SARIMA Modeling for the M1 Supply

Pstat 274 Final Project

Jake Merry

06.06.2025

Abstract

We analyze the U.S. M1 monetary supply in order to produce short-term forecasts of economic growth. We begin by transforming the data to create a stationary series in order to apply Box-Jenkins methodology and fit the data set to a SARIMA model. We perform diagnostic checks and spectral analysis to ensure the validity of our chosen model. Using the fitted SARIMA$(3, 2, 3) \times (1, 0, 1)_{12}$ model, we are able to forecast observations 2 years in advance. These predictions help us better understand the influence of trend and seasonality on liquid assets and gain insight into how public spending and bank lending changes with time.

Introduction

The M1, M2, and M3 monetary aggregates are measures of the U.S. money supply and are used to analyze macroeconomic activity. The M1 supply represents liquid assets including physical currency, checks, and checking accounts. The M2 supply is made of everything included in the M1 supply, plus assets that are not considered as liquid, such as savings accounts and mutual funds. The M3 supply is less commonly used than the M1 and M2. It includes the M2 as well as larger deposits. This report will focus on the M1 supply, a key indicator of economic activity and liquidity, in order to reflect current day-to-day spending and purchasing power of the public and businesses in the U.S..

The monthly data used in this report was downloaded on May 28, 2025 from the **Federal Reserve Bank of St. Louis**. Our main goal in this report is to use Box-Jenkins methodology to fit this data set to a SARIMA model of the form $X_t = m_t + s_t + S_t$, where m_t is the trend component, s_t is the seasonal component, and S_t is a stationary process. We do this in order to forecast future observations up to 24 months in advance. In doing so, we hope to gain insight into how public spending and bank lending changes with time.

The steps taken to forecast this data set are standard. We first perform data transformations to achieve a stationary series. We then examine ACFs and PACFS, as well as AICc values, in order to pick candidate models to fit the series. Next, we check invertiblilty and causality of the selected models and perform diagnostic checking of the residuals to ensure homoskedasticity, and finally we can forecast future values using the selected model. Additionally, we will also perform spectral analysis to ensure that the period we chose for our model is accurate.

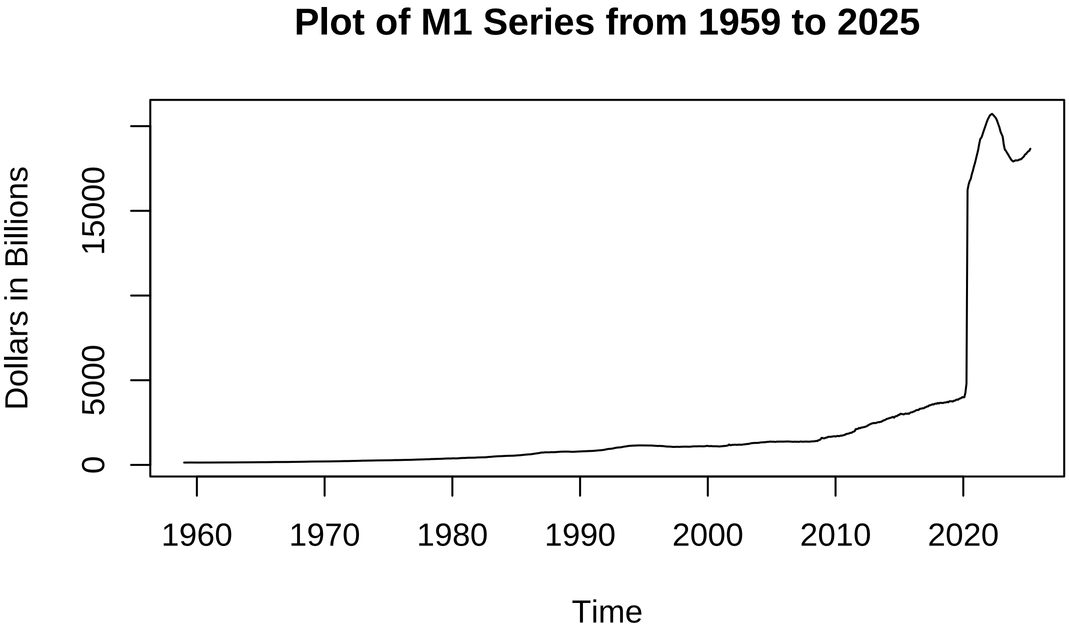
Below is the setup chunk used for the report, including all libraries used:

```r
knitr::opts_chunk$set(echo=FALSE, fig.dim=c(6, 3.8), warning=FALSE)
library(forecast)
library(UnitCircle)
library(TSA)
library(GeneCycle)
```

Data Analysis

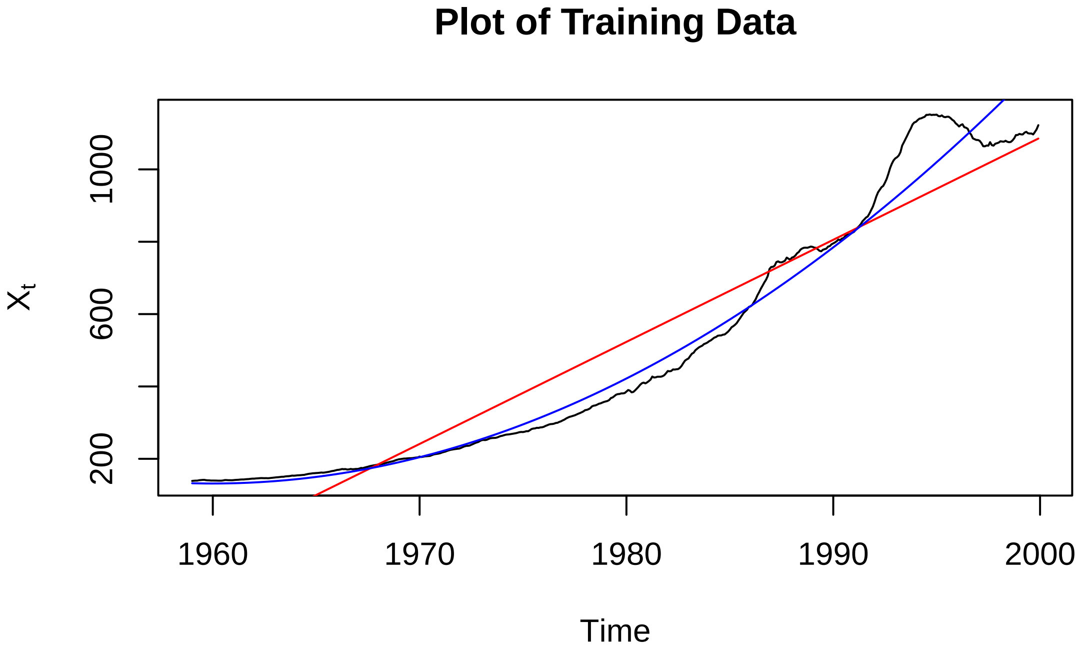
Exploration of Data Set

Our first step is to analyze the raw data in our series. We will do this simply by converting the data to a time series object and plotting against time.



The main problem that we see here is that around 2020 (COVID years), the data spikes considerably. This would not only make prediction of these years extremely difficult, but would also influence our model in ways that would make forecasting past these dates inaccurate. Hence, we will restrict to data before the year 2000. Specifically, our training set will consist of data from January 1959 to December 1999, and our testing set will consist of the data from January 2000 to December 2001. We will denote the training series as X_t.

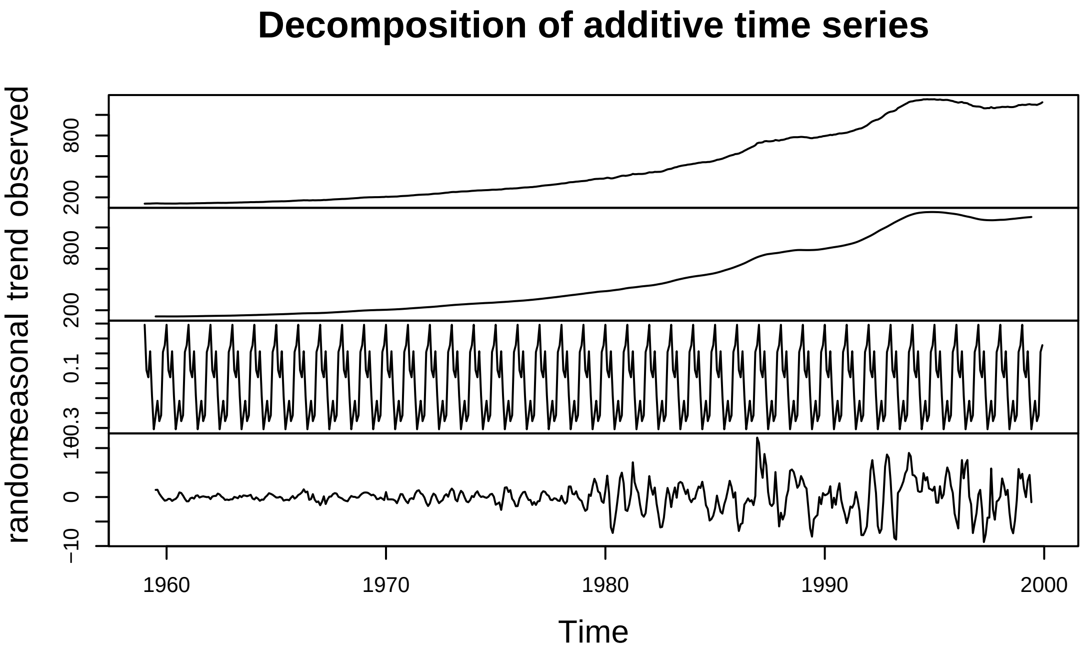
We can then plot the training set to get a better idea of the specific problem we are faced with.



Our training data appears to follow a quadratic trend, so we note that differencing twice lag one may be our best bet for creating a stationary series. First, however, we take note of the current variance of the data set:

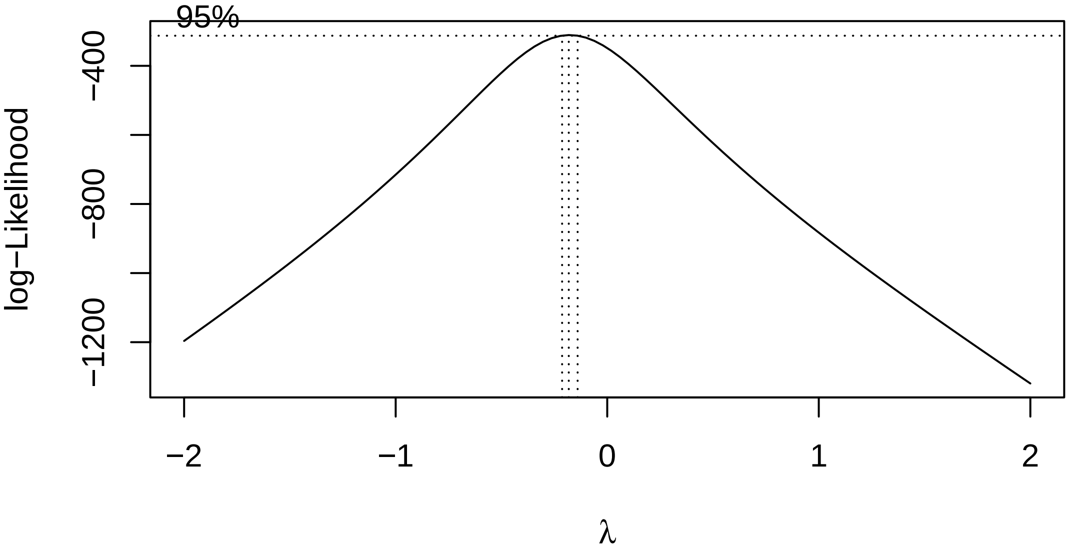
```
## [1] 123121.6
```

We can also plot the decomposition of the data set in order to see how each component of the series will affect our model.



Data Transformation

We now need to perform transformations to our data set in order to create a stationary series to follow the Box-Jenkins methodology. Our first attempt at stabilizing the data will be through a Box-Cox transformation.



We note that zero is not included in the confidence interval of our λ values, so we pull the true value of λ for our transformation, given as

```
## [1] -0.1818182
```

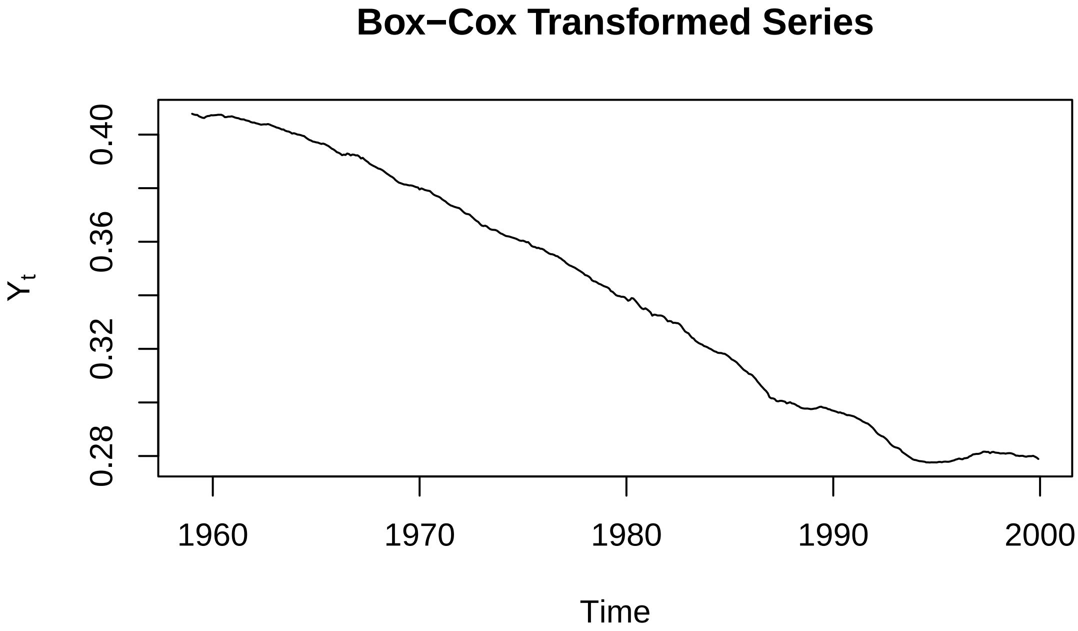
For sake of simplicity, we will not use the full Box-Cox transformation formula,

$$Y_t = \frac{X_t^\lambda - 1}{\lambda},$$

but rather simplify to

$$Y_t = X_t^\lambda.$$

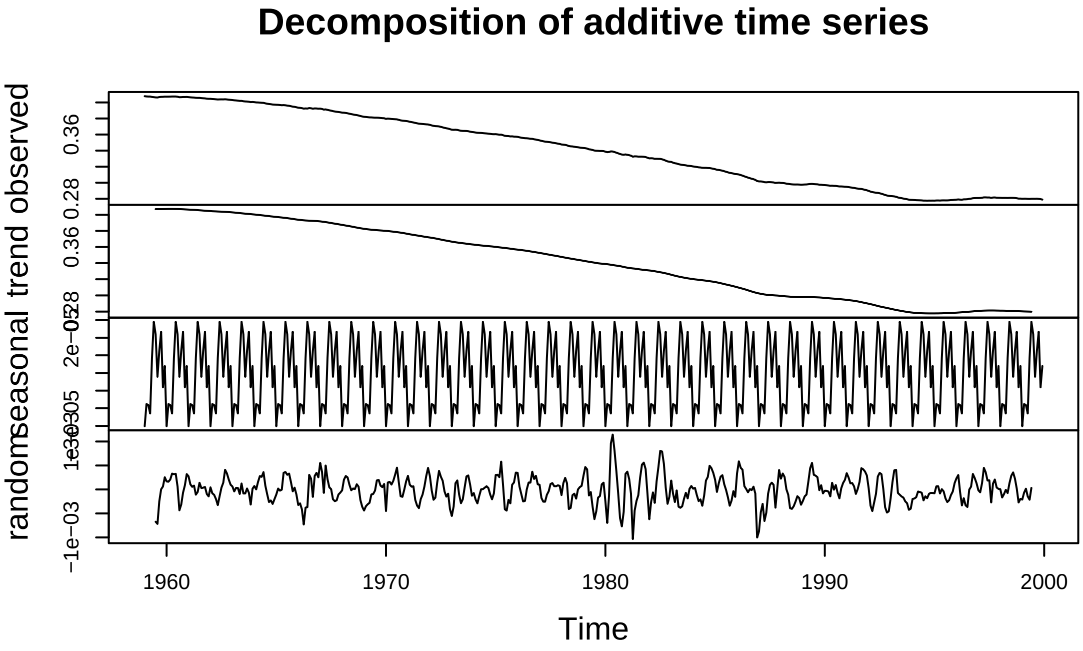
Plotting the transformed data, we have



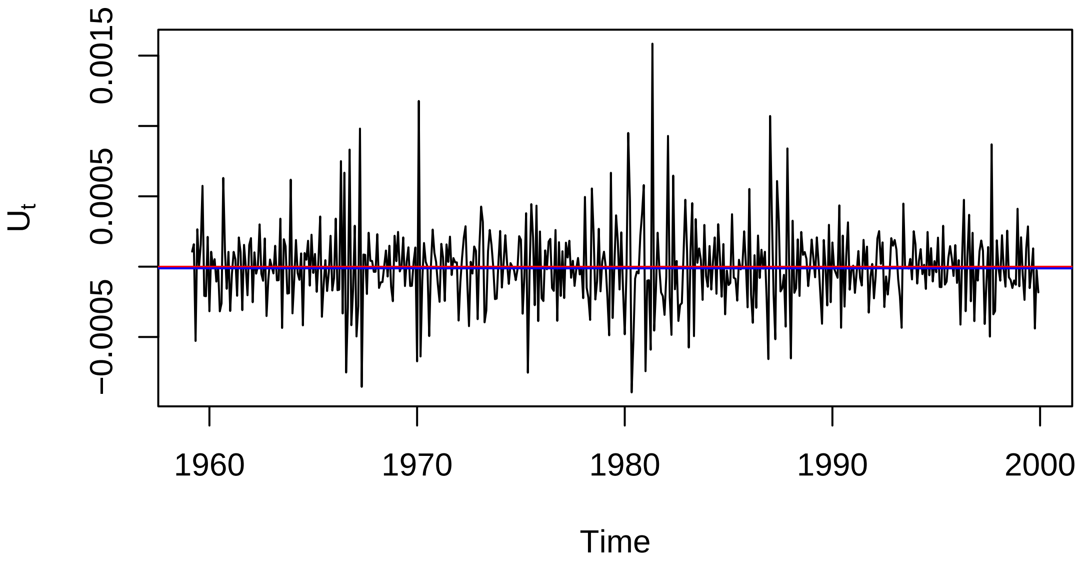
And, observing the new variance, we have

```
## [1] 0.002012525
```

which is considerably lower than the variance of the original data set (123121.6). Hence, we accept this transformation as valid for our scenario. Then, plotting the decomposed series, we observe



Our plot of the transformed data shows that our series does not appear to have a strong seasonal pattern, but since it does show a clear quadratic trend, we choose to difference twice at lag 1 to remove this, letting $U_t = (1 - B)^2 Y_t$.


Series Differenced Twice at Lag 1

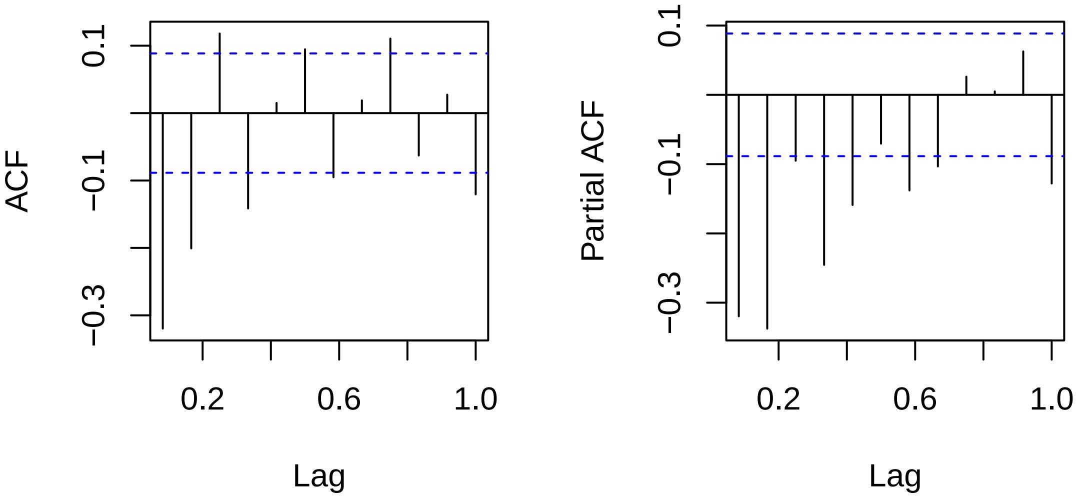
Observing the new variance as

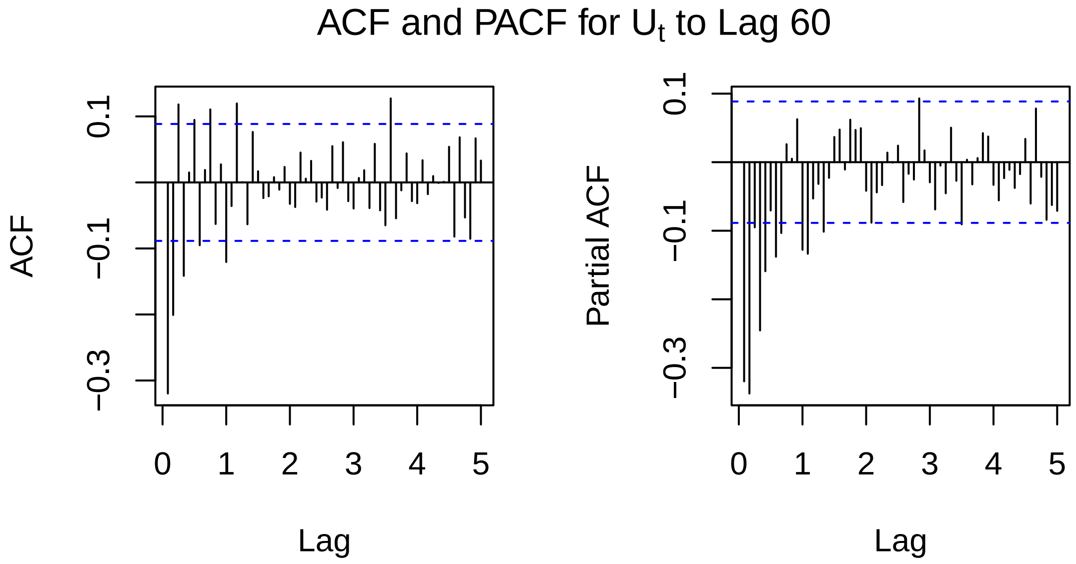
```
## [1] 8.323216e-08
```

it is clear that U_t is stationary and is the best presentation of the series for our purposes.

Model Selection

We can start our model selection process by analyzing the ACF and PACF plots for our stationary series. We analyze the ACFs and PACFs between lags 1 and 12 in order to choose candidate p and q values, and we observe the ACFs and PACFs at lags $12k$ for $k \in \mathbb{N}$ to choose candidate P and Q values.


ACF and PACF for U_t to Lag 12



We see that the intra-year ACFs have significant values at lags 1, 2, and 3, and maybe at 5, 6, and 8, while those of the PACFs have signficant values at lags 1, 2, 4, 5, 7, and maybe at 3 and 8. For the inner-year ACFs, we see significant values at only at lag 1, and similarly for the PACFs. Hence, we choose the three candidate models:

1. $\text{SARIMA}(8,2,4) \times (1,0,1)_{12}$,
2. $\text{SARIMA}(2,2,4) \times (1,0,1)_{12}$,
3. $\text{ARIMA}(8,2,4)$.

These models will allow us to test a variety of different p and q values while following the observations we made of the ACFs and PACFs. We also add an ARIMA model to test a simpler model than we may expect to be accurate for our purposes.

We can now perform AICc selection on each of the models. For each, we perform the selection criterion on the original model. This results in multiple of the parameters being statistically insignificant, and we take these to be equal to zero. Below we only show the first and last steps, but it should be noted that the last step was achieved by removing one insignificant parameter at a time until none remain.

Model 1: $\text{SARIMA}(8,2,4) \times (1,0,1)_{12}$

```
## 
## Call:
## arima(x = Y_t, order = c(8, 2, 4), seasonal = list(order = c(1, 0, 1), period = 12), 
##     method = "ML")
## 
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
##       -1.0212  -0.8508  -0.1117  -0.1737  -0.1436  -0.1000  -0.0476  -0.0667
## s.e.   0.4297   0.4017   0.3507   0.1923   0.1570   0.1361   0.1114   0.0713
##          ma1      ma2      ma3      ma4    sar1     sma1
##       0.4058  -0.0088  -0.5626  -0.1458  0.3625  -0.6178
## s.e.  0.4296   0.2170   0.1502   0.3043  0.1467   0.1243
## 
## sigma^2 estimated as 5.385e-08:  log likelihood = 3404.09,  aic = -6780.18
```

```
## 
## Call:
## arima(x = Y_t, order = c(3, 2, 3), seasonal = list(order = c(1, 0, 1), period = 12), 
##     fixed = c(NA, NA, NA, 0, 0, NA, NA, NA), method = "ML")
##
## Coefficients:
##           ar1      ar2     ar3  ma1  ma2      ma3    sar1     sma1
##       -0.6378  -0.5652  0.2884    0    0  -0.6824  0.4271  -0.6721
## s.e.   0.0443   0.0465  0.0522    0    0   0.0531  0.1259   0.1041
##
## sigma^2 estimated as 5.439e-08:  log likelihood = 3401.61,  aic = -6791.23
```

Model 2: $\text{SARIMA}(2,2,4) \times (1,0,1)_{12}$

```
##
## Call:
## arima(x = Y_t, order = c(2, 2, 4), seasonal = list(order = c(1, 0, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1     ma2      ma3      ma4    sar1     sma1
##       -0.9321  -0.8831  0.3177  0.0881  -0.6608  -0.2501  0.4181  -0.6604
## s.e.   0.0713   0.0651  0.0794  0.0962   0.0639   0.0499  0.1317   0.1097
##
## sigma^2 estimated as 5.402e-08:  log likelihood = 3403.32,  aic = -6790.65
```

```
##
## Call:
## arima(x = Y_t, order = c(2, 2, 4), seasonal = list(order = c(1, 0, 1), period = 12),
##     fixed = c(NA, NA, NA, 0, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1  ma2      ma3      ma4    sar1     sma1
##       -0.9684  -0.8432  0.3480    0  -0.6406  -0.2298  0.4333  -0.6603
## s.e.   0.0933   0.0734  0.1006    0   0.0823   0.0493  0.1365   0.1150
##
## sigma^2 estimated as 5.411e-08:  log likelihood = 3402.93,  aic = -6791.85
```

Model 3: $\text{ARIMA}(8,2,4)$

```
##
## Call:
## arima(x = Y_t, order = c(8, 2, 4), method = "ML")
##
## Coefficients:
##          ar1     ar2     ar3      ar4     ar5     ar6     ar7     ar8      ma1
##       0.0937  0.0054  0.8060  -0.2823  0.1186  0.0091  0.0876  0.0127  -0.6827
## s.e.  0.2054  0.2415  0.1849   0.1138  0.0730  0.0850  0.0816  0.0748   0.2008
##           ma2      ma3     ma4
##       -0.2300  -0.7070  0.6197
## s.e.   0.2438   0.2613  0.1867
##
## sigma^2 estimated as 5.6e-08:  log likelihood = 3393.78,  aic = -6763.56
```

```
##
## Call:
## arima(x = Y_t, order = c(7, 2, 4), fixed = c(NA, NA, NA, 0, 0, 0, NA, NA, NA,
##     0, NA), method = "ML")
##
## Coefficients:
```

```
##          ar1      ar2      ar3  ar4  ar5  ar6      ar7      ma1     ma2  ma3
##      -0.3309  -0.7292  -0.4130    0    0    0  -0.0958  -0.2551  0.2821    0
## s.e.  0.0855   0.0837   0.0596    0    0    0   0.0395   0.0780  0.0930    0
##          ma4
##      -0.5745
## s.e.  0.0513
##
## sigma^2 estimated as 5.642e-08:  log likelihood = 3393.18,  aic = -6772.35
```

The only model with the same SARIMA representation as the original candidates is Model 2, but we see that each has decreased in AICc value as we removed parameters. Below is a summary of the updated model, their algebraic expressions, and their AICc values:

1. $\text{SARIMA}(3,2,3) \times (1,0,1)_{12}$ given by

$$(1 + 0.6377B + 0.5652B^2 - 0.2884B^3)(1 - 0.4271B^{12})U_t = (1 - 0.6824B^3)(1 - 0.6721B^{12})Z_t$$

   with AICc $= -6791.23$.

2. $\text{SARIMA}(2,2,4) \times (1,0,1)_{12}$ given by

$$(1 + 0.9654B + 0.8407B^2)(1 - 0.4341B^{12})U_t = (1 + 0.3452B - 0.6380B^3 - 0.2303B^4)(1 - 0.6606B^{12})Z_t$$

   with AICc $= -6791.85$.

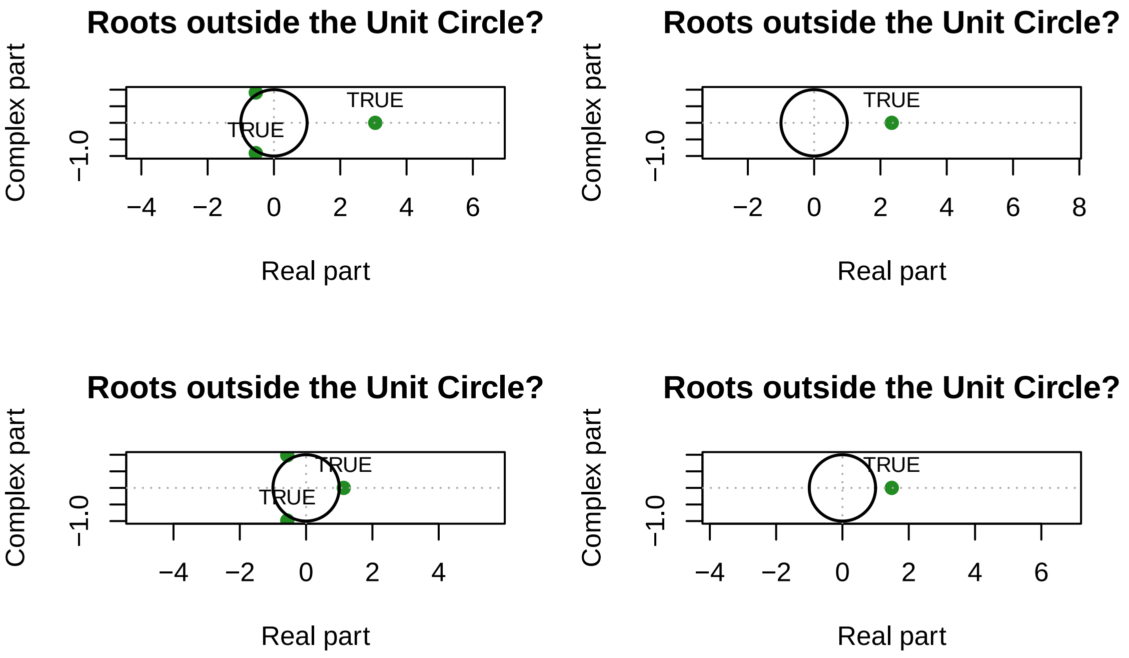
3. $\text{ARIMA}(7,0,4)$ given by

$$(1 + 0.3299B + 0.7292B^2 + 0.4126B^3 + 0.0958B^7)U_t = (1 - 0.2561B + 0.2826B^2 - 0.5742B^4)Z_t$$

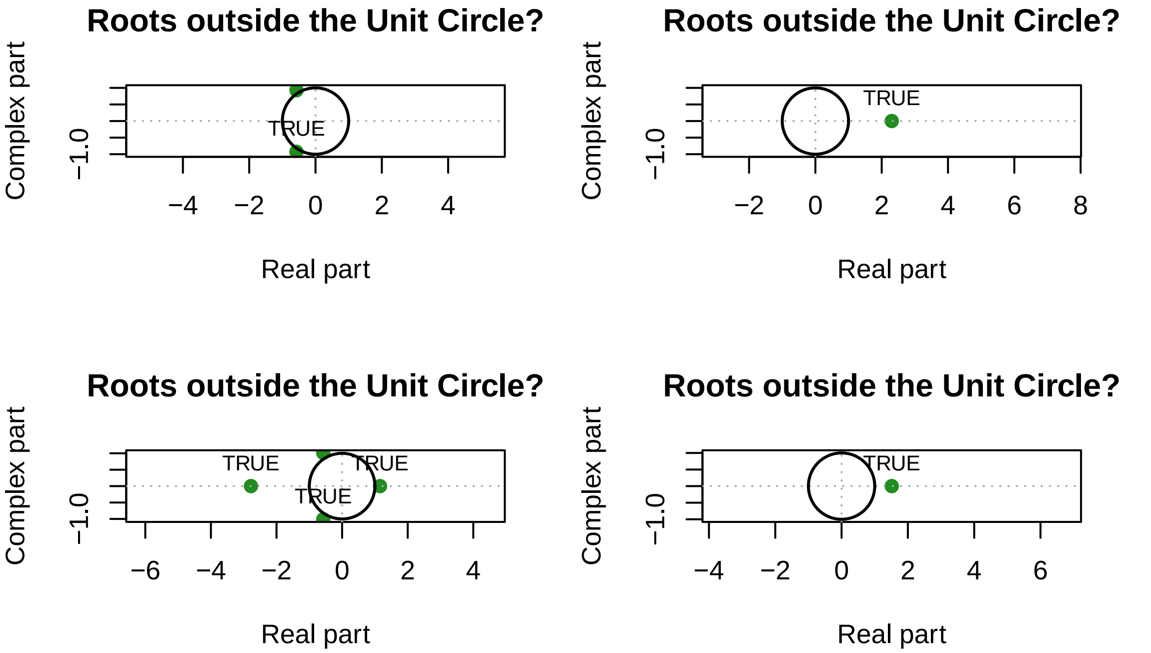
   with AICc $= -6772.35$.

Now, we can check the causality and invertibility of each model to help us decide which is the best fit for our data.

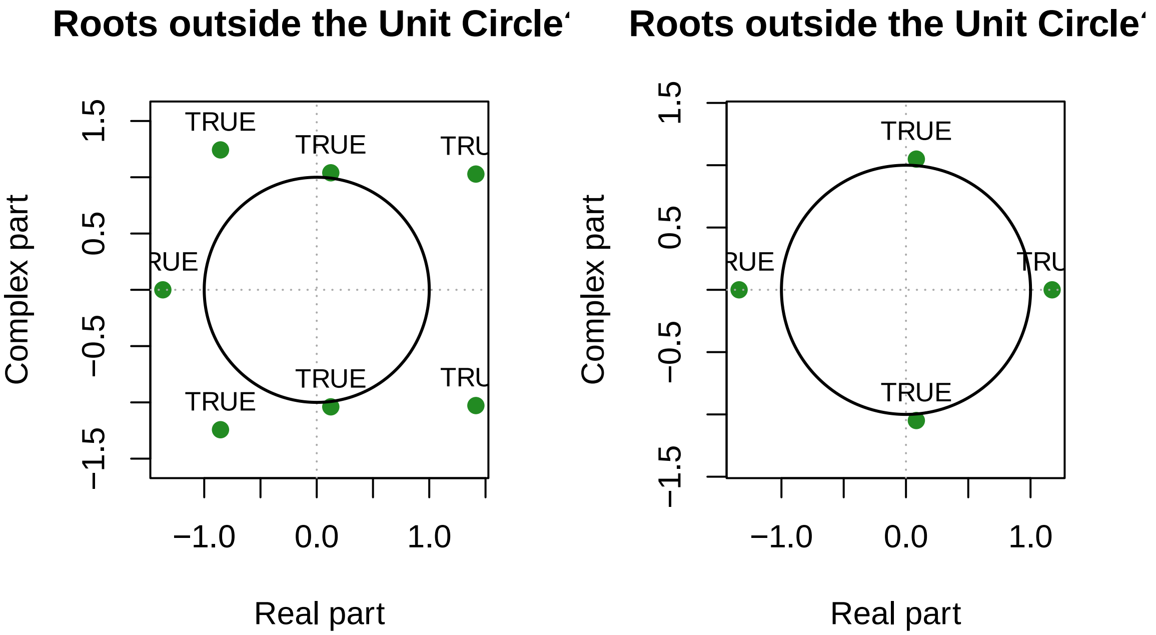
Model 1: $\text{SARIMA}(3,2,3) \times (1,0,1)_{12}$



Model 2: $\text{SARIMA}(2,2,4) \times (1,0,1)_{12}$



Model 3: $\text{ARIMA}(7,0,4)$

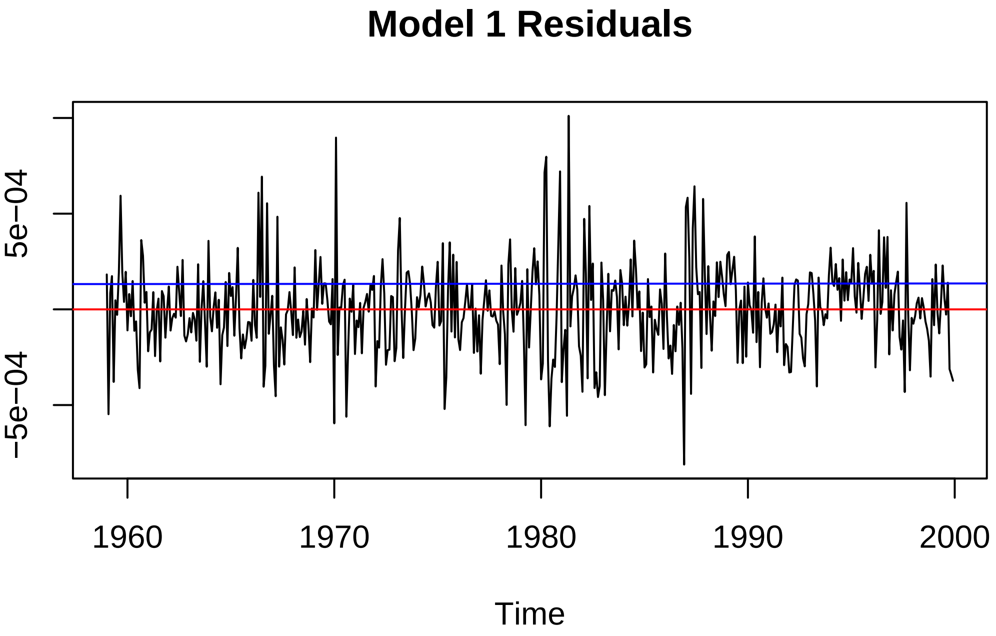


The roots of $\phi(B)$, $\Phi(B)$, $\theta(B)$, and $\Theta(B)$ for all three models all lay strictly outside of the unit circle, and hence every model is casual and invertible. Hence, we choose drop Model 3 from our current analysis since this is the model with the highest AICc value.

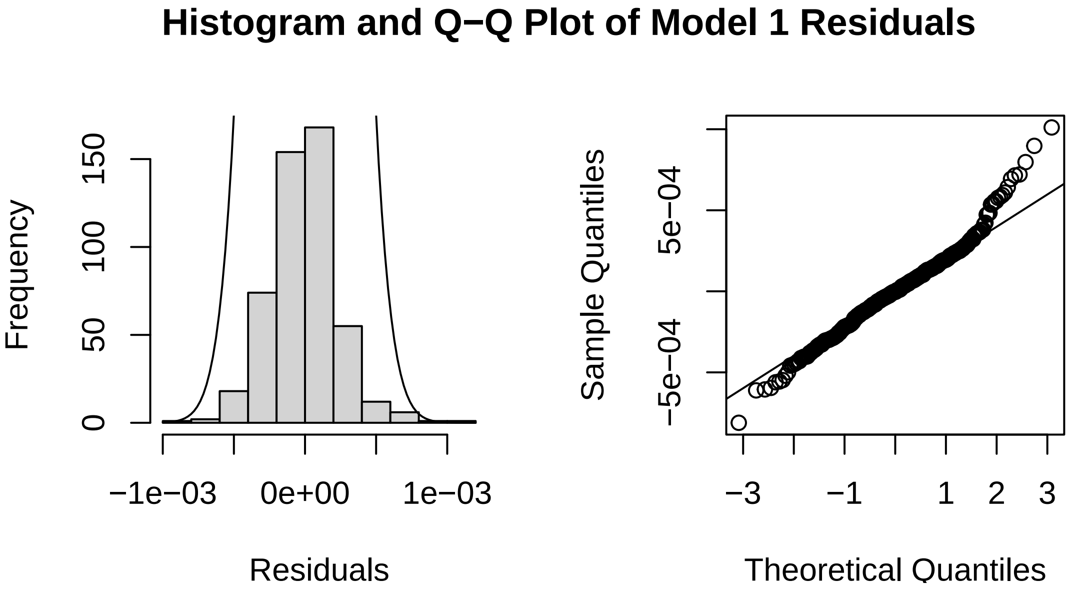
Diagnostic Checking

We can now begin our proper model diagnostics for Models 1 and 2 above. This includes plotting the residuals, analyzing the ACFs and PACFS, and performing Portmanteau tests. Let's start with Model 1.

Model 1:

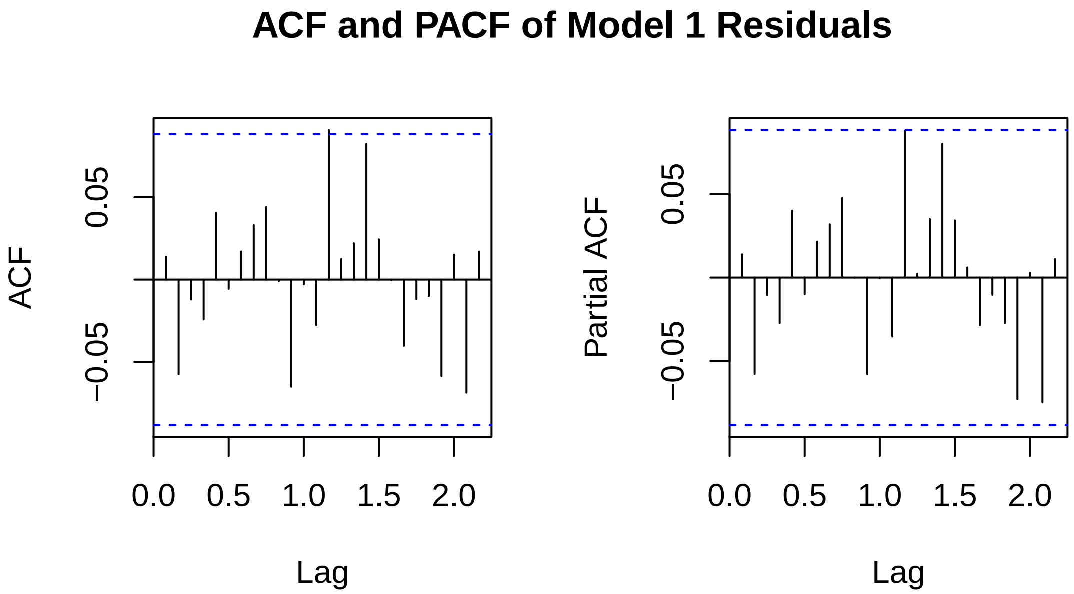


Model 1 Residuals



Histogram and Q−Q Plot of Model 1 Residuals

From the plots alone, we can see that our residuals may not be follow a white noise process as the histogram does not follow the normal curve very closely and the Q-Q Plot seems to diverge from the normal line more than we would like. However, we will still perform the remaining tests to make sure this conclusion is accurate.



The ACFs and PACFs do appear to follow a white noise process as, aside from lag 14 on the ACF plot, all values remain inside the confidence intervals. If we were to only examine these plots alone, we might assume that the residuals do actually follow a white noise process. Finally, let's perform our tests to get quantitative results for our hypothesis.

```
##
##  Shapiro-Wilk normality test
##
## data:  res1
## W = 0.97998, p-value = 2.728e-06
```

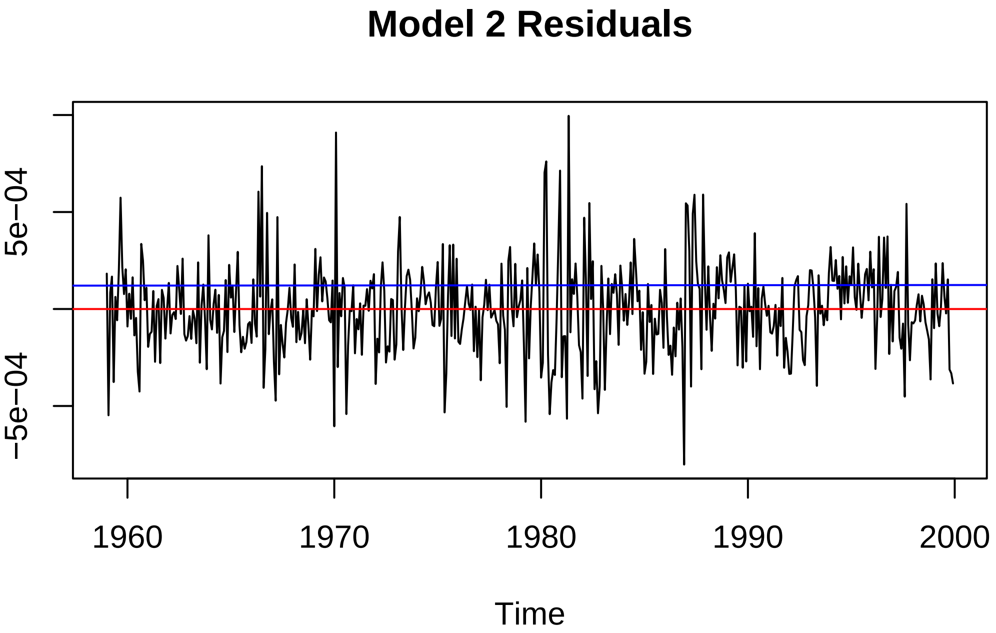
```
##
##  Box-Pierce test
##
## data:  res1
## X-squared = 15.932, df = 16, p-value = 0.4577
```

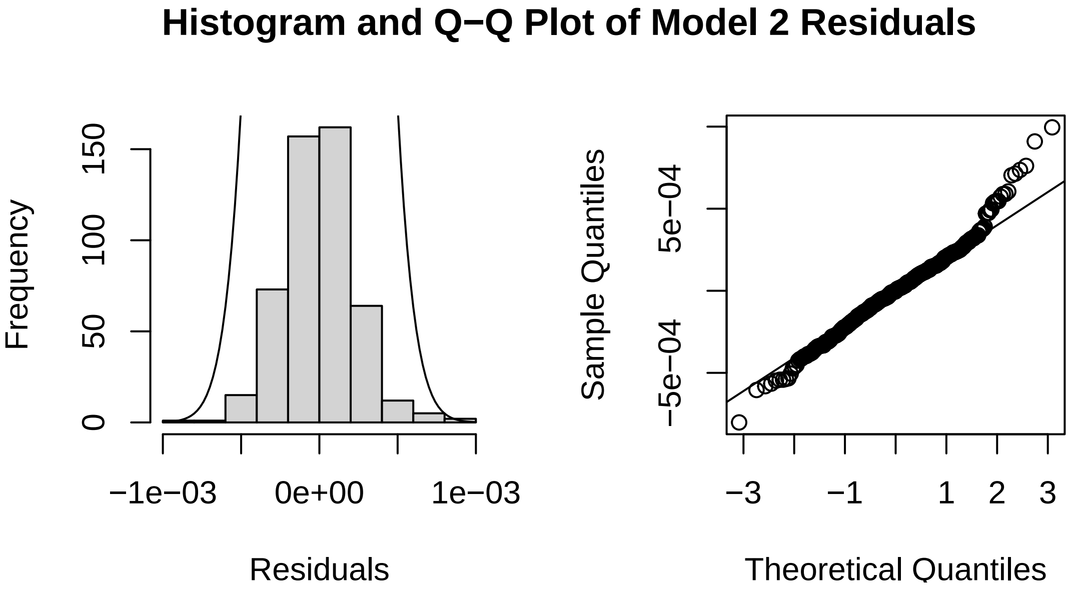
```
##
##  Box-Ljung test
##
## data:  res1
## X-squared = 16.403, df = 16, p-value = 0.4252
```

```
##
##  Box-Ljung test
##
## data:  res1^2
## X-squared = 99.018, df = 22, p-value = 9.573e-12
```

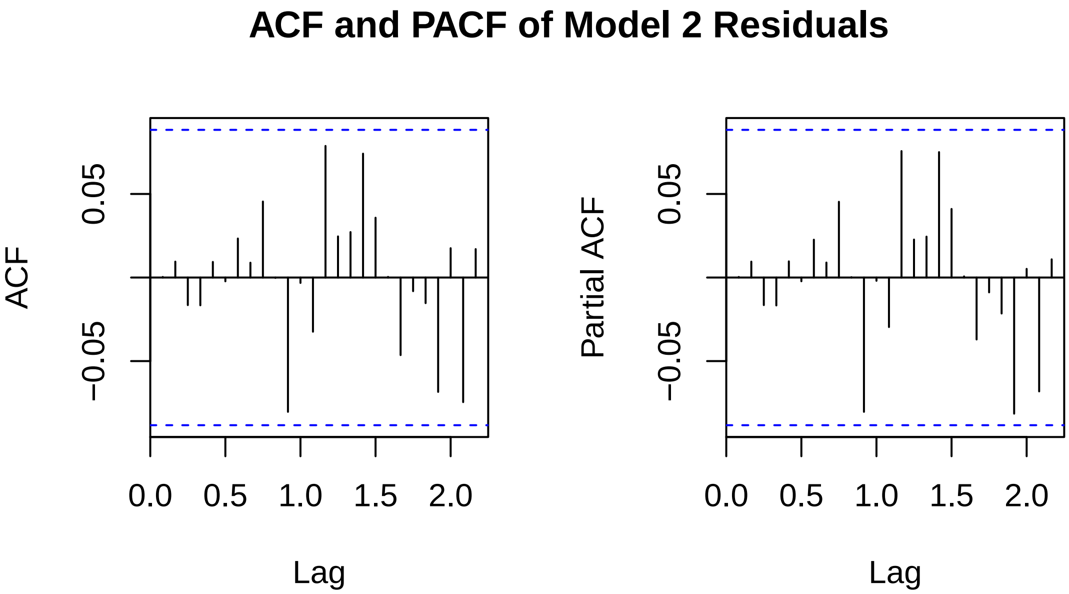
Seeing as the p-values for both the Shapiro and McLeod-Li test are less that 0.05, we can say for certain that the residuals of this model do not follow a white noise process and, in fact, have some sort of nonlinear correlation. We can now do the same for Model 2 to see if it performs any better.

Model 2:





The plots of the residuals for Model 2 appears very similar to those for Model 1, which is not a very promising sign that this model is much better than the last. However, let's continue to see if this is true.



Again, the ACFs and PACFs may be misleading as they do appear to follow a white noise process. But, let's perform the quantitative test to get a sure answer.

```
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.98203, p-value = 9.05e-06
```

```
##
##  Box-Pierce test
##
## data:  res2
## X-squared = 13.635, df = 15, p-value = 0.5533
```

```
##
##  Box-Ljung test
##
## data:  res2
## X-squared = 14.089, df = 15, p-value = 0.5188
```

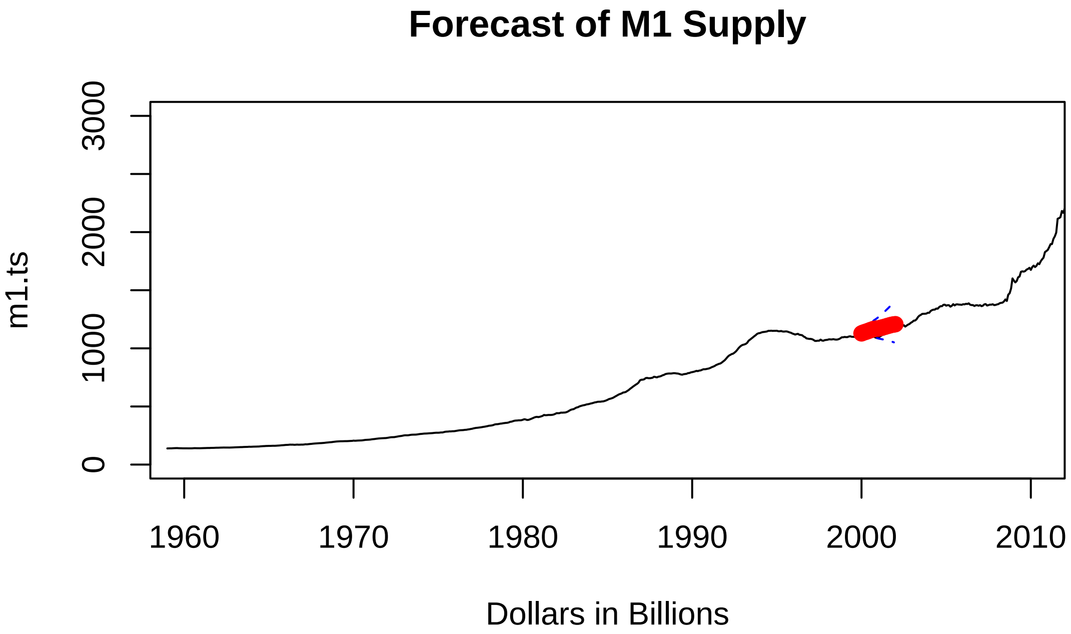
```
##
##  Box-Ljung test
##
## data:  res2^2
## X-squared = 95.159, df = 22, p-value = 4.471e-11
```

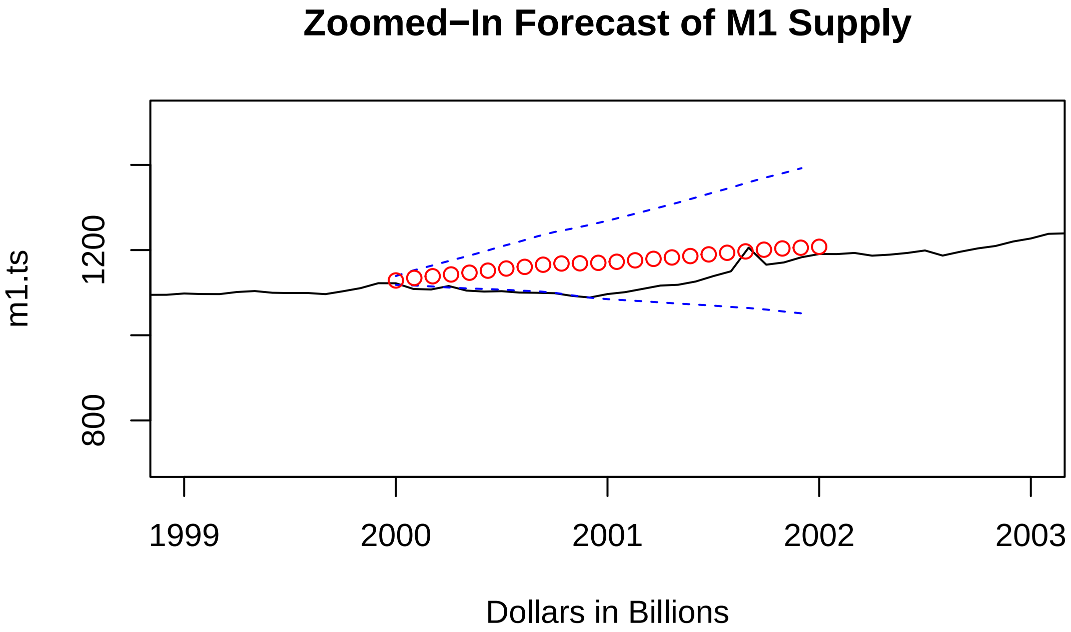
This model, similar to what we mentioned previously, failed the same tests as Model 1, meaning that both of our models have a nonlinear dependence in the residuals.

Our conclusion is that these models include non-Gaussian noise terms, and we choose to use Model 1 for forecasting as both models were very similar to each other, but Model 1 has less parameters.

Forecasting

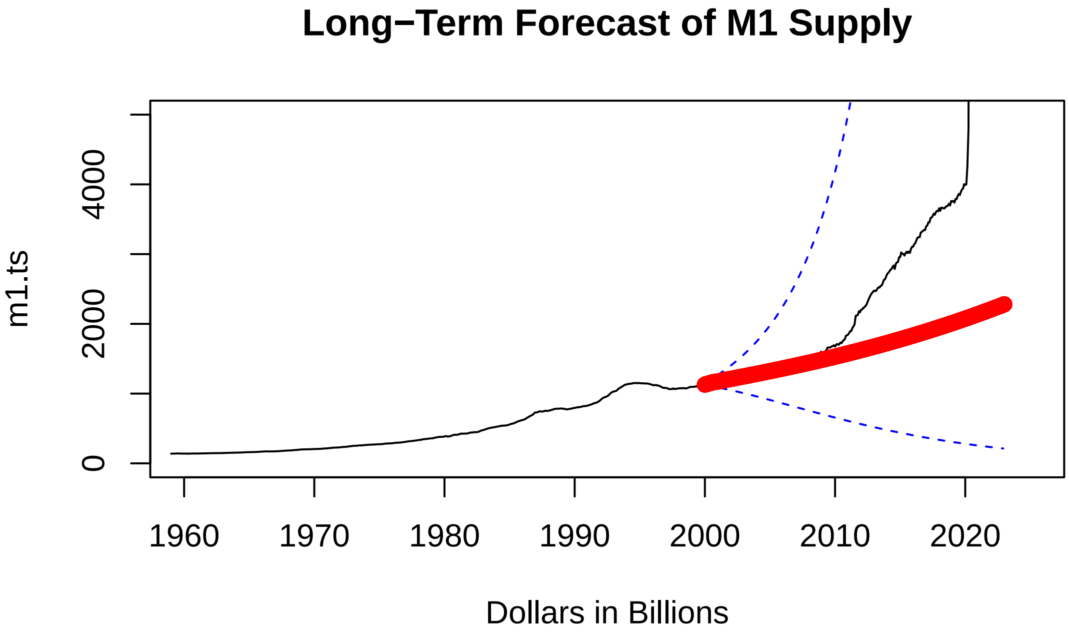
Continuing with Model 1, we first make predictions from the model spanning 24 months of time into the future and form confidence intervals for our forecast. Then, we convert the prediction and intervals back to the scale of the data before we performed our transformations and create our plots to compare our forecasts with the real data in our testing set.

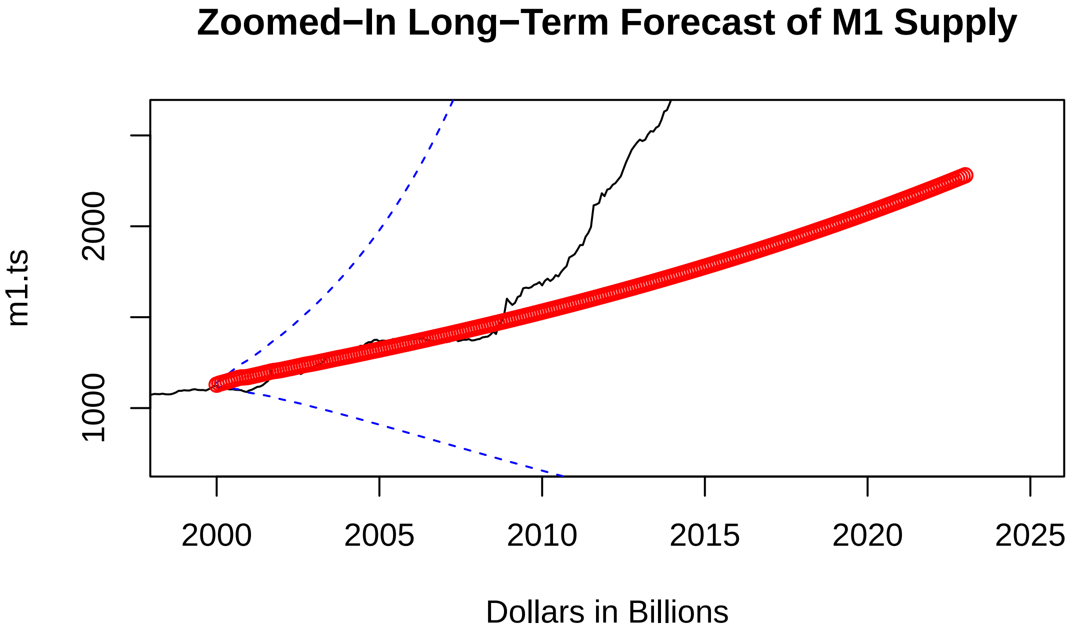




We see that our prediction is fairly accurate. The true values of the M1 supply tend to ride along the lower confidence interval of our predictions, but never stray too far from our forecasted values. As a note, if we had simply predicted values 12 months in advance, our predictions would have never truely matched the test data, but at month 20, we are very close to the actual value.

Mostly for fun, we can also see what happens if we were to attempt to use this model to predict values much further into the future. With the original thought being to see how far off our predictions are from the data of 2020, when the data spikes, seemingly uncontrollably, we will make forecasts 276 months (23 years) into the future.

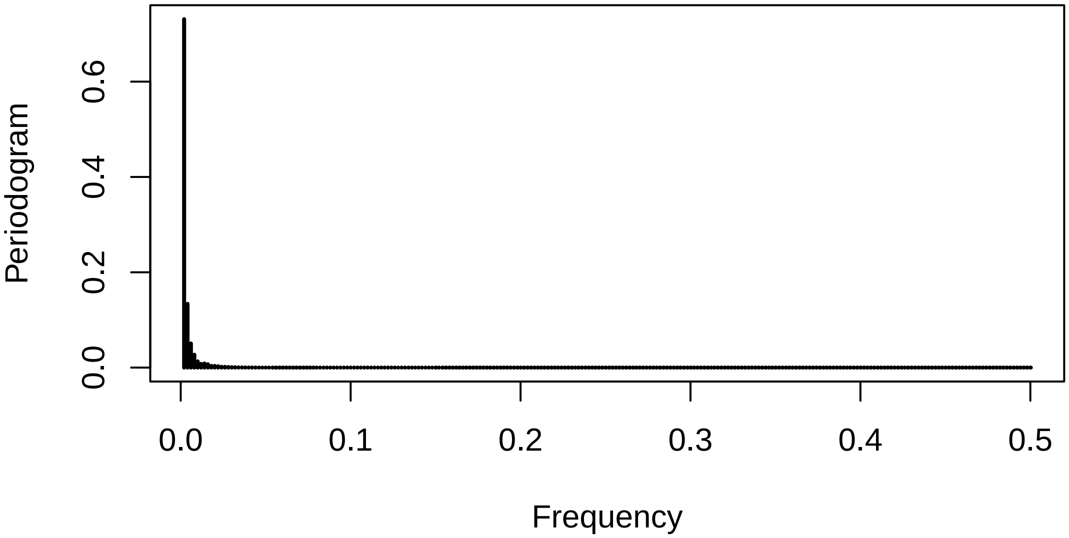




Ideally, we would have had somewhat accurate predictions until about 2020 when the original data exploded, but this is simply not plausible for financial data, especially with such a simply model. Our data actually seems to lose accuracy at around 2008. This is the year that the first major recession of this century took place, which is not something that was originally considered in the project. Looking back at the original data, we can see that there is actually a smaller rise in the trend of the data around this time, but was overshadowed by the large spike in 2020. However, having a even a somewhat accurate prediction 8 years ahead of the training data doesn't seem too bad in this scenario.

Spectral Analysis

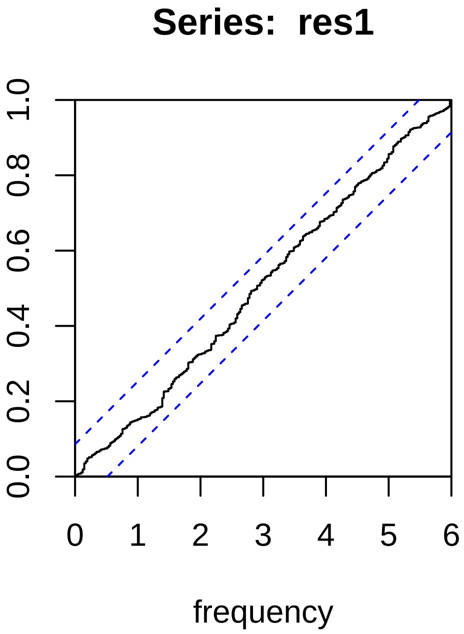
We will now consider the model defined by $X_t = \mu + a\cos(\omega t) + b\sin(\omega t) + Z_t$ where $\{Z_t\} \sim WN(0, \sigma_Z^2)$, in order to determine the periodicity of our transformed data.



The graph of the periodogram shows does not show ant large spikes, and we can hence say that our data does not display a significant periodicity or seasonality. This is consistent with our observation from before. Now, performing the Fisher test, we see

```
## [1] 0.7176285
```

Since $0.7176285 > 0.05$, we cannot reject the null hypothesis that the residuals are white noise from this test alone. Next, we perform the Kolmogorov-Smirnov test on our residuals.



Since the plot stays entirely within the confidence intervals, we see that our residuals also pass this test of normality. However, even though they have passed these two tests, we already know from the Shapiro and McLeod-Li tests that our residuals have some sort of nonlinear dependence between them.

Conclusion

The objective for this report was to forecast values of the M1 monetary aggregate up to 24 months in advance. We fitted our data to a SARIMA$(3,2,3)\times(1,0,1)_{12}$ model given by the algebraic expression

$$(1+0.6377B+0.5652B^2-0.2884B^3)(1-0.4271B^{12})(1-B)^2X_t^{-0.1818182} = (1-0.6824B^3)(1-0.6721B^{12})Z_t.$$

Our main problem in this analysis was that the residuals of this model $\{Z_t\}$ did not follow a Gaussian white noise process. Despite this, we were still able to fairly accurately predict future values of the series up to 24 months in advance. Additionally, we performed spectral analysis that certified some claims we made earlier in the report about the seasonality of our data set.

References

- M1. FRED. (2025, May 27). https://fred.stlouisfed.org/series/M1SL
- Brockwell, P. J., & Davis, R. A. (2016). *Introduction to time series and forecasting*. Springer.
- Feldman, R. (n.d.). Lecture Notes for PSTAT W 174/274. Lecture.

Appendix

Below is a report of the R code used for this project, with comments to direct the reader to which codes were used for each part of the project:

```
# Data Analysis
## Exploration of Data Set
m1_raw <- read.csv("data/m1Monthly.csv")
m1.ts <- ts(m1_raw[ ,2], start=c(1959, 1), end=c(2025, 4), frequency=12)
plot.ts(m1.ts, type='l')
  # Split into training and test sets
train <- window(m1.ts, start=c(1959, 1), end=c(1999, 12), freq=12) # 480 obs
test <- window(m1.ts, start=c(2000, 1), end=c(2001, 12), freq=12) # 24 obs
  # Plot training set
plot.ts(train, type='l')
lines(tslm(train ~ trend)$fitted, col="red") # Linear trend
lines(tslm(train ~ trend + I(trend^2))$fitted, col="blue") # Quadratic trend
var(train)
plot(decompose(train))

  # Box-Cox transformation
index <- 1:length(train)
bcTransform <- boxcox(train ~ index, plotit=TRUE)
lambda <- bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda
train.bc <- train^lambda
#train.bc <- (1/lambda)*(train^lambda - 1)
plot.ts(train.bc)
var(train.bc)
plot(decompose(train.bc))

  # Difference twice at lag 1
train.diff2 <- diff(train.bc, lag=1, differences=2)
index <- 1:length(train.diff2)
plot(train.diff2)
abline(h=mean(train.diff2), col="red")
```

```r
abline(lm(train.diff2 ~ index), col="blue")
var(train.diff2)
m1 <- train.diff2

## Model Selection
  # ACF and PACFS
par(mfrow=c(1, 2))
acf(m1, lag.max=12) # To lag 12
pacf(m1, lag.max=12)
par(mfrow=c(1, 2))
acf(m1, lag.max=60) # To lag 60
pacf(m1, lag.max=60)

  # AICc Selection
    # Model 1
arima(train.bc, order=c(8, 2, 4), seasonal=list(order=c(1, 0, 1),
      period=12), method="ML")
arima(train.bc, order=c(3, 2, 3), seasonal=list(order=c(1, 0, 1),
      period=12), fixed=c(NA, NA, NA, 0, 0, NA, NA, NA), method="ML")
    # Model 2
arima(train.bc, order=c(8, 2, 4),
      method="ML")
arima(train.bc, order=c(7, 2, 4),
      fixed=c(NA, NA, NA, 0, 0, 0, NA, NA, NA, 0, NA), method="ML")
    # Model 3
arima(train.bc, order=c(2, 2, 4), seasonal=list(order=c(1, 0, 1),
      period=12), method="ML")
arima(train.bc, order=c(2, 2, 4), seasonal=list(order=c(1, 0, 1),
      period=12), fixed=c(NA, NA, NA, 0, NA, NA, NA, NA), method="ML")

  # Plotting roots relative to the unit circle
    # Model 1
par(mfrow=c(2, 2))
uc.check(pol_=c(1, 0.6377, 0.5652, -0.2884), print_output=F)
uc.check(pol_=c(1, -0.4271), print_output=F)
uc.check(pol_=c(1, 0, 0, -0.6824), print_output=F)
uc.check(pol_=c(1, -0.6721), print_output=F)
    # Model 2
par(mfrow=c(1, 2))
uc.check(pol_=c(1, 0.3299, 0.7292, 0.4126, 0, 0, 0, 0.0958), print_output=F)
uc.check(pol_=c(1, -0.2561, 0.2826, 0, -0.5742), print_output=F)
    # Model 3
par(mfrow=c(2, 2))
uc.check(pol_=c(1, 0.9654, 0.8407), print_output=F)
uc.check(pol_=c(1, -0.4341), print_output=F)
uc.check(pol_=c(1, 0.3452, 0, -0.6380, -0.2303), print_output=F)
uc.check(pol_=c(1, -0.6606), print_output=F)

## Diagnostic Checking
  # Select Models 1 and 3
model1 <- arima(train.bc, order=c(3, 2, 3), seasonal=list(order=c(1, 0, 1),
                period=12), fixed=c(NA, NA, NA, 0, 0, NA, NA, NA), method="ML")
model3 <- arima(train.bc, order=c(2, 2, 4), seasonal=list(order=c(1, 0, 1),
                period=12), fixed=c(NA, NA, NA, 0, NA, NA, NA, NA), method="ML")
  # Pull residuals
res1 <- model1$residuals
res3 <- model3$residuals

    # Model 1
index <- 1:length(res1)
plot(res1) # Plot Residuals
abline(h=mean(res1), col="red")
abline(lm(res1 ~ index), col="blue")
par(mfrow=c(1, 2)) # Histogram with normal curve and QQ-Plot
hist(res1)
m <- mean(res1)
sd <- sqrt(var(res1))
curve(dnorm(x, m, sd), add=TRUE)
qqnorm(res1)
qqline(res1)
par(mfrow=c(1, 2)) # ACF and PACF
acf(res1)
pacf(res1)
    # Portmanteau Tests
shapiro.test(res1) #Shapiro
Box.test(res1, lag=21, type=c("Box-Pierce"), fitdf=6) # Box-Pierce
Box.test(res1, lag=21, type=c("Ljung-Box"), fitdf=6) # Ljung-Box
Box.test(res1^2, lag=21, type=c("Ljung-Box"), fitdf=0) #McLeod-Li

    # Model 3
index <- 1:length(res3)
plot(res3) # Plot residuals
abline(h=mean(res3), col="red")
abline(lm(res3 ~ index), col="blue")
par(mfrow=c(1, 2)) # Histogram with normal curve and QQ-PLot
hist(res3)
m <- mean(res3)
sd <- sqrt(var(res3))
curve(dnorm(x, m, sd), add=TRUE)
qqnorm(res3)
qqline(res3)
par(mfrow=c(1, 2)) # ACF and PACF
acf(res3)
pacf(res3)
      # Portmanteau Tests
shapiro.test(res3) # Shapiro
Box.test(res3, lag=21, type=c("Box-Pierce"), fitdf=7) # Box-Pierce
Box.test(res3, lag=21, type=c("Ljung-Box"), fitdf=7) # Ljung-Box
Box.test(res3^2, lag=21, type=c("Ljung-Box"), fitdf=0) # McLeod-Li

## Forecasting
  # 2-Year Prediction
pred.tr <- predict(model1, n.ahead=24)
U.tr <- pred.tr$pred + 2*pred.tr$se
L.tr <- pred.tr$pred - 2*pred.tr$se
    # Convert to original scale
```

```r
pred.orig <- pred.tr$pred^(1/lambda)
U = U.tr^(1/lambda)
L = L.tr^(1/lambda)
plot.ts(m1.ts, xlim=c(1960, 2010), ylim=c(0, 3000))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points(seq(1960+length(train.bc)/12-1, 1960+length(train.bc)/12+1,
           length.out=length(pred.tr$pred)), pred.orig, col="red")
    # Zoomed-in plot
plot.ts(m1.ts, xlim=c(1999, 2003), ylim=c(700, max(U)+400))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points(seq(1960+length(train.bc)/12-1, 1960+length(train.bc)/12+1,
           length.out=length(pred.tr$pred)), pred.orig, col="red")
  # 23-Year Prediction
pred.tr <- predict(model1, n.ahead=276)
U.tr <- pred.tr$pred + 2*pred.tr$se
L.tr <- pred.tr$pred - 2*pred.tr$se
    # Convert to original scale
pred.orig <- pred.tr$pred^(1/lambda)
U = U.tr^(1/lambda)
L = L.tr^(1/lambda)
plot.ts(m1.ts, xlim=c(1960, 2025), ylim=c(0, 5000))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points(seq(1960+length(train.bc)/12-1, 1960+length(train.bc)/12+22,
           length.out=length(pred.tr$pred)), pred.orig, col="red")
    # Zoomed-in plot
plot.ts(m1.ts, xlim=c(1999, 2025), ylim=c(700, max(U)+1500))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points(seq(1960+length(train.bc)/12-1, 1960+length(train.bc)/12+22,
           length.out=length(pred.tr$pred)), pred.orig, col="red")

## Spectral Analysis
TSA::periodogram(m1.ts)
fisher.g.test(res1)
cpgram(res1)
```