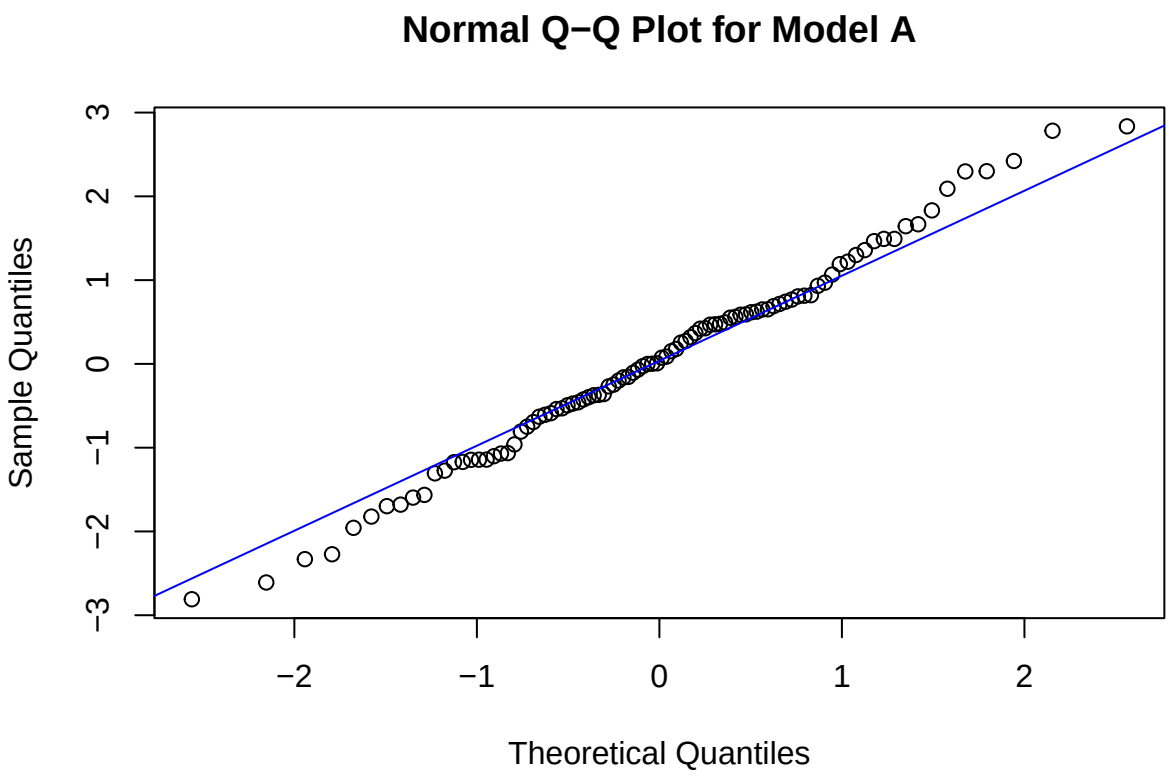
```
hist(res1, density=20,breaks=20, col="blue", xlab="", main = "Histogram of Model A Residuals", prob=TRU
m1 <- mean(res1)
std1 <- sqrt(var(res1))
curve(dnorm(x,m1,std1), add=TRUE )
```



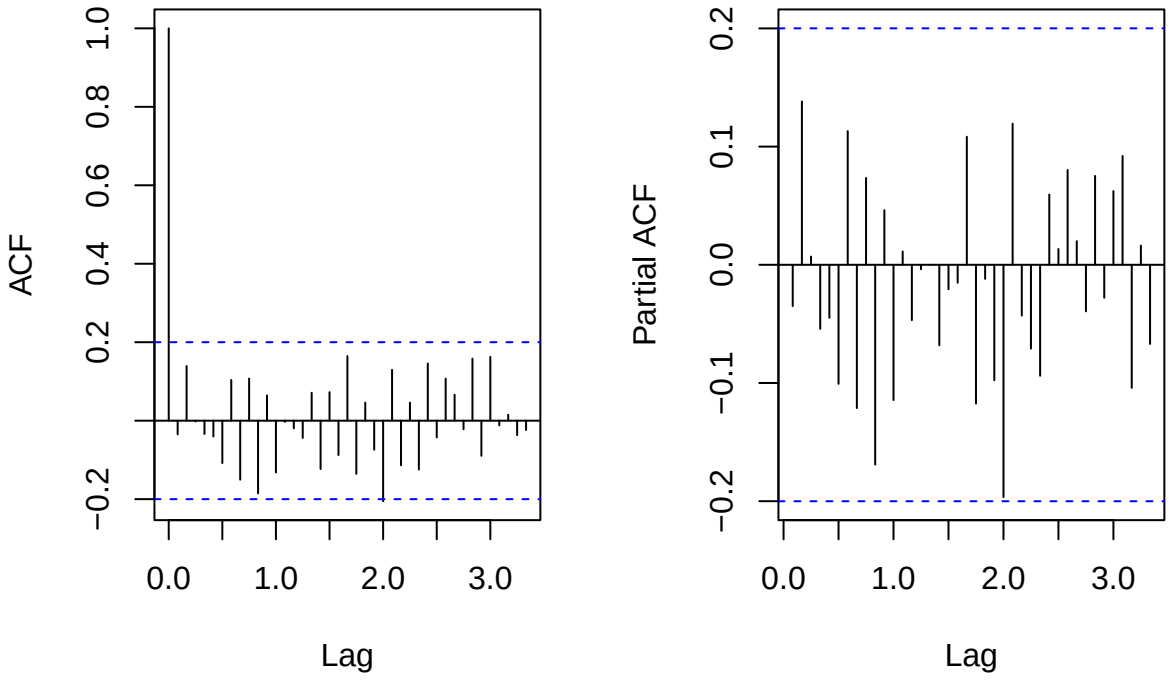

```
#Plotting residuals as time series, qqplot, and acf/pacf
plot.ts(res1, main="Time Series of Model A Residuals")
fitt1 <- lm(res1 ~ as.numeric(1:length(res1))); abline(fitt1, col="red")
abline(h=mean(res1), col="blue")
```




```
qqnorm(res1,main= "Normal Q-Q Plot for Model A")
qqline(res1,col="blue")
```




```
#ACF/PACF
op = par(mfrow = c(1,2))
acf(res1, lag.max=40, main="")
pacf(res1, lag.max=40, main="")
title("ACF & PACF of Model A Residuals", line = -1, outer = TRUE)
```




```
par(op)

#Box Tests
Box.test(res1, lag = 12, type = c("Box-Pierce"), fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res1
## X-squared = 13.064, df = 10, p-value = 0.2201
```

```
Box.test(res1, lag = 12, type = c("Ljung-Box"), fitdf = 2)
```

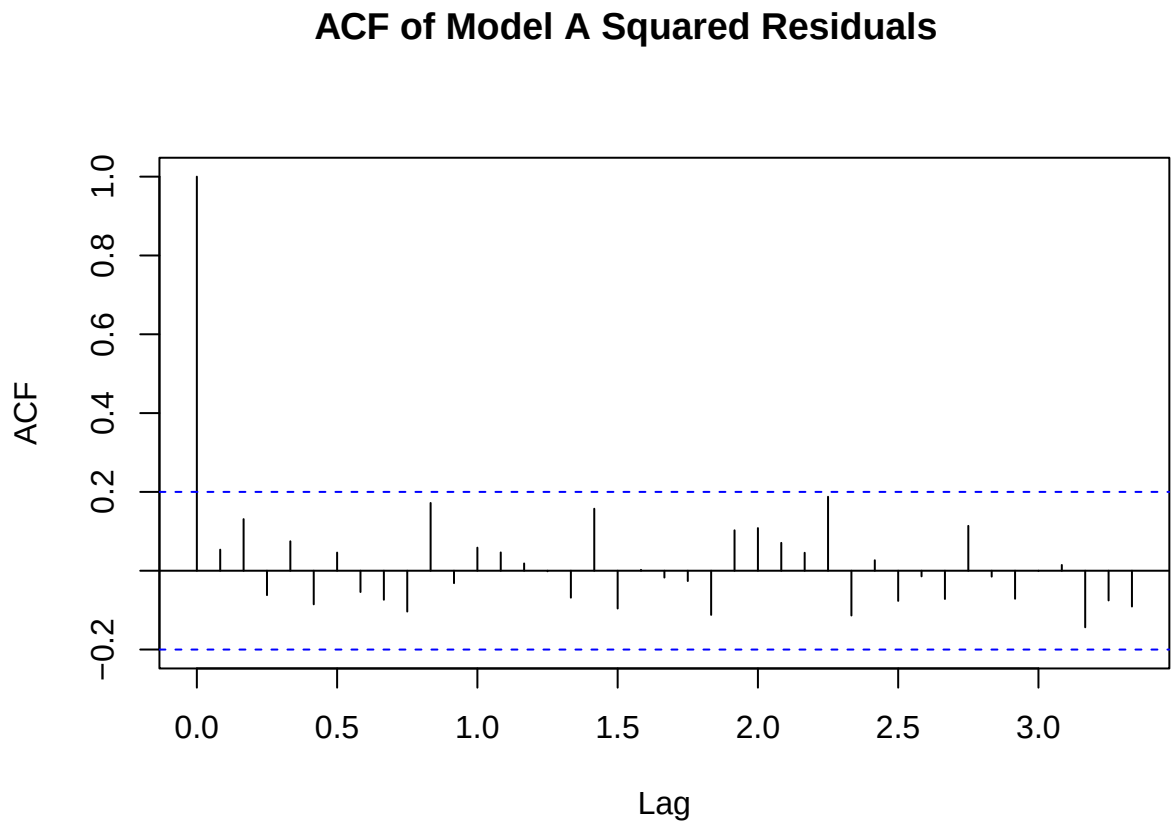
```
##
##  Box-Ljung test
##
## data:  res1
## X-squared = 14.559, df = 10, p-value = 0.149
```

```
Box.test(res1^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
```

```
## data:  res1^2
## X-squared = 9.7066, df = 12, p-value = 0.6417
```

```
acf(res1^2, lag.max=40,, main="")
title("ACF of Model A Squared Residuals", line = -1, outer = TRUE)
```




```
ar(res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  1.396
```

```
#Conclusion: All ACF and PACF of residuals are within confidence intervals and can be counted as zeros
```

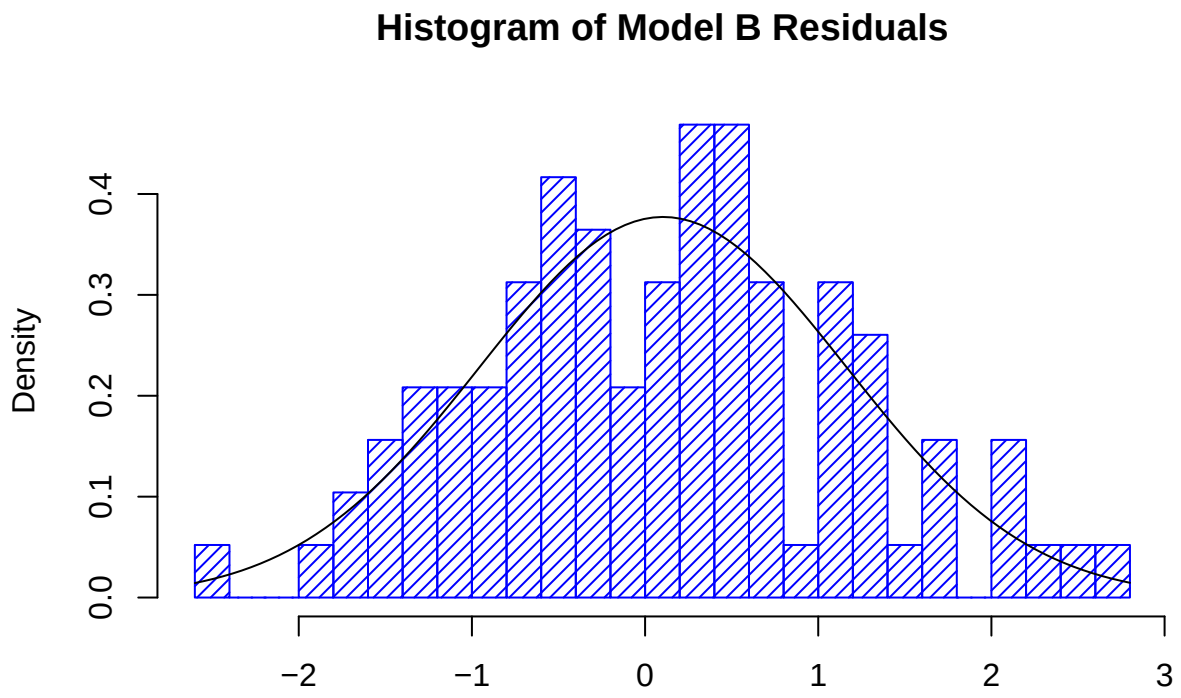
DIAGNOSTIC CHECKING FOR MODEL B

```
#Analyzing normal distribution of residuals
res3 <- residuals(fit3_fixed)
shapiro.test(res3)
```

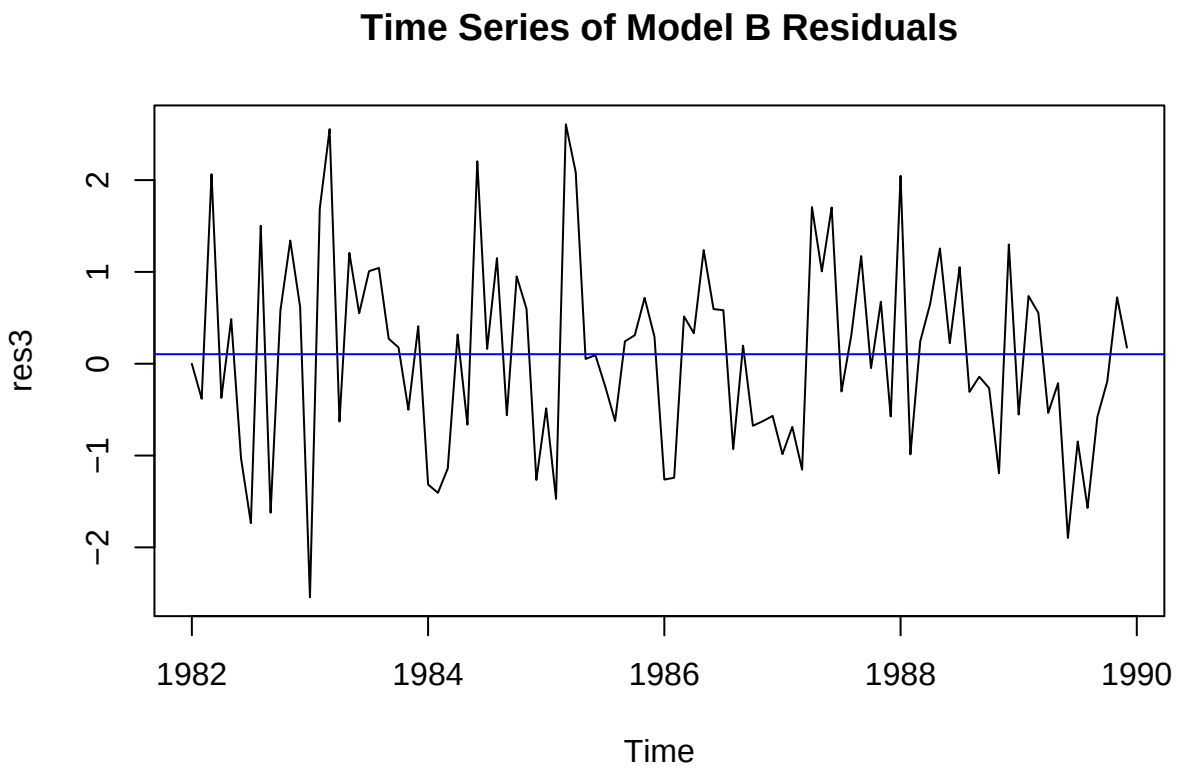
```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  res3
## W = 0.99172, p-value = 0.8205
```

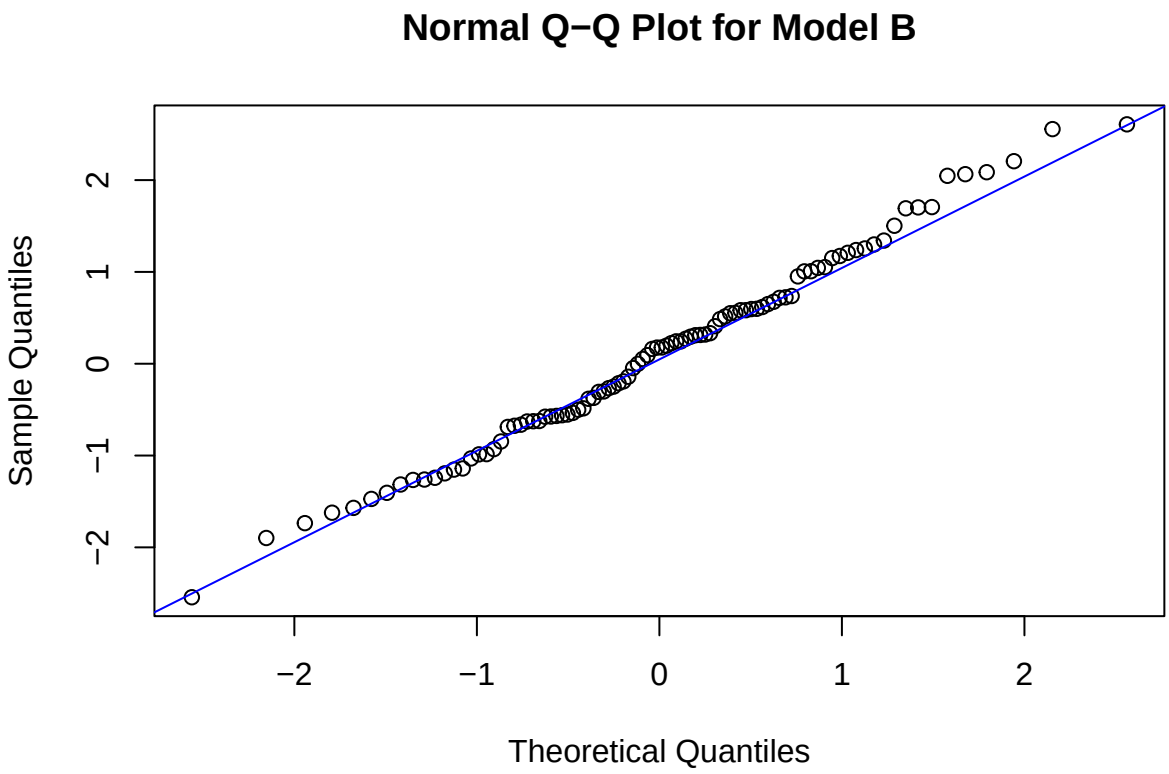
```
hist(res3, density=20,breaks=20, col="blue", xlab="", main = "Histogram of Model B Residuals", prob=TRU
m3 <- mean(res3)
std3 <- sqrt(var(res3))
curve(dnorm(x,m3,std3), add=TRUE )
```



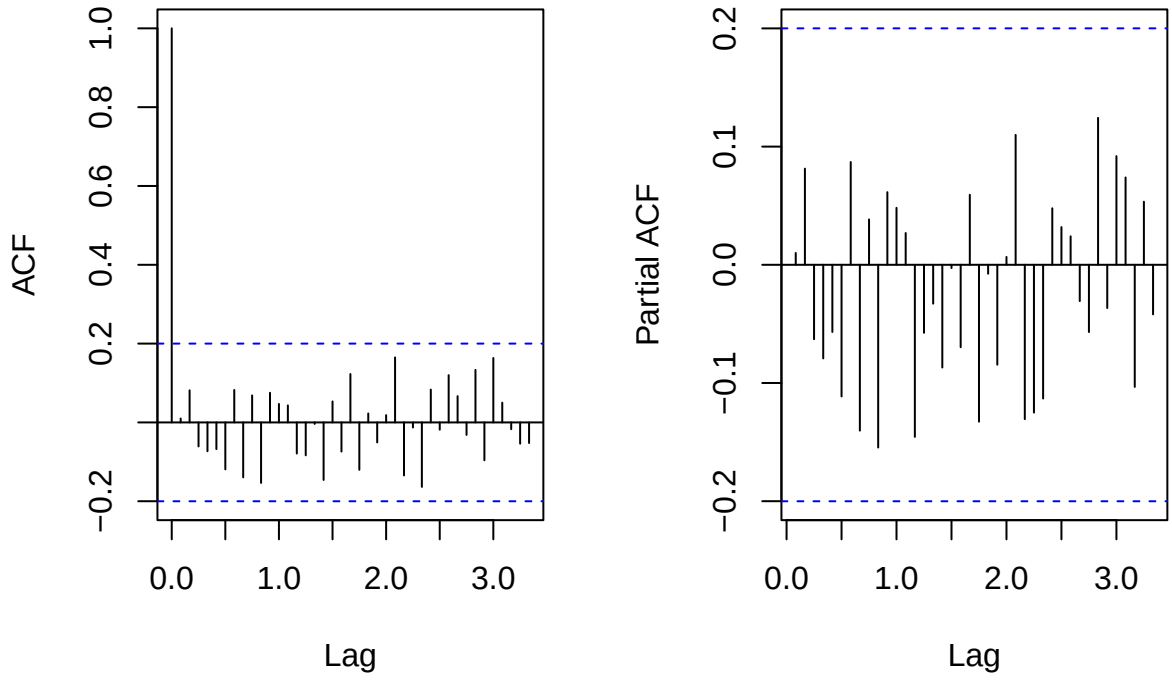

```
#Plotting residuals as time series, qqplot, and acf/pacf
plot.ts(res3, main="Time Series of Model B Residuals")
fitt3 <- lm(res3 ~ as.numeric(1:length(res3))); abline(fitt3, col="red")
abline(h=mean(res3), col="blue")
```




```
qqnorm(res3,main= "Normal Q-Q Plot for Model B")
qqline(res3,col="blue")
```




```
op = par(mfrow = c(1,2))
acf(res3, lag.max=40, main="")
pacf(res3, lag.max=40, main="")
title("ACF & PACF of Model B Residuals", line = -1, outer = TRUE)
```




```
par(op)

#Box Tests
Box.test(res3, lag = 12, type = c("Box-Pierce"), fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res3
## X-squared = 9.3144, df = 10, p-value = 0.5025
```

```
Box.test(res3, lag = 12, type = c("Ljung-Box"), fitdf = 2)
```

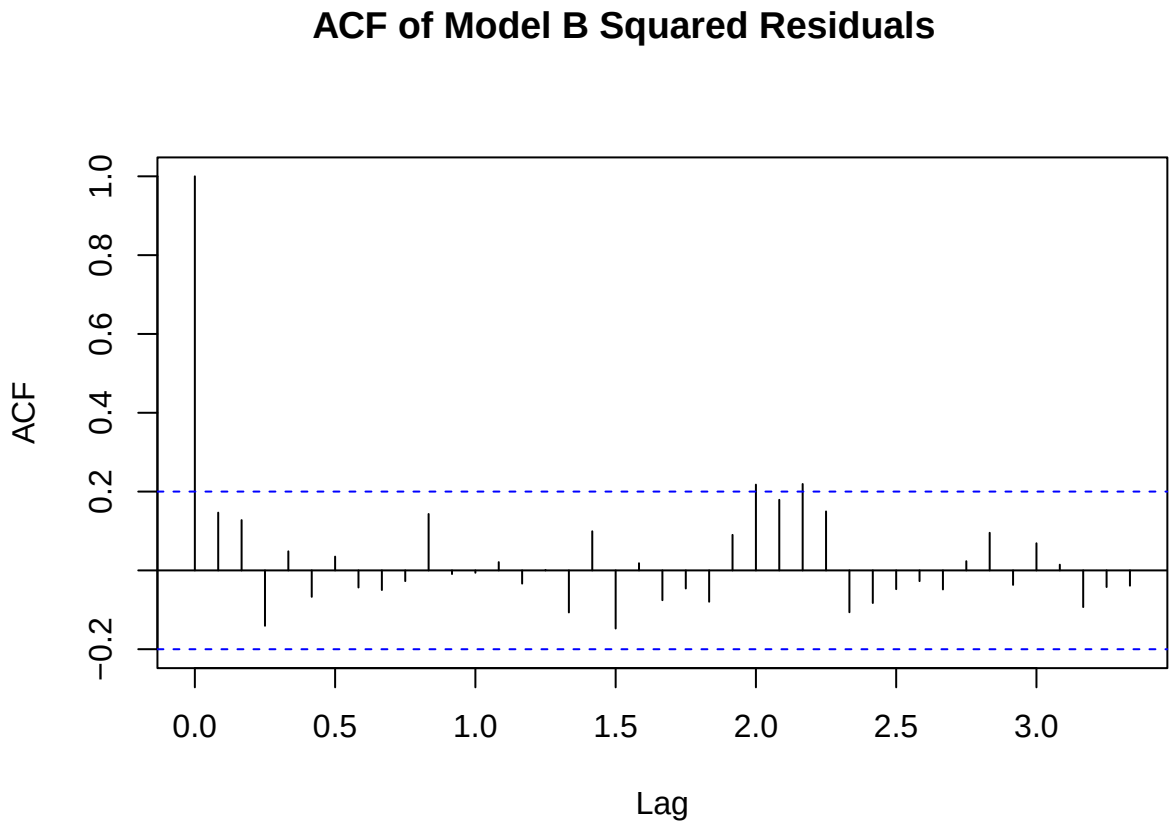
```
##
##  Box-Ljung test
##
## data:  res3
## X-squared = 10.319, df = 10, p-value = 0.4129
```

```
Box.test(res3^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
```

```
## data:  res3^2
## X-squared = 9.384, df = 12, p-value = 0.6698
```

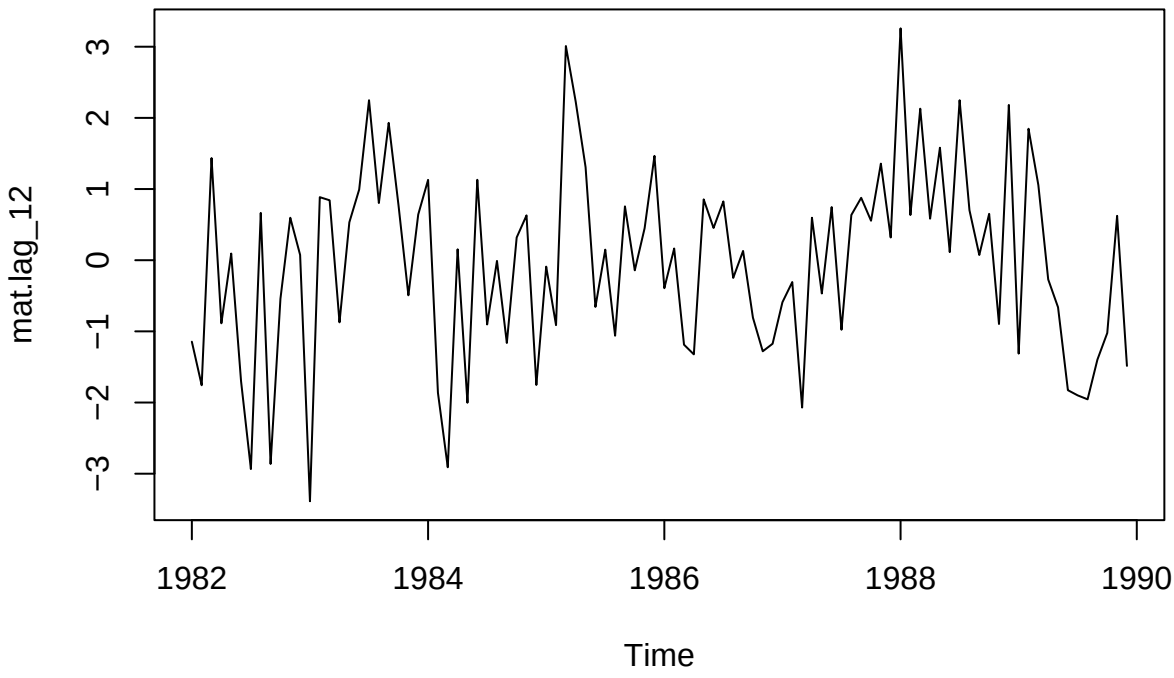
```
acf(res3^2, lag.max=40,, main="")
title("ACF of Model B Squared Residuals", line = -1, outer = TRUE)
```




```
#Con
```

FORECASTING USING MODEL B

```
ts.plot(mat.lag_12)
```

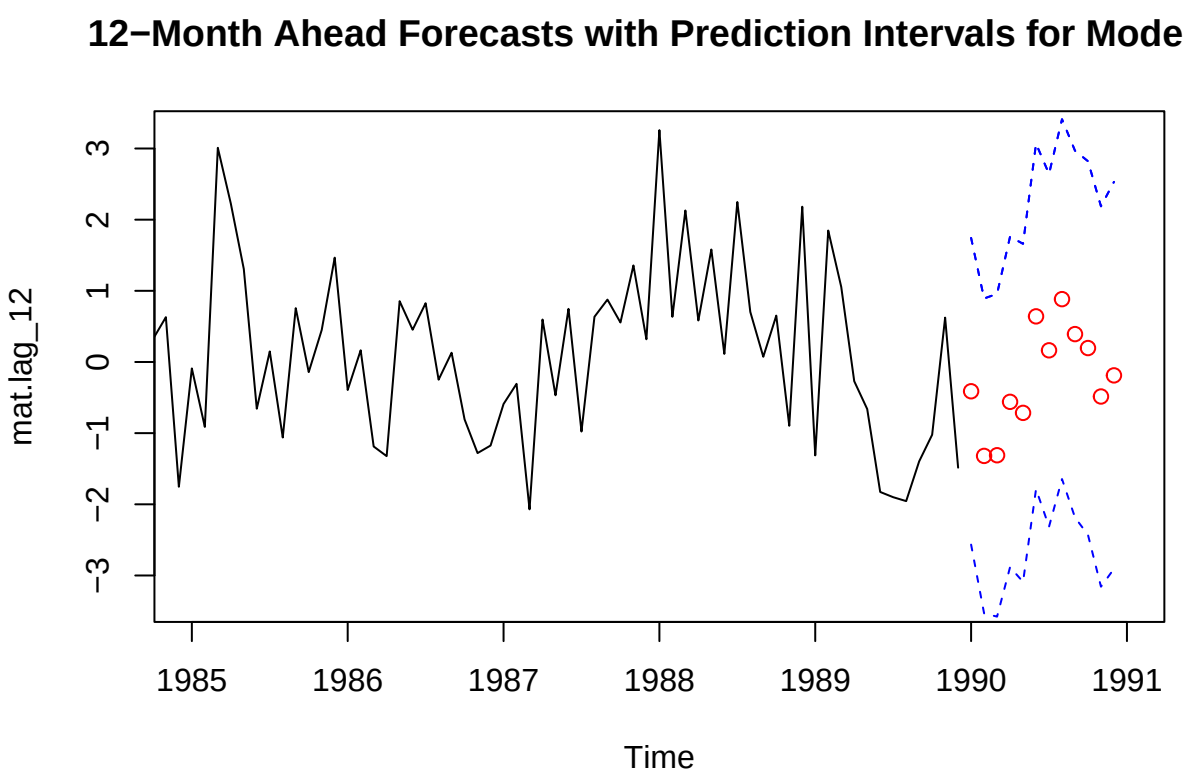



```
fib <- Arima(mat.lag_12, order=c(1,1,1), seasonal=list(order=c(0,0,1), period=12), fixed = c(0, NA, NA))
forecast(fib)
```

```
##          Point Forecast      Lo 80     Hi 80     Lo 95     Hi 95
## Jan 1990    -0.41150871 -1.7922868 0.9692694 -2.523227 1.7002092
## Feb 1990    -1.32002157 -2.7374676 0.0974245 -3.487818 0.8477751
## Mar 1990    -1.31058090 -2.7637700 0.1426082 -3.533042 0.9118800
## Apr 1990    -0.56007495 -2.0481488 0.9279989 -2.835887 1.7157376
## May 1990    -0.71502194 -2.2371812 0.8071373 -3.042964 1.6129198
## Jun 1990     0.64108535 -0.9144126 2.1965833 -1.737844 3.0200142
## Jul 1990     0.16400283 -1.4241342 1.7521398 -2.264843 2.5928488
## Aug 1990     0.88310129 -0.7370173 2.5032199 -1.594656 3.3608589
## Sep 1990     0.39172937 -1.2597516 2.0432104 -2.133993 2.9174516
## Oct 1990     0.19586115 -1.4863976 1.8781199 -2.376932 2.7686539
## Nov 1990    -0.48484602 -2.1973295 1.2276375 -3.103864 2.1341715
## Dec 1990    -0.18812927 -1.9303132 1.5540547 -2.852570 2.4763112
## Jan 1991    -0.08271919 -1.9813113 1.8158729 -2.986365 2.8209268
## Feb 1991    -0.08271919 -1.9825562 1.8171178 -2.988269 2.8228308
## Mar 1991    -0.08271919 -1.9838003 1.8183619 -2.990172 2.8247335
## Apr 1991    -0.08271919 -1.9850436 1.8196052 -2.992073 2.8266350
## May 1991    -0.08271919 -1.9862861 1.8208477 -2.993974 2.8285352
## Jun 1991    -0.08271919 -1.9875278 1.8220894 -2.995873 2.8304342
## Jul 1991    -0.08271919 -1.9887687 1.8233303 -2.997770 2.8323319
## Aug 1991    -0.08271919 -1.9900087 1.8245703 -2.999667 2.8342284
## Sep 1991    -0.08271919 -1.9912480 1.8258096 -3.001562 2.8361237
## Oct 1991    -0.08271919 -1.9924864 1.8270481 -3.003456 2.8380178
```

```
## Nov 1991    -0.08271919 -1.9937241 1.8282857 -3.005349 2.8399106
## Dec 1991    -0.08271919 -1.9949609 1.8295226 -3.007241 2.8418022
```

```r
pre <- predict(fib, n.ahead = 12)
U.t <- pre$pred + 2*pre$se

# % lower bound of prediction interval
L.t  = pre$pred - 2*pre$se

ts.plot(mat.lag_12, xlim = c(1985,1991), main = "12-Month Ahead Forecasts with Prediction Intervals for
lines(U.t, col = "blue", lty = "dashed")

lines(U.t, col = "blue", lty = "dashed")
lines(L.t, col = "blue", lty = "dashed")
points(pre$pred, col = "red")
```



Although the model above scored a lower AICc upon visual inspection, the model with the second lower AICc performed better when plotting within confidence intervals. Likewise as both are stationary AND invertible, I decided to proceed with Model A for the forecasting elements.

#underfitting, overfitting, or too simplistic

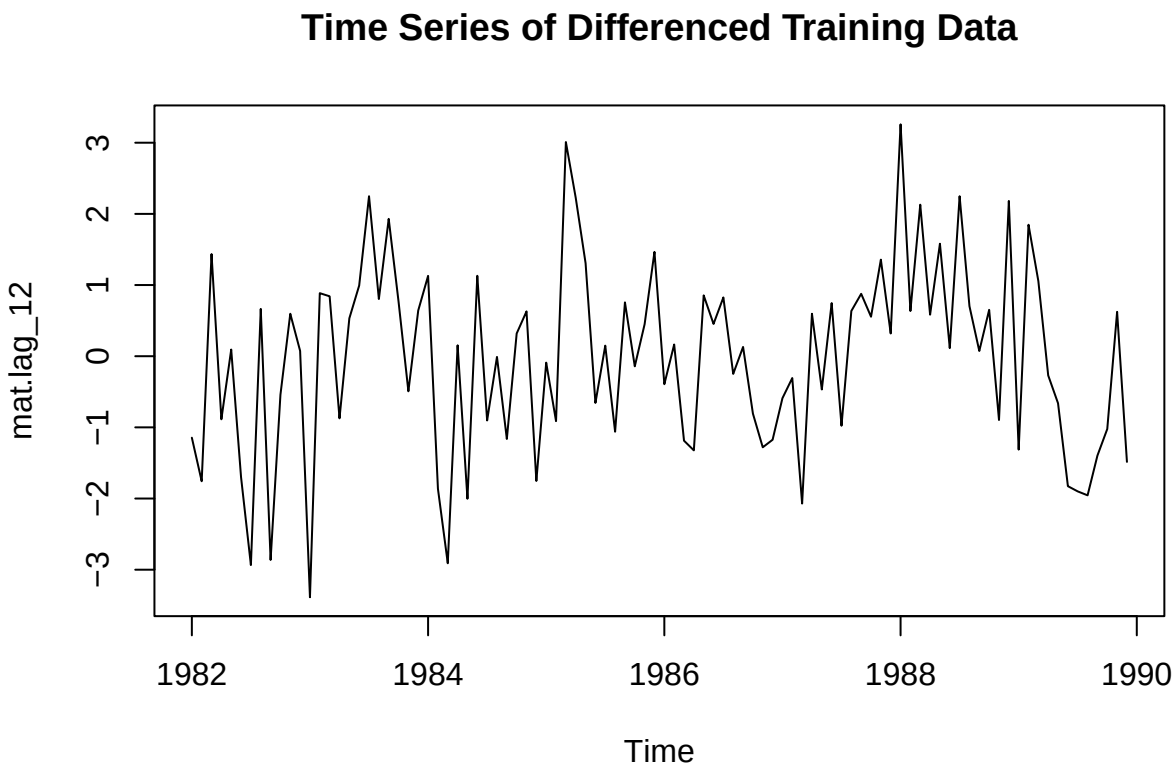
MODEL A Forecasting

```
fit.A <- arima(mat.lag_12, order=c(1,1,1), seasonal=list(order=c(1,0,0), period = 12), method = "ML")
forecast(fit.A)
```

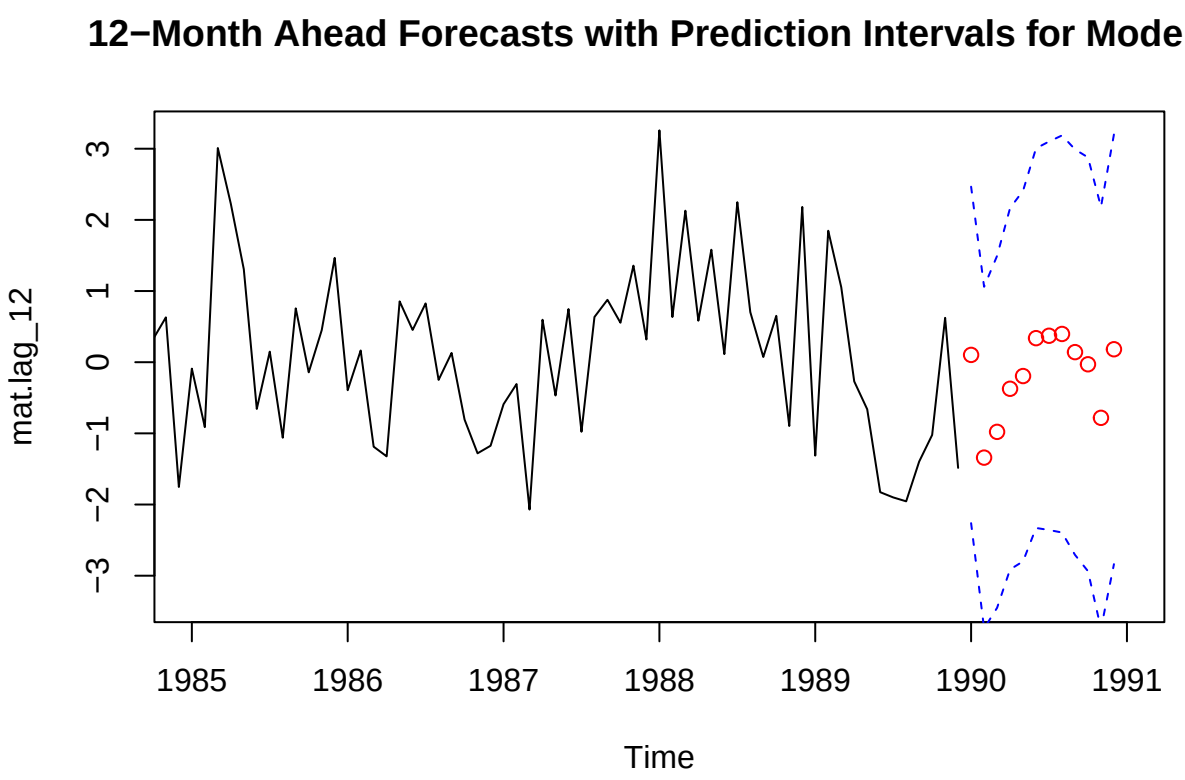
```
##          Point Forecast     Lo 80     Hi 80     Lo 95    Hi 95
## Jan 1990     0.10248045 -1.412261 1.6172221 -2.214117 2.419078
## Feb 1990    -1.34083374 -2.879249 0.1975813 -3.693636 1.011969
## Mar 1990    -0.97877091 -2.563271 0.6057295 -3.402055 1.444513
## Apr 1990    -0.37333462 -2.000362 1.2536932 -2.861659 2.114990
## May 1990    -0.19452192 -1.863186 1.4741425 -2.746524 2.357480
## Jun 1990     0.33803610 -1.371234 2.0473065 -2.276067 2.952140
## Jul 1990     0.37154823 -1.377387 2.1204835 -2.303217 3.046314
## Aug 1990     0.39660863 -1.391112 2.1843289 -2.337474 3.130691
## Sep 1990     0.14153312 -1.684148 1.9672146 -2.650606 2.933672
## Oct 1990    -0.02941809 -1.892287 1.8334511 -2.878431 2.819595
## Nov 1990    -0.78157391 -2.680903 1.1177551 -3.686347 2.123199
## Dec 1990     0.18192458 -1.753177 2.1170266 -2.777559 3.141408
## Jan 1991    -0.54355264 -2.505237 1.4181320 -3.543690 2.456585
## Feb 1991     0.11601838 -1.861218 2.0932551 -2.907904 3.139941
## Mar 1991    -0.04943842 -2.036465 1.9375879 -3.088333 2.989456
## Apr 1991    -0.32611292 -2.323328 1.6711019 -3.380589 2.728364
## May 1991    -0.40782741 -2.415139 1.5994844 -3.477746 2.662091
## Jun 1991    -0.65119773 -2.668559 1.3661640 -3.736486 2.434091
## Jul 1991    -0.66651223 -2.693874 1.3608493 -3.767094 2.434070
## Aug 1991    -0.67796442 -2.715277 1.3593479 -3.793765 2.437836
## Sep 1991    -0.56139907 -2.608614 1.4858156 -3.692344 2.569546
## Oct 1991    -0.48327716 -2.540347 1.5737922 -3.629293 2.662739
## Nov 1991    -0.13955423 -2.206431 1.9273229 -3.300570 3.021462
## Dec 1991    -0.57985731 -2.656496 1.4967812 -3.755802 2.596087
```

```
pred.tr <- predict(fit.A, n.ahead = 12)

U.tr = pred.tr$pred + 2*pred.tr$se

# % lower bound of prediction interval
L.tr = pred.tr$pred - 2*pred.tr$se
```

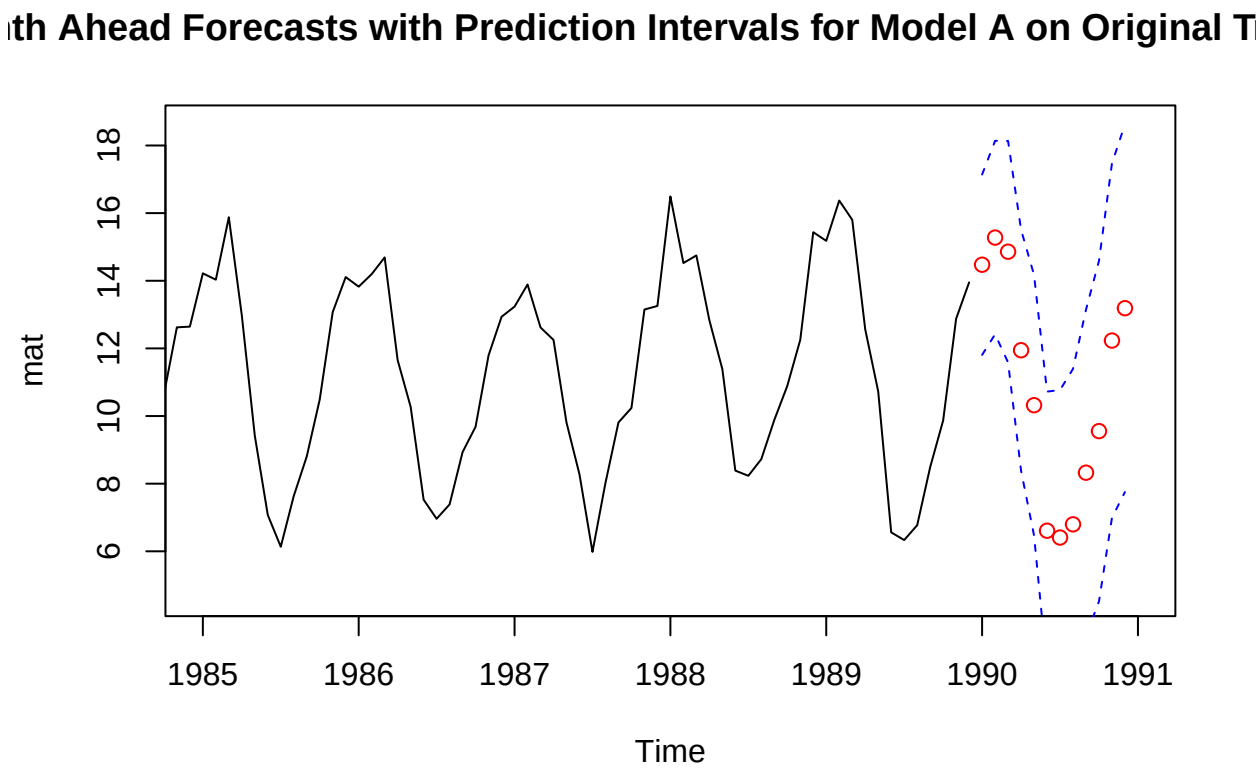
```
ts.plot(mat.lag_12, main="Time Series of Differenced Training Data")
```



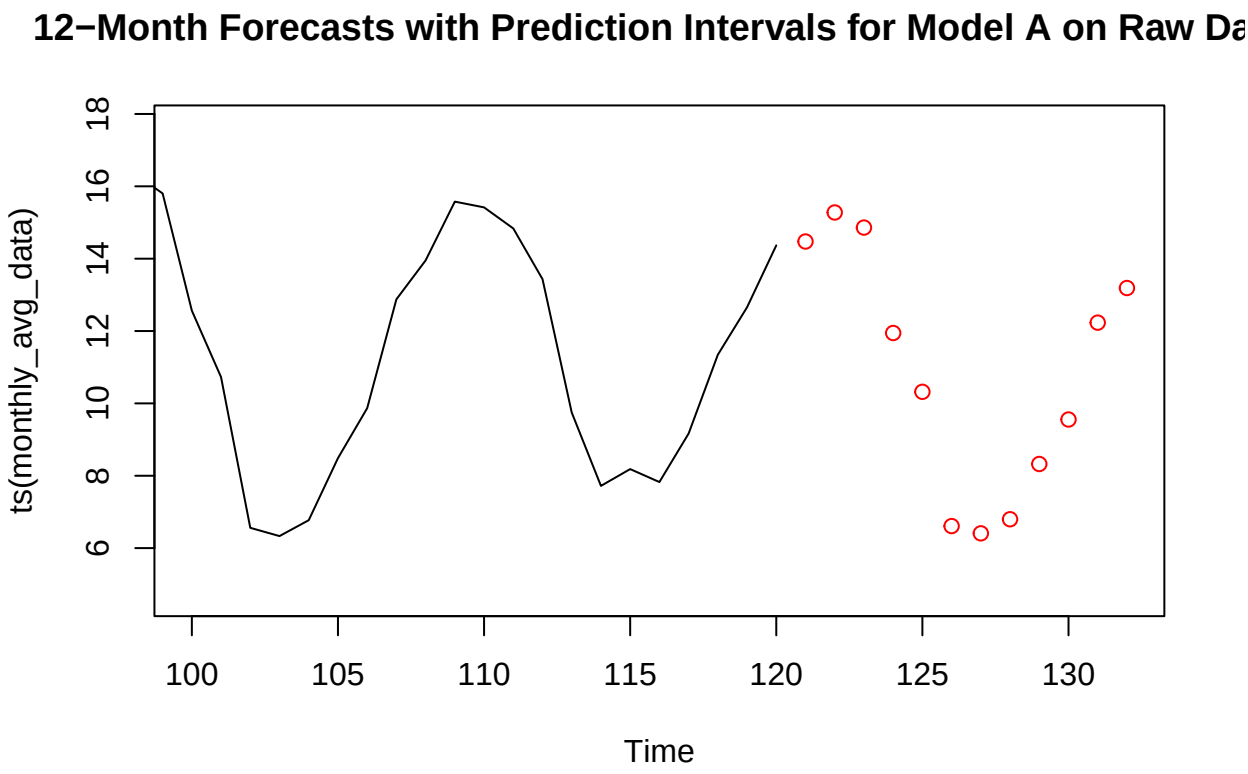

```
ts.plot(mat.lag_12, xlim = c(1985,1991), main = "12-Month Ahead Forecasts with Prediction Intervals for

lines(U.tr, col = "blue", lty = "dashed")
lines(L.tr, col = "blue", lty = "dashed")
points(pred.tr$pred, col = "red")
```




```
fit.A <- arima(mat, order=c(1,1,1), seasonal=list(order=c(1,0,0), period = 12), method = "ML")
ab <- predict(fit.A, n.ahead = 12)

u <- ab$pred + 2* ab$se
l <- ab$pred - 2* ab$se
ts.plot(mat, xlim = c(1985,1991), ylim = c(min(mat),max(u)), main = "12-Month Ahead Forecasts with Predi

lines(u, col = "blue", lty = "dashed")
lines(l, col = "blue", lty = "dashed")
points(ab$pred, col = "red")
```




```
ts.plot(ts(monthly_avg_data), xlim = c(100, length(monthly_avg_data)+12), ylim = c(min(monthly_avg_data)
lines(u, col = "blue", lty = "dashed")
lines(l, col = "blue", lty = "dashed")
points((length(monthly_avg_data)+1):(length(monthly_avg_data)+12), ab$pred, col = "red")
```



Takeaways from this project is that although the data does follow a time series, the need to compile the daily entries into months clearly caused issues. The variances of outlier dates can significantly effect how my information will behave and I attribute that to a degree on the forecasting results and difficulties modeling throughout this project. In the future I will perhaps work exclusively with monthly entries in order to reduce the potential of noise in my end project result.