Time Series Project: U.S. Real GDP Forecast

Ryan Yee

5/29/2023

Contents

Abstract:

In this time series project, the goal was to forecast U.S. real GDP using quarterly data from 2002 to 2023. The project aims to identify key trends and patterns in U.S. real GDP over this period. Various time series techniques such as ARIMA, SARIMA, and data transformation were applied to the historical U.S. real GDP data. The data were analyzed for trends, seasonality, and other patterns. The selected forecasting models were evaluated based on their accuracy and performance using the 12 data points from the test sample.

The key results of the project reveal the important trends and patterns in U.S. real GDP, such as long-term growth, business cycles, and seasonality. The forecasting models demonstrated their effectiveness in capturing and predicting these patterns. The project provides valuable insights into the future trajectory of U.S. real GDP and offers guidance for policymakers and decision-makers.

In conclusion, the model SARIMA (p = 1, d = 1, q = 0) × (P = 0, D = 1, Q = 1)s=4 was determined to be the most appropriate and accurate forecasting model that captures trends and patterns in U.S. real GDP from 2002 to 2023. However, there were multiple models that present fairly accurate forecasts. Suggesting that more than one model may be appropriate in forecasting U.S. Real GDP. It is not clear which model best captures the trend and trajectory. This is reasonable since the original data is not Gaussian due to volatility during the Great Recession (December 2007 – June 2009) and the Pandemic (April 2020). The non-Gaussian like behavior may cause the forecasting values to shift up and down depending on when the volatility occurred in historical data. For example, the forecast produced by our final model may appear to be overestimating. This is because there was a recent crash in economy that decreases U.S. Real GDP drastically and is slowly recovering.

Introduction:

The objective of this time series project is to forecast U.S. real GDP using quarterly data from 2002 to 2023. Real GDP is a crucial indicator of economic performance and plays a significant role in informing policy decisions, assessing economic health, and guiding investment strategies. Understanding the trends, patterns, and future trajectory of U.S. real GDP is of great importance for policymakers, economists, and businesses alike.

The dataset used in this project consists of quarterly U.S. real GDP data spanning a period of 22 years. Real GDP represents the inflation-adjusted value of all final goods and services produced within the United States, providing a measure of the country's economic output.

The primary problem addressed in this project is forecasting U.S. real GDP. The goal is to develop accurate and reliable models that can capture the underlying trends, seasonality, and other dynamics in the data and provide forecasts for future quarters. To achieve this, various time series techniques such as SARIMA (Seasonal Auto Regressive Integrated Moving Average), and data transformation.

The results of this project will provide insights into the historical behavior of U.S. real GDP and shed light on its future trajectory. The accuracy and performance of the forecasting models will be evaluated based on transformed data variances, ACFs and PACFs of residuals, and model diagnostics. Positive results will indicate the efficacy of the selected models in capturing and predicting the dynamics of U.S. real GDP. Negative results, if any, will also be discussed, offering insights into potential challenges or limitations of the forecasting process.

The data for this project was obtained from a reliable and reputable source, the U.S. Bureau of Economic Analysis (BEA) and the Federal Reserve Economic Data (FRED) database. The specific source will be acknowledged, ensuring transparency and credibility. The software used for data analysis, modeling, and forecasting include R and its specialized time series analysis packages like "astsa", "tsdl", "MASS", "ggplot2", "ggfortify", "qpcR" and "forecast".

Data Source Details:

Source: U.S. Bureau of Economic Analysis

Release: Gross Domestic Product

Units: Billions of Chained 2012 Dollars, Not Seasonally Adjusted

Frequency: Quarterly

BEA Account Code: ND000334

U.S. Bureau of Economic Analysis, Real Gross Domestic Product [ND000334Q], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/ND000334Q, May 30, 2023.

```r
gdp.csv <- read.table("data/ND000334Q.csv", sep = ",", header = FALSE, skip = 1, nrows = 85)
head(gdp.csv)
```

V1	V2
2002-01-01	3263.869
2002-04-01	3362.508
2002-07-01	3401.820
2002-10-01	3460.159
2003-01-01	3340.163
2003-04-01	3429.079

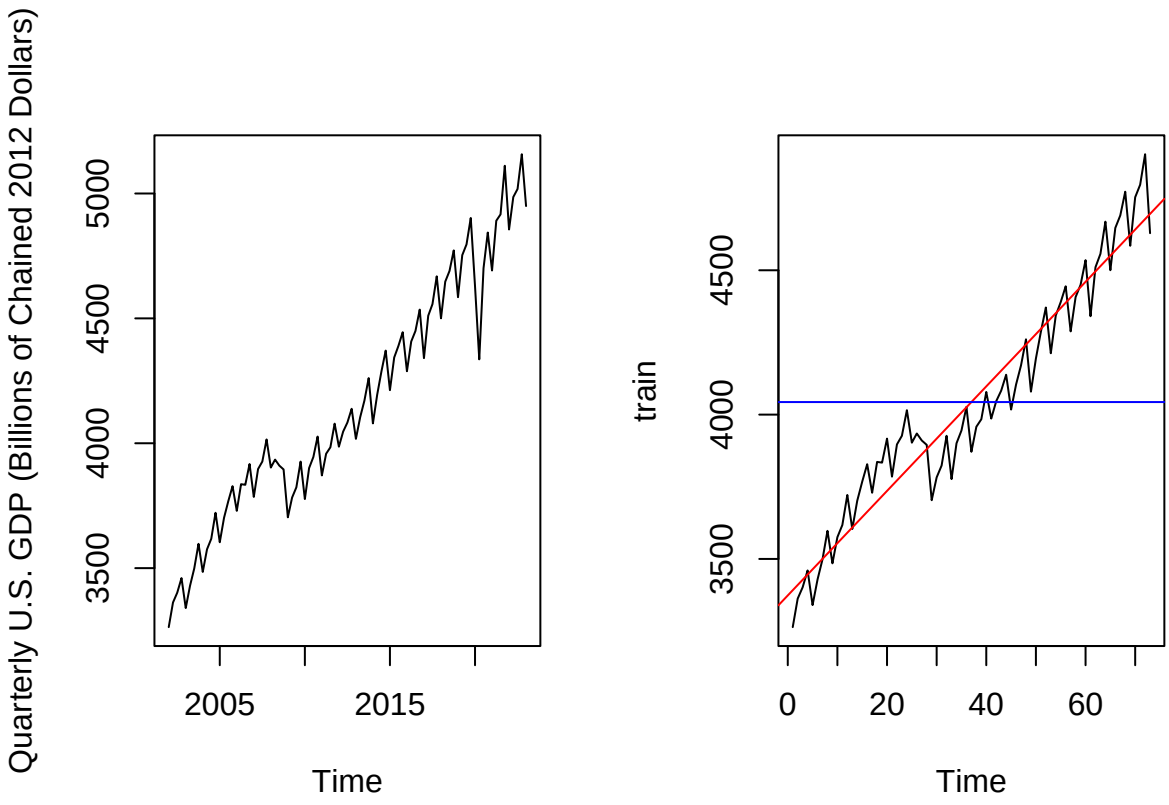
```r
tail(gdp.csv)
```

	V1	V2
80	2021-10-01	5110.951
81	2022-01-01	4855.857
82	2022-04-01	4985.795
83	2022-07-01	5018.093
84	2022-10-01	5157.178
85	2023-01-01	4949.655

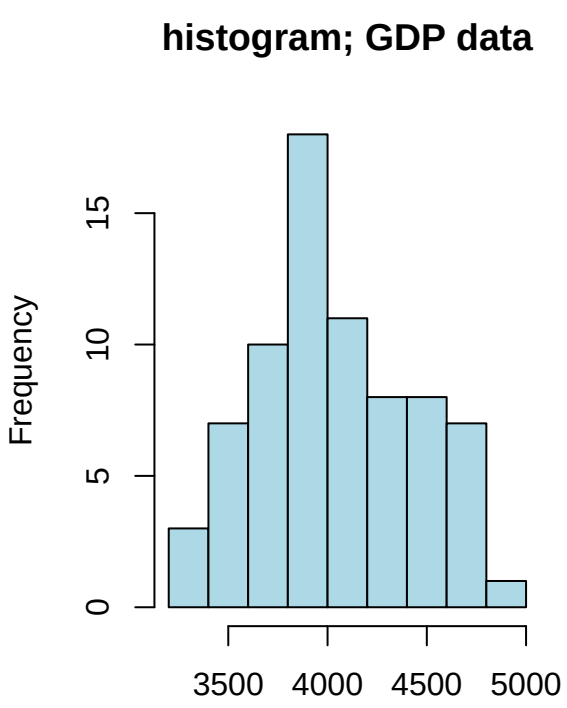
DISCLAIMER: The following data set is not Gaussian distributed due to volatility during the Great Recession (December 2007 – June 2009) and the Pandemic (April 2020).

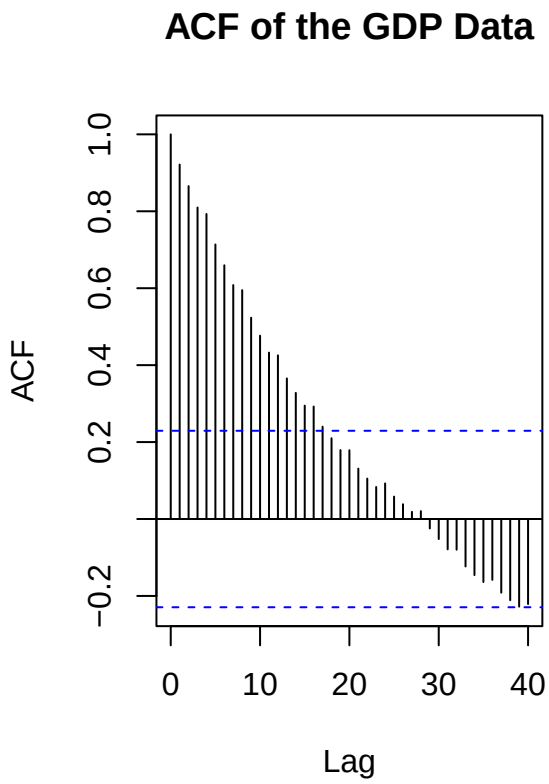
```
gdp <- ts(gdp.csv[, 2], start = c(2002, 1), frequency = 4)
train <- gdp[1:73]
test <- gdp[74:85]
par(mfrow = c(1, 2))
ts.plot(gdp, ylab = "Quarterly U.S. GDP (Billions of Chained 2012 Dollars)")
plot.ts(train)

fit <- lm(train ~ as.numeric(1:length(train)))
abline(fit, col = "red")
abline(h = mean(train), col = "blue")
```




```
hist(train, col = "light blue", xlab = "", main = "histogram; GDP data")
acf(train, lag.max = 40, main = "ACF of the GDP Data")
```

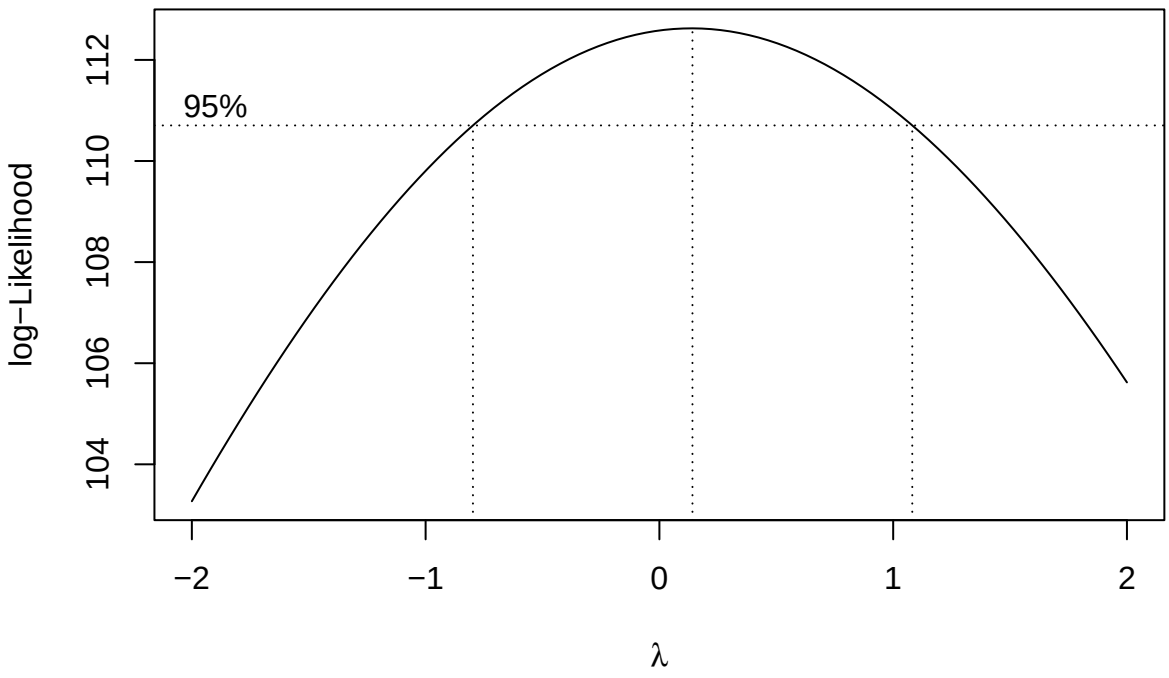




Histogram appear to be symmetric or bell-shaped (However, it is not Gaussian). One might argue it is a little bit skewed right. ACF remains large.

Compare transformations

```
bcTransform <- boxcox(train ~ as.numeric(1:length(train)))
```

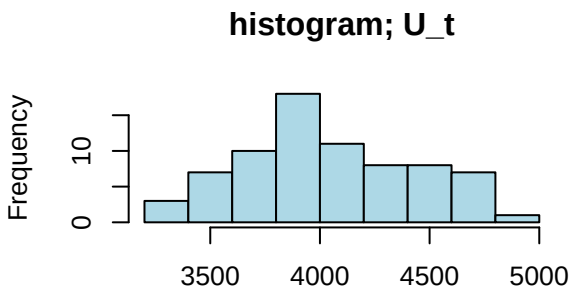


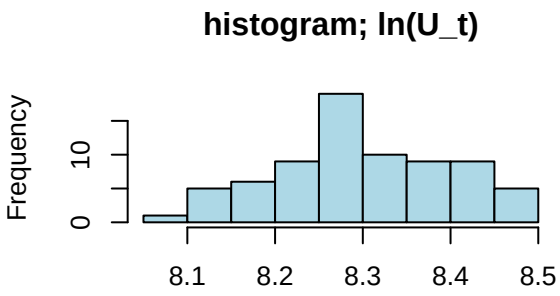

```
bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
```

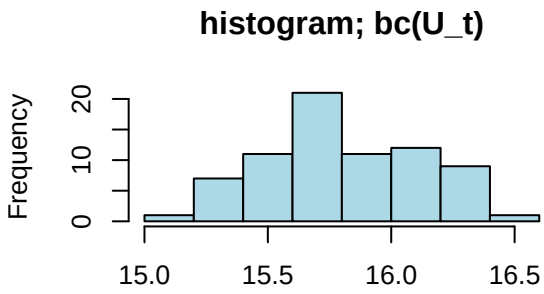
```
## [1] 0.1414141
```

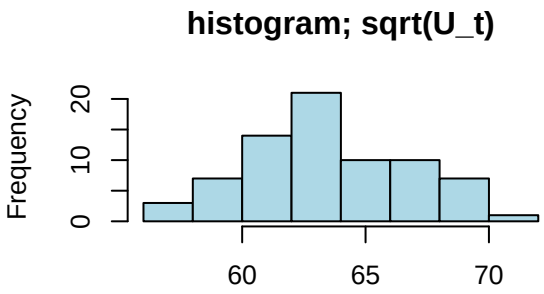
```
lambda <- bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
train.bc <- (1 / lambda) * (train^lambda - 1)
test.bc <- (1 / lambda) * (test^lambda - 1)
train.log <- log(train)
train.sqrt <- sqrt(train)

op <- par(mfrow = c(2, 2))
hist(train, col = "light blue", xlab = "", main = "histogram; U_t")
hist(train.log, col = "light blue", xlab = "", main = "histogram; ln(U_t)")
hist(train.bc, col = "light blue", xlab = "", main = "histogram; bc(U_t)")
hist(train.sqrt, col = "light blue", xlab = "", main = "histogram; sqrt(U_t)")
```

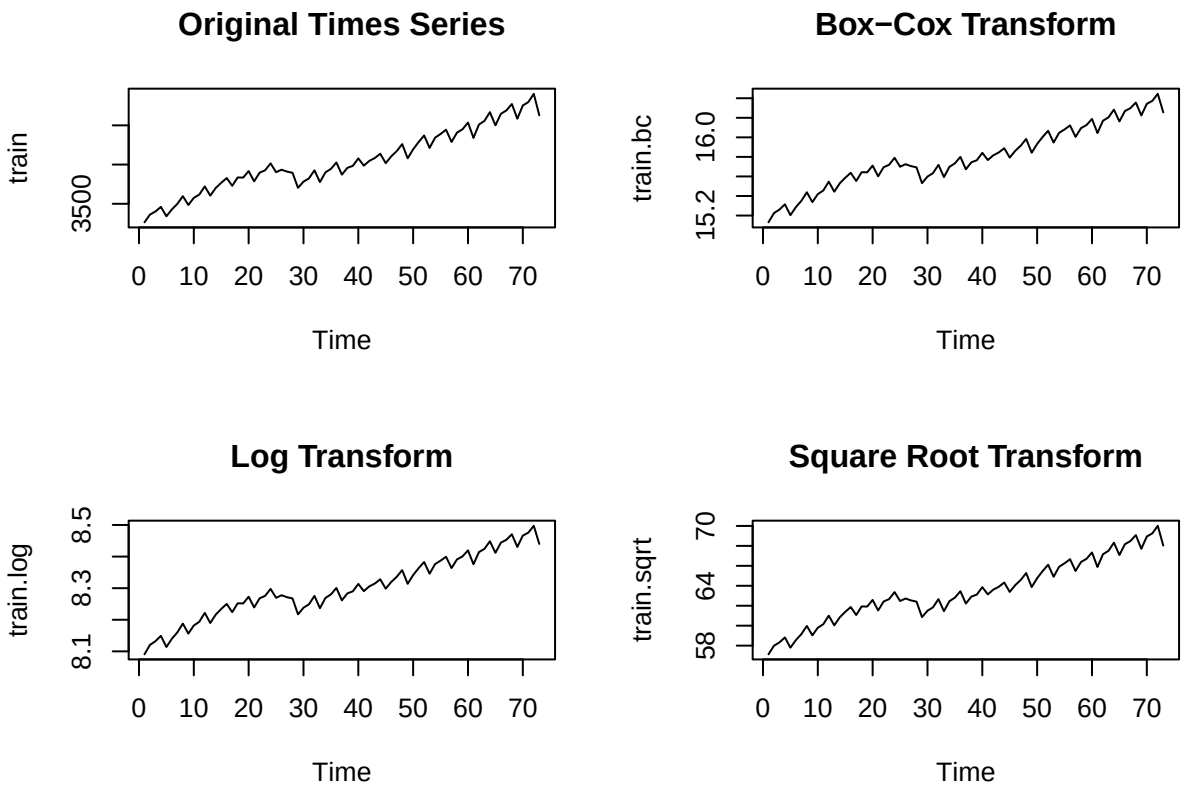








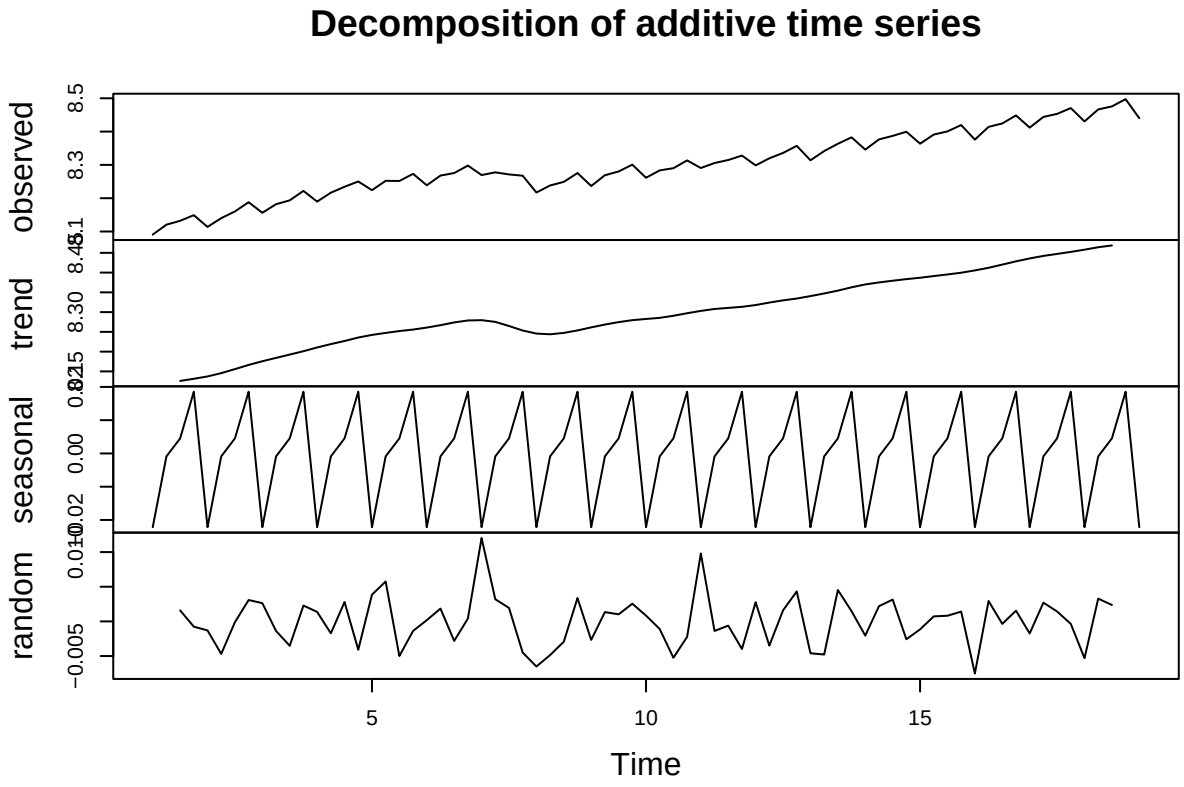

```r
# Compare transforms
op <- par(mfrow = c(2, 2))
ts.plot(train, main = "Original Times Series")
ts.plot(train.bc, main = "Box-Cox Transform")
ts.plot(train.log, main = "Log Transform")
ts.plot(train.sqrt, main = "Square Root Transform")
```



There isn't much difference between transformations. One might argue that a transformation is unnecessary. Choose log(U_t) because it is more symmetric and seems to have more even variance.

Decomposition of $ln(U_t)$ shows seasonality and almost linear trend

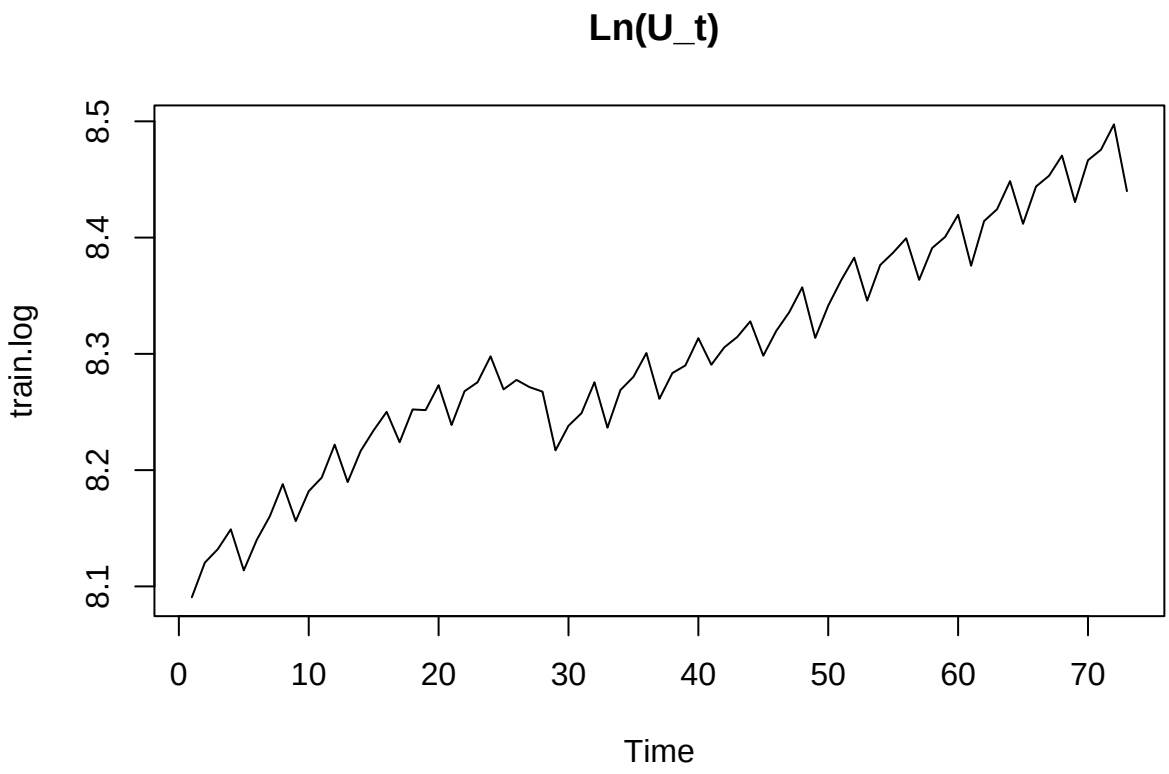
```
y <- ts(as.ts(train.log), frequency = 4)
decomp <- decompose(y)
plot(decomp)
```



```
var(train.log)
```

```
## [1] 0.009664501
```

```
plot.ts(train.log, main = "Ln(U_t) ")
```



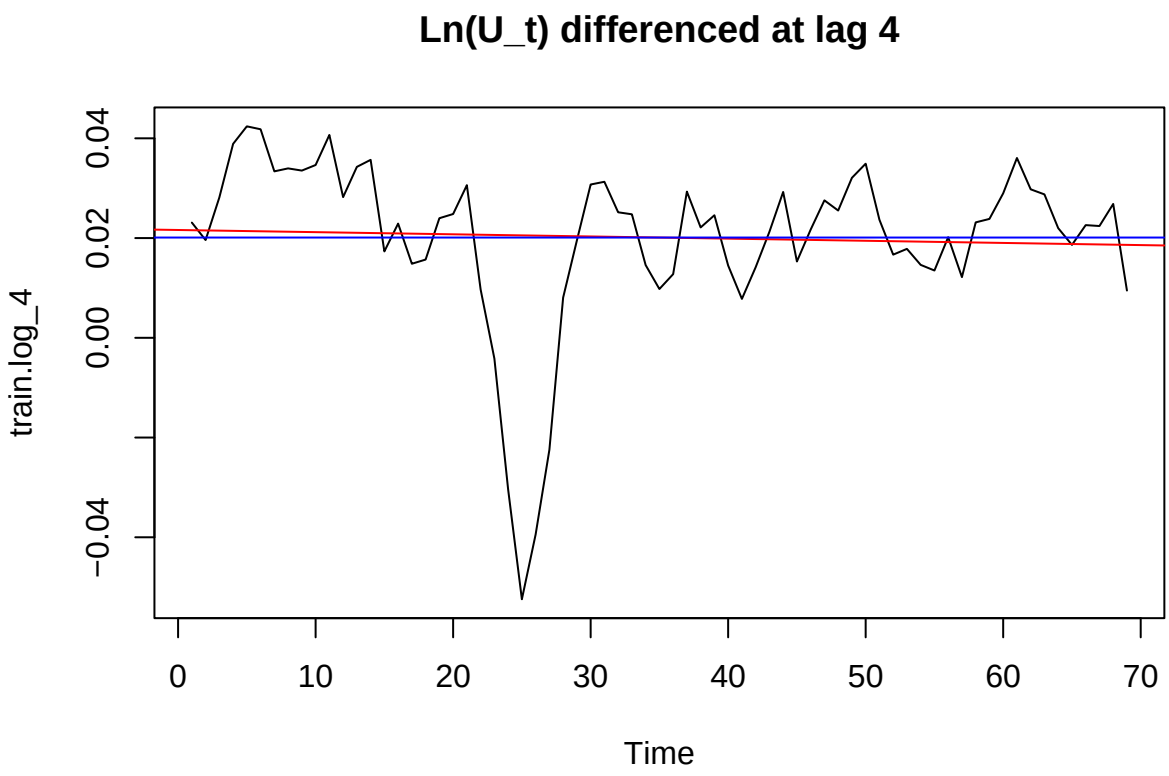

```
train.log_4 <- diff(train.log, lag = 4)
plot.ts(train.log_4, main = "Ln(U_t) differenced at lag 4")
var(train.log_4)
```

```
## [1] 0.0002872528
```

```
fit <- lm(train.log_4 ~ as.numeric(1:length(train.log_4)))
abline(fit, col = "red")
mean(train.log_4)
```

```
## [1] 0.02010507
```

```
abline(h = mean(train.log_4), col = "blue")
```



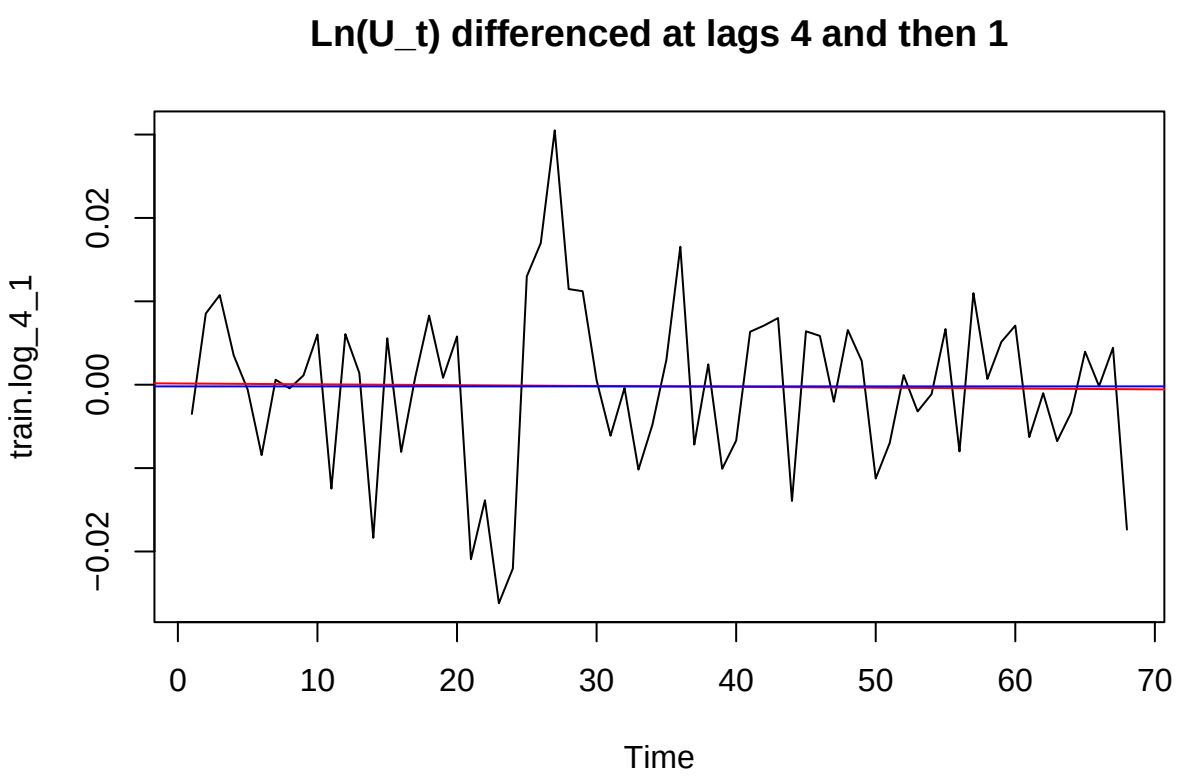

```
train.log_4_1 <- diff(train.log_4, lag = 1)
plot.ts(train.log_4_1, main = "Ln(U_t) differenced at lags 4 and then 1")
var(train.log_4_1)
```

```
## [1] 9.788039e-05
```

```
fit <- lm(train.log_4_1 ~ as.numeric(1:length(train.log_4_1)))
abline(fit, col = "red")
mean(train.log_4_1)
```

```
## [1] -0.0002004654
```

```
abline(h = mean(train.log_4_1), col = "blue")
```



Plot of $ln(U_t)$

- Seasonality
- Trend
- Variance: 0.009665

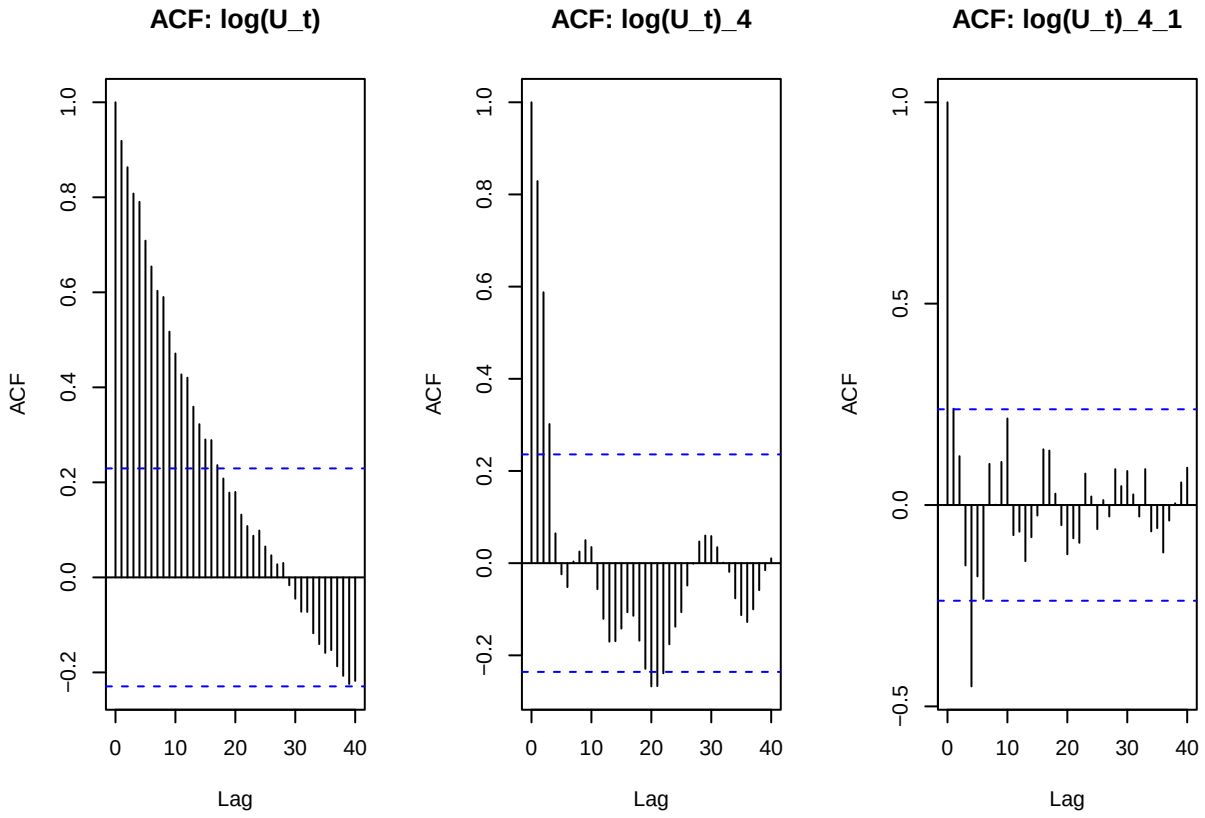
Plot of $ln(U_t)$ differenced at lag 4

- Seasonality no longer apparent
- Variance: 0.0002873 – lower!
- Trend is still here

Plot of $ln(U_t)$ differenced at lags 4 and then 1

- No Seasonality
- Variance: 9.788e-05 – even lower!
- No trend
- Data looks stationary, but check ACFs

```
par(mfrow = c(1, 3))
acf(train.log, lag.max = 40, main = "ACF: log(U_t)")
acf(train.log_4, lag.max = 40, main = "ACF: log(U_t)_4")
acf(train.log_4_1, lag.max = 40, main = "ACF: log(U_t)_4_1")
```



Plot of ACF of $ln(U_t)$

- Slows decay indicated non-stationarity
- One sees seasonality

Plot of ACF of $ln(U_t)$ differenced at lag 4

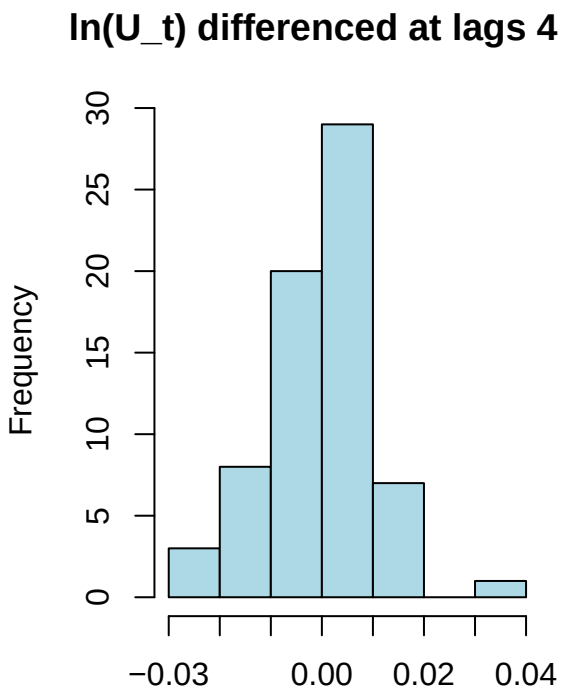
- Seasonality no longer apparent
- ACF decays slowly indicating non-stationarity

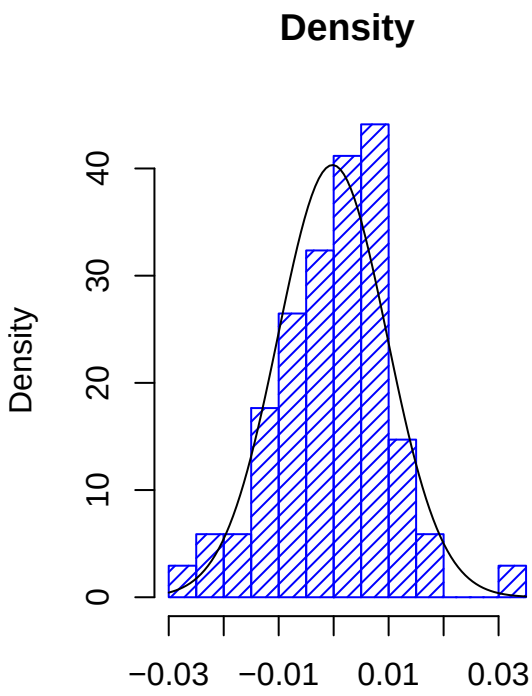
Plot of ACF of $ln(U_t)$ differenced at lags 4 & 1

- ACF decay corresponds to a stationary process Conclude: Work with data $ln(U_t)$ differenced at lags 4 & 1, U_t = the first 73 observations of the original data.

Histogram of $\nabla 1 \nabla 4 ln(U_t)$ looks symmetric and almost Gaussian.

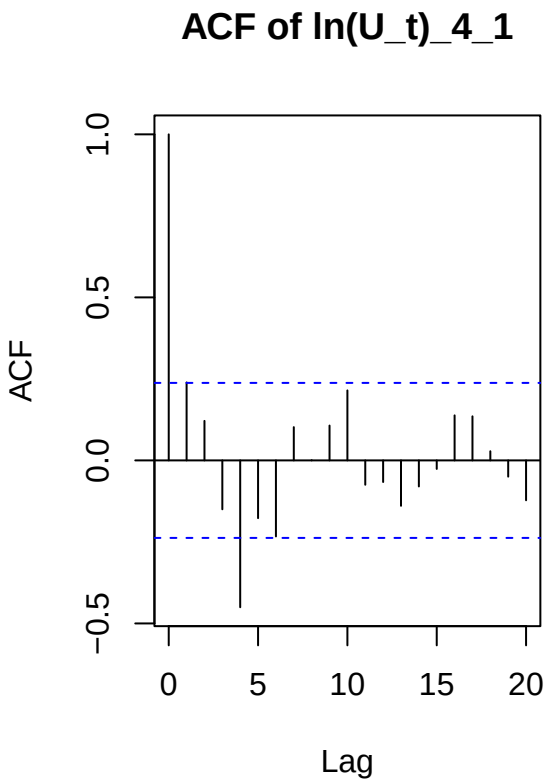
```
par(mfrow = c(1, 2))
hist(train.log_4_1, col = "light blue", xlab = "", main = "ln(U_t) differenced at lags 4 & 1")
hist(train.log_4_1, density = 20, breaks = 20, col = "blue", xlab = "", main = "Density", prob = TRUE)
m <- mean(train.log_4_1)
std <- sqrt(var(train.log_4_1))
curve(dnorm(x, m, std), add = TRUE)
```

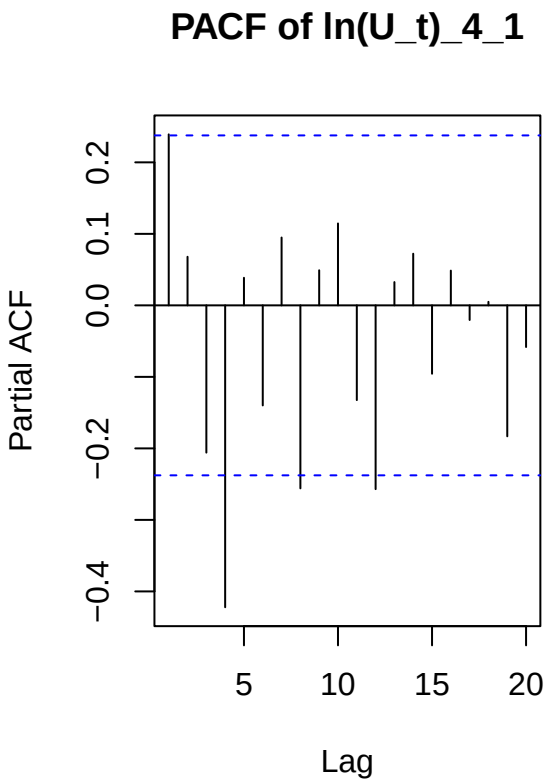




ACF and PACF of train.log_4_1 = $\nabla 1 \nabla 4 ln(U_t)$,

```
par(mfrow = c(1, 2))
acf(train.log_4_1, lag.max = 20, main = "ACF of ln(U_t)_4_1")
pacf(train.log_4_1, lag.max = 20, main = "PACF of ln(U_t)_4_1")
```





Determine possible candidate models SARIMA$(p, d, q) \times (P, D, Q)_s$ for the series $ln(U_t)$.

Modeling the seasonal part (P, D, Q): For this part, focus on the seasonal lags h = 1s, 2s, etc.

- We applied one seasonal differencing so D = 1 at lag s = 4.
- The ACF shows a strong peak at h = 4s.

A good choice for the SMA part could be Q = 0, Q = 1

- The PACF shows a peak at h = 4s, 8s and 12s. A good choice for the SAR part could be P = 0, P = 1, P = 3.

Modeling the non-seasonal part (p , d, q): In this case focus on the within season lags, h = 1,. . . ,11.

- We applied one differencing to remove the trend: d = 1
- The ACF seems to be tailing off. Or perhaps cuts off at lag 4.

A good choice for the MA part could be q = 0, 1.

- The PACF cuts off at lag h=1 or 4.

A good choice for the AR part could be p = 0, 1.

As an illustration we fit the following model:

SARIMA (p = 1, d = 1, q = 1) × (P = 1, D = 1, Q = 1)s=4

SARIMA (p = 0, d = 1, q = 1) × (P = 1, D = 1, Q = 1)s=4

SARIMA $(p = 0, d = 1, q = 0) \times (P = 0, D = 1, Q = 1)s{=}4$

SARIMA $(p = 1, d = 1, q = 0) \times (P = 0, D = 1, Q = 1)s{=}4$

Evaluate candidate models by comparing their AICc.

```
arima(train.log, order = c(4, 1, 4), seasonal = list(order = c(4, 1, 4), period = 4), method = "ML")
```

```
## Warning in arima(train.log, order = c(4, 1, 4), seasonal = list(order = c(4, :
## possible convergence problem: optim gave code = 1
```

```
##
## Call:
## arima(x = train.log, order = c(4, 1, 4), seasonal = list(order = c(4, 1, 4),
##     period = 4), method = "ML")
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##           ar1     ar2     ar3      ar4     ma1      ma2      ma3     ma4
##       -0.1871  0.5162  0.0915  -0.8161  0.4565  -0.3199  -0.2425  0.4751
## s.e.   0.1592  0.2002  0.1429   0.1303  0.2256   0.2830   0.2411  0.2109
##          sar1     sar2    sar3    sar4  sma1     sma2     sma3   sma4
##       -0.4343  -0.2269  0.3206  0.0915  0.05  -0.3344  -0.7050  0.204
## s.e.      NaN      NaN     NaN     NaN   NaN      NaN   0.4045    NaN
##
## sigma^2 estimated as 4.662e-05:  log likelihood = 237.54,  aic = -441.09
```

```
AICc(arima(train.log, order = c(4, 1, 4), seasonal = list(order = c(4, 1, 4), period = 4), method = "ML
```

```
## Warning in arima(train.log, order = c(4, 1, 4), seasonal = list(order = c(4, :
## possible convergence problem: optim gave code = 1
```

```
## [1] -431.3722
```

```
arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML")
```

```
##
## Call:
## arima(x = train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1),
##     period = 4), method = "ML")
##
## Coefficients:
##          ar1      ma1    sar1     sma1
##       0.4739  -0.2655  0.0334  -0.8102
## s.e.  0.3534   0.3747  0.1762   0.1477
##
## sigma^2 estimated as 5.746e-05:  log likelihood = 233.48,  aic = -456.96
```

```
AICc(arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML
```

```
## [1] -456.3734
```

```
arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 4), fixed = c(NA, 0, 
```

```
## Warning in arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
##
## Call:
## arima(x = train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1),
##     period = 4), fixed = c(NA, 0, 0, NA), method = "ML")
##
## Coefficients:
##          ar1  ma1  sar1     sma1
##       0.2132    0     0  -0.7857
## s.e.  0.1246    0     0   0.1197
##
## sigma^2 estimated as 5.808e-05:  log likelihood = 233.2,  aic = -460.39
```

```
AICc(arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 4), fixed = c(NA
```

```
## Warning in arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
## [1] -459.8038
```

```
arima(train.log, order = c(1, 1, 0), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML")
```

```
##
## Call:
## arima(x = train.log, order = c(1, 1, 0), seasonal = list(order = c(0, 1, 1),
##     period = 4), method = "ML")
##
## Coefficients:
##          ar1     sma1
##       0.2132  -0.7857
## s.e.  0.1246   0.1197
##
## sigma^2 estimated as 5.808e-05:  log likelihood = 233.2,  aic = -460.39
```

```
AICc(arima(train.log, order = c(1, 1, 0), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML
```

```
## [1] -460.2206
```

```
arima(train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML")
```

```
##
## Call:
## arima(x = train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1),
##     period = 4), method = "ML")
##
## Coefficients:
##          ma1     ma2      ma3     ma4     sar1     sma1
##       0.1984  0.2131  -0.0709  0.8025  -0.6712  -0.7937
## s.e.  0.0990  0.0906   0.1167  0.1242   0.1412   0.1261
##
## sigma^2 estimated as 5.009e-05:  log likelihood = 236.7,  aic = -459.39
```

```
AICc(arima(train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML
```

```
## [1] -458.1217
```

```
arima(train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1), period = 4), fixed = c(NA, NA,
```

```
##
## Call:
## arima(x = train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1),
##     period = 4), fixed = c(NA, NA, 0, NA, NA, NA), method = "ML")
##
## Coefficients:
##          ma1     ma2  ma3     ma4     sar1     sma1
##       0.1811  0.2353    0  0.8422  -0.6987  -0.7934
## s.e.  0.0826  0.0782    0  0.1030   0.1465   0.1237
##
## sigma^2 estimated as 5.1e-05:  log likelihood = 236.48,  aic = -460.96
```

```
AICc(arima(train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML
```

```
## [1] -458.1217
```

```
arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML")
```

```
##
## Call:
## arima(x = train.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1),
##     period = 4), method = "ML")
##
## Coefficients:
##          ar1      ma1     sma1
##       0.4934  -0.2857  -0.7922
## s.e.  0.3434   0.3679   0.1163
##
## sigma^2 estimated as 5.759e-05:  log likelihood = 233.46,  aic = -458.93
```

```
AICc(arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML
```

```
## [1] -458.5779
```

```r
arima(train.log, order = c(0, 1, 0), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML")
```

```
##
## Call:
## arima(x = train.log, order = c(0, 1, 0), seasonal = list(order = c(0, 1, 1),
##     period = 4), method = "ML")
##
## Coefficients:
##          sma1
##       -0.8046
## s.e.   0.1237
##
## sigma^2 estimated as 6.032e-05:  log likelihood = 231.77,  aic = -459.53
```

```r
AICc(arima(train.log, order = c(0, 1, 0), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML
```

```
## [1] -459.4738
```

Conclusion:

Do not choose SARIMA (p = 4, d = 1, q = 4) × (P = 4, D = 1, Q = 4)s=4 because it has too many parameters and will not pass the model diagnostic.

Model B: SARIMA (p = 1, d = 1, q = 1) × (P = 1, D = 1, Q = 1)s=4 has second-lowest AICc of -459.8.

model A: SARIMA (p = 1, d = 1, q = 0) × (P = 0, D = 1, Q = 1)s=4 has the lowest AICc of -460.2.

Model B: $(1 + 0.474_{(0.353)}B)(1 - 0.033_{(0.176)}B^4)Y_t = (1 - 0.265_{(0.375)}B)(1 - 0.810_{(0.148)}B^4)Z_t$, where $\hat{\sigma}_z^2 = 0.0000575$.

Model A: $(1 - 0.213_{(0.125)}B)Y_t = (1 - 0.786_{(0.120)}B^4)Z_t$, where $\hat{\sigma}_z^2 = 0.0000581$.

Check invertibility of model B: SARIMA (p = 1, d = 1, q = 1) × (P = 1, D = 1, Q = 1)s=4

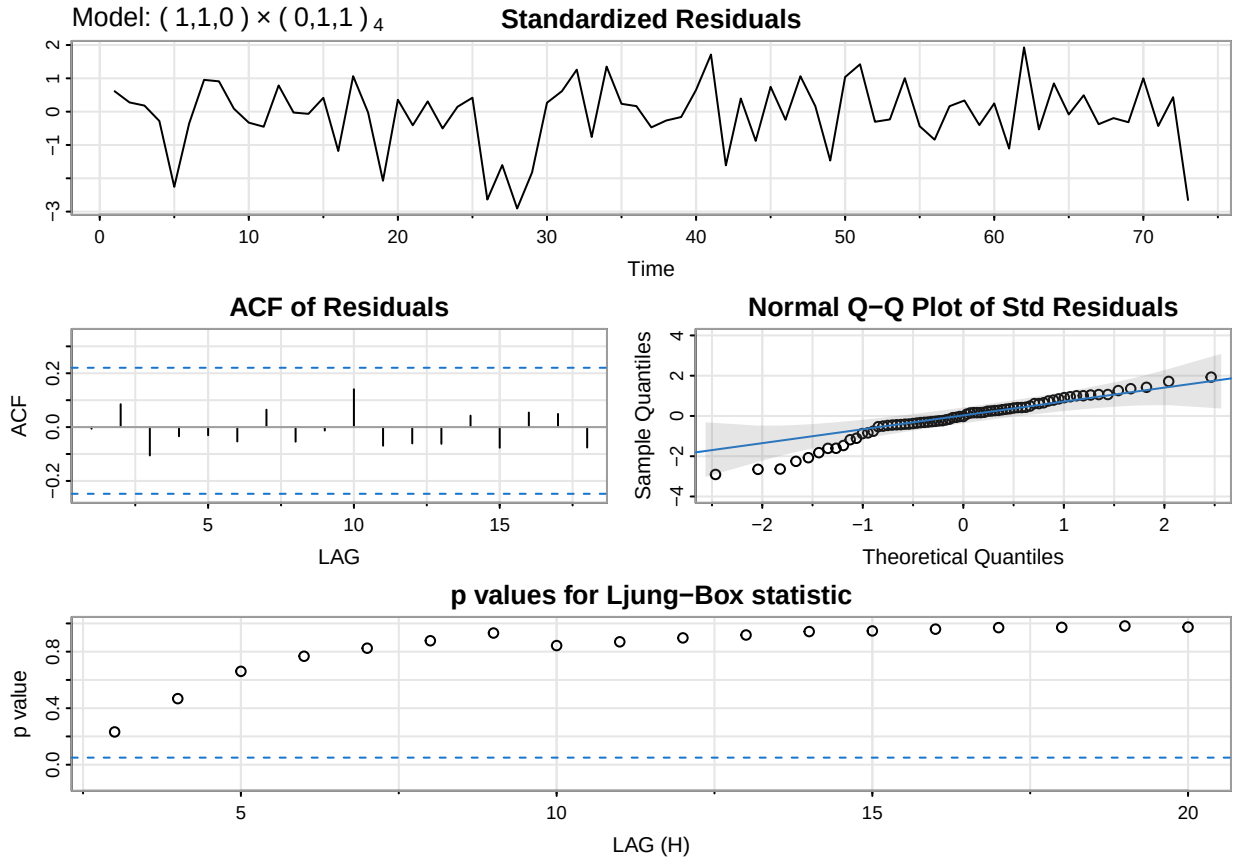
- ma = -0.265, sma = -0.810
- Model (B) is invertible because $|\theta_1| < 1; |\Theta_1 < 1|$

Check invertibility of model A: SARIMA (p = 1, d = 1, q = 0) × (P = 0, D = 1, Q = 1)s=4

- it has no MA component so Model (A) is invertible.

Since model A has lower AICc and less parameters than model B.

Choose model A: SARIMA (p = 1, d = 1, q = 0) × (P = 0, D = 1, Q = 1)s=4 as our final model



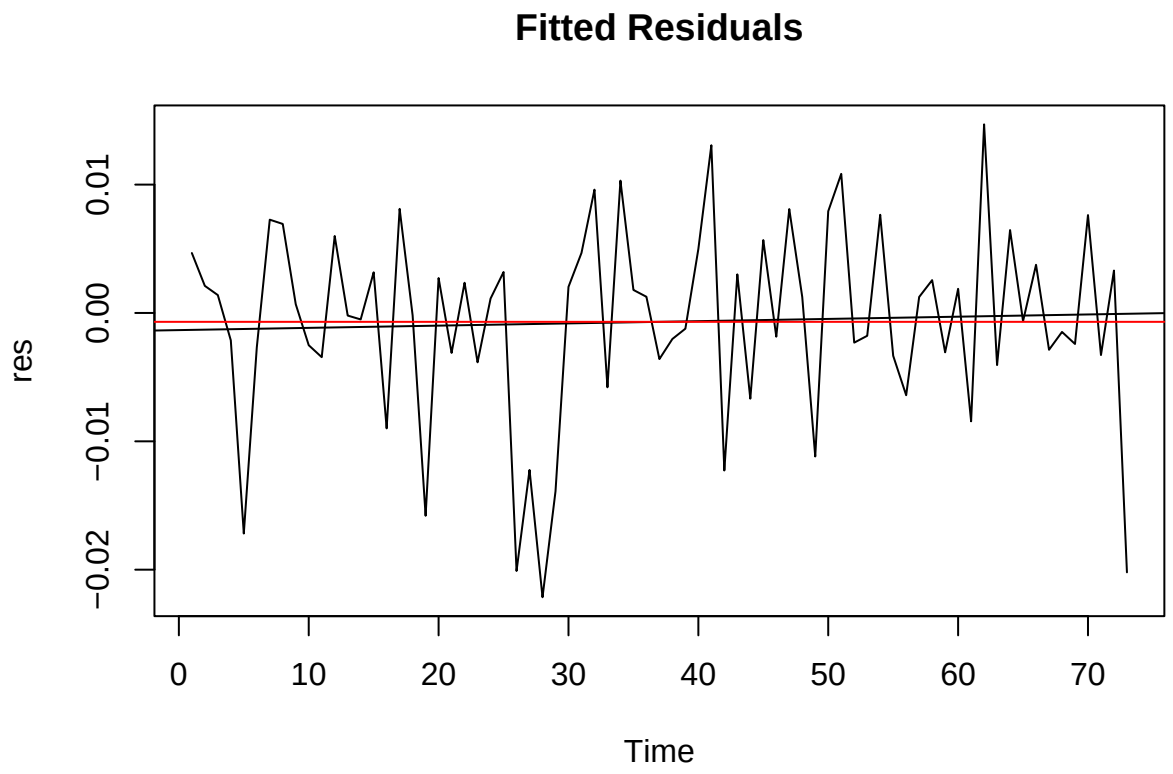
```
# Residual plots:
res <- fit.i$fit$residuals
mean(res)
```

```
## [1] -0.0006919207
```

```
var(res)
```

```
## [1] 5.892286e-05
```

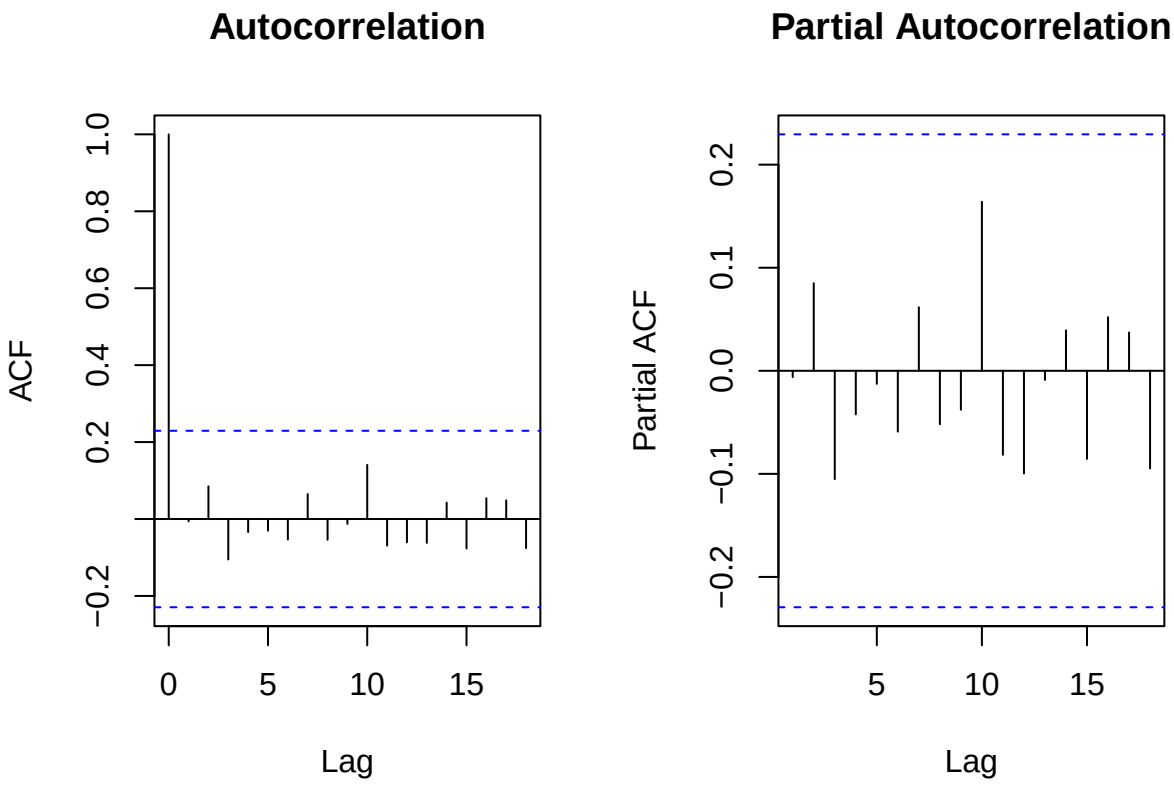
```
# layout(matrix(c(1, 1, 2, 3), 2, 2, byrow=T))
par(mfrow = c(1, 1))
ts.plot(res, main = "Fitted Residuals")
t <- 1:length(res)
fit.res <- lm(res ~ t)
abline(fit.res)
abline(h = mean(res), col = "red")
```




```r
var(fit.res$residuals)
```

```
## [1] 5.878606e-05
```

```r
# ACF and PACF:
par(mfrow = c(1, 2))
acf(res, main = "Autocorrelation")
pacf(res, main = "Partial Autocorrelation")
```



All acf and pacf of residuals are within confidence intervals and can be counted as zeros!

Model A passes all diagnostic except Shapiro-Wilk test (rejected at significance level 0.006).

```
# Test for independence of residuals
Box.test(res, lag = 9, type = c("Box-Pierce"), fitdf = 2)
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 2.231, df = 7, p-value = 0.946
```

Use lag = 9 because our sample size is approximately 81. Use fitdf = 2 because p + q = 2

```
Box.test(res, lag = 9, type = c("Ljung-Box"), fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 2.4344, df = 7, p-value = 0.932
```

```r
Box.test(res^2, lag = 9, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 7.647, df = 9, p-value = 0.5701
```

```r
# Test for normality of residuals
shapiro.test(res)
```

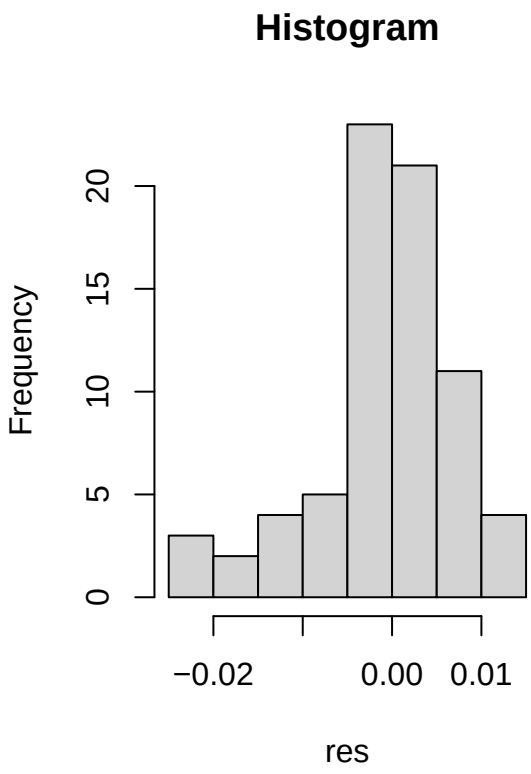
```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.95066, p-value = 0.006283
```

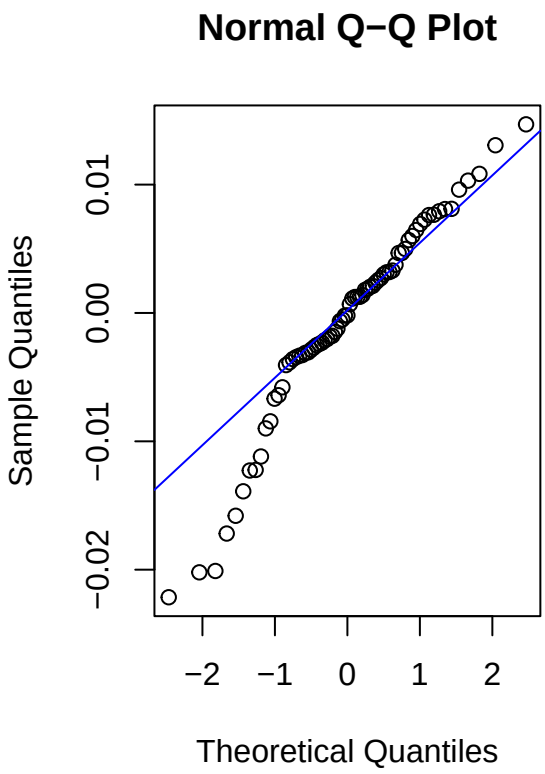
```r
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  5.892e-05
```

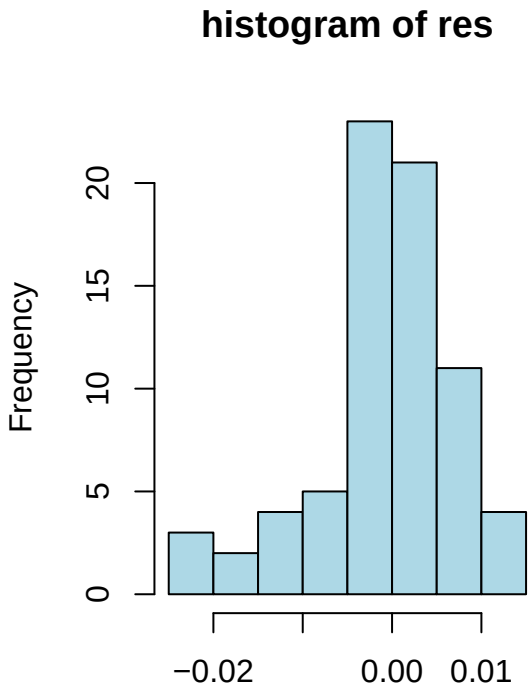
Residual appears not Gaussian as expected.

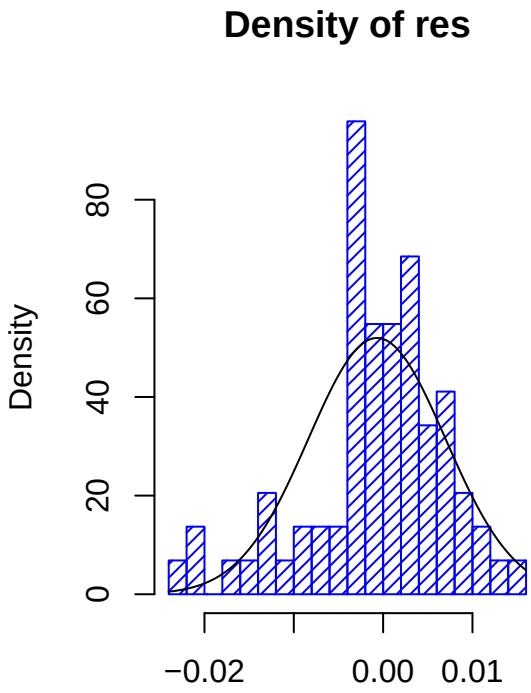
```r
# Histogram and QQ-plot:
par(mfrow = c(1, 2))
hist(res, main = "Histogram")
qqnorm(res)
qqline(res, col = "blue")
```






```r
par(mfrow = c(1, 2))
hist(res, col = "light blue", xlab = "", main = "histogram of res")
hist(res, density = 20, breaks = 20, col = "blue", xlab = "", main = "Density of res", prob = TRUE)
m <- mean(res)
# 0.001234
std <- sqrt(var(res))
curve(dnorm(x, m, std), add = TRUE)
```





```
# Access the coefficients of the SARIMA model
coefficients <- coef(fit.i$fit)

# Print the coefficients
coefficients
```

```
##        ar1       sma1
##  0.2132024 -0.7857388
```

```
# Access the variance
var <- fit.i$fit$sigma2
var
```

```
## [1] 5.807806e-05
```

Conclusions:

The residuals does not appear to be white noise and they are not normally distributed since the histogram appears skewed-left and the QQ plot shows residuals mostly aligning along the diagonal line but has few that doesn't.

Model (A) passed all diagnostic checking except Shapiro-Wilk test and is the final model.

Model (B) and other models that have too many parameters will not be used because they are likely to fail the diagnostic test.

Final Model for the logarithm transform of original data: ln(U_t) follows SARIMA (1,1,0)(0,1,1) 4 model.

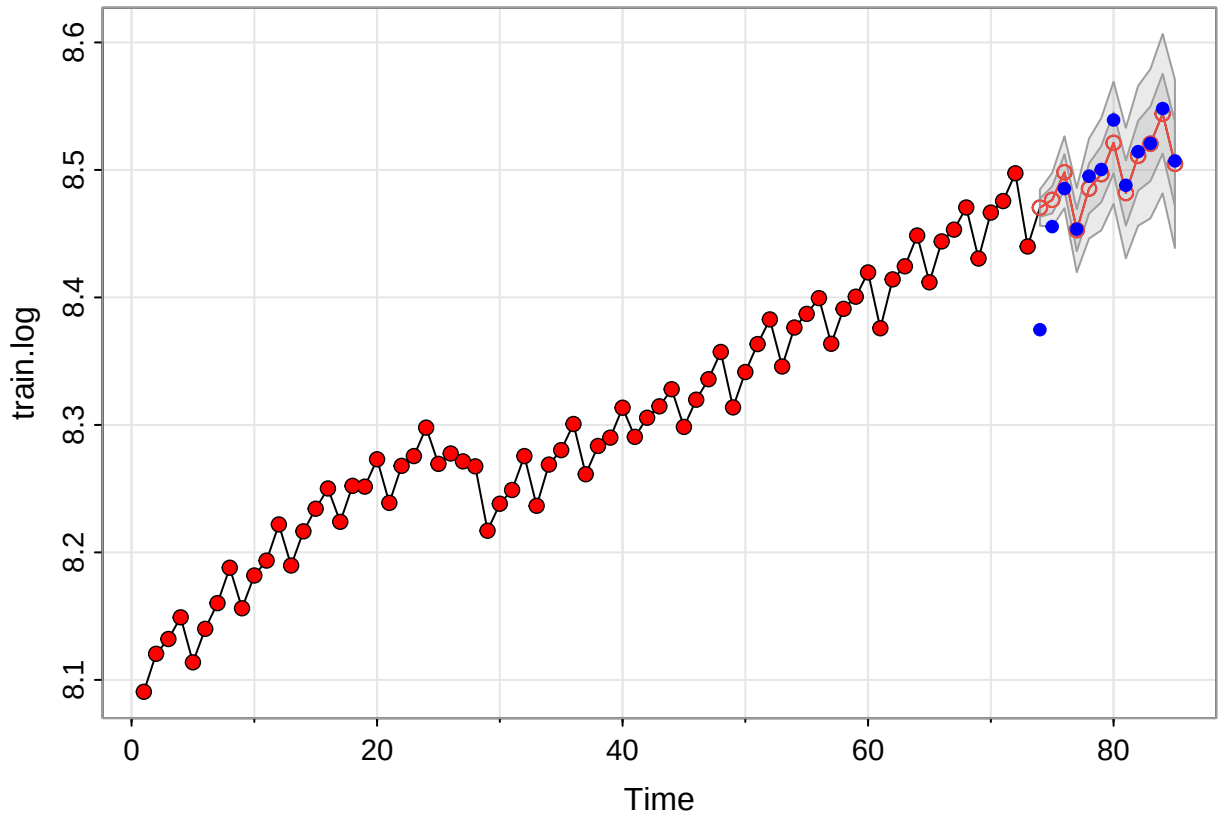
$\nabla 1 \nabla 4 ln(U_t) = (1 - 0.213_{(0.125)}B)Y_t = (1 - 0.786_{(0.120)}B^4)Z_t$, where $\hat{\sigma}_z^2 = 0.0000581$.

Compare and evaluate model forecasts using 12 test values from the true sample

```
train.log = ts(train.log) # Convert train.log to time series object
```

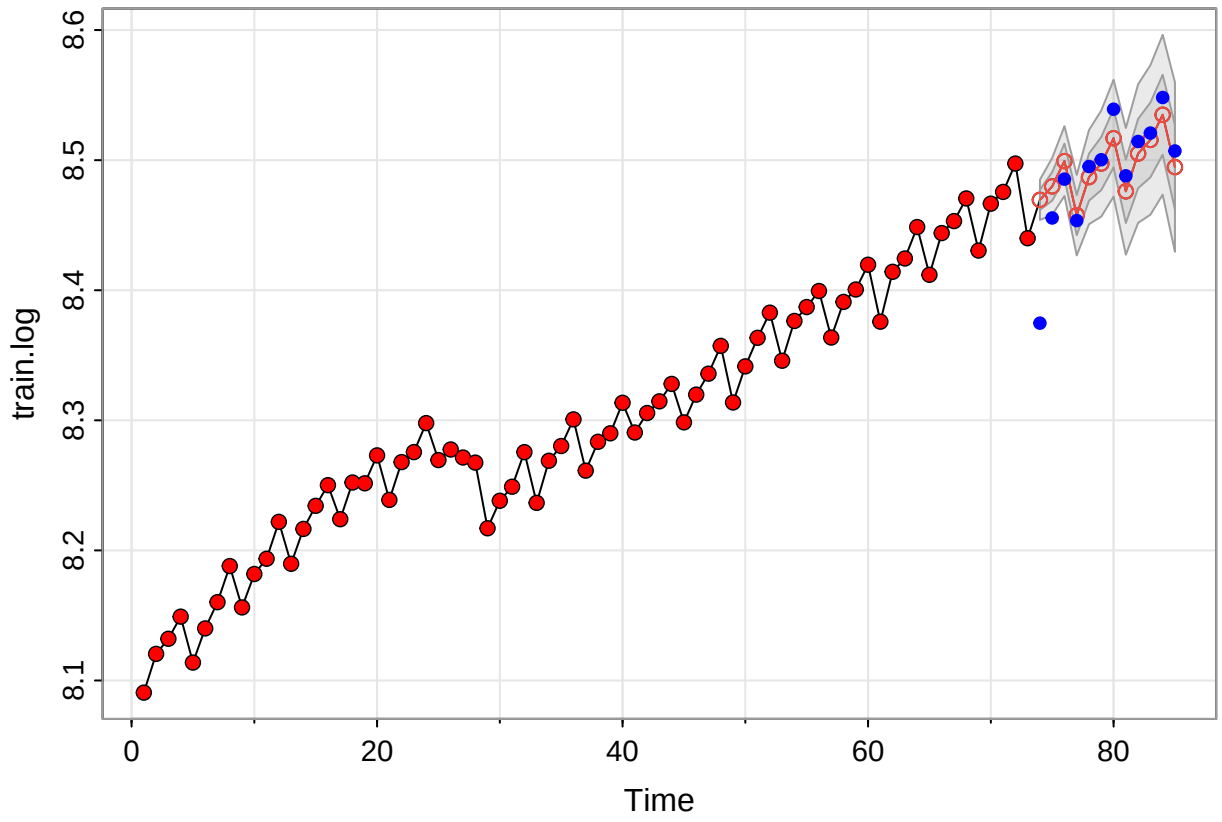
SARIMA (p = 4, d = 1, q = 4) × (P = 4, D = 1, Q = 4)s=4

```
x <- sarima.for(xdata = train.log, n.ahead = 12, p = 4, d = 1, q = 4, P = 4, D = 1, Q = 4, S = 4)
points(1:length(train), train.log, col = "red", pch = 19, cex = 0.8)
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```



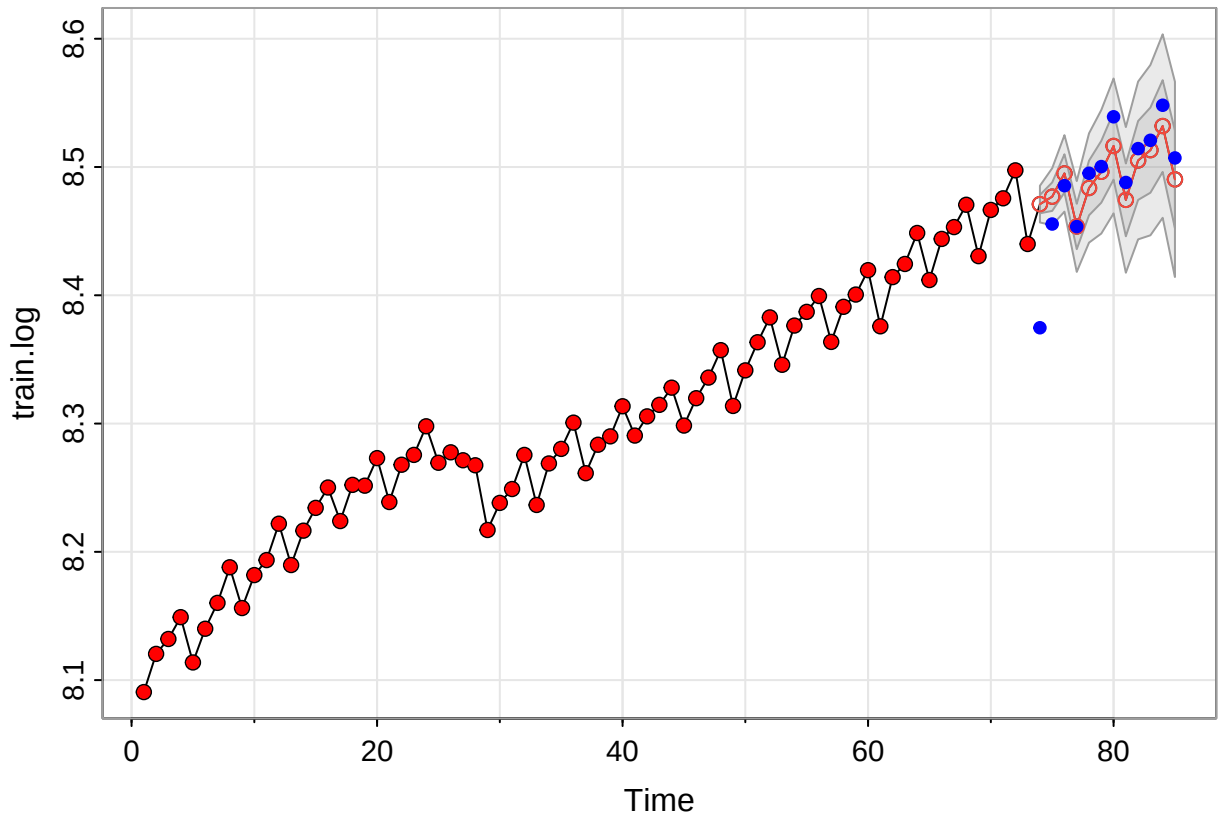
SARIMA (p = 0, d = 1, q = 0) × (P = 0, D = 1, Q = 4)s=4

```
x <- sarima.for(xdata = train.log, n.ahead = 12, p = 0, d = 1, q = 0, P = 0, D = 1, Q = 4, S = 4)
points(1:length(train), train.log, col = "red", pch = 19, cex = 0.8)
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```



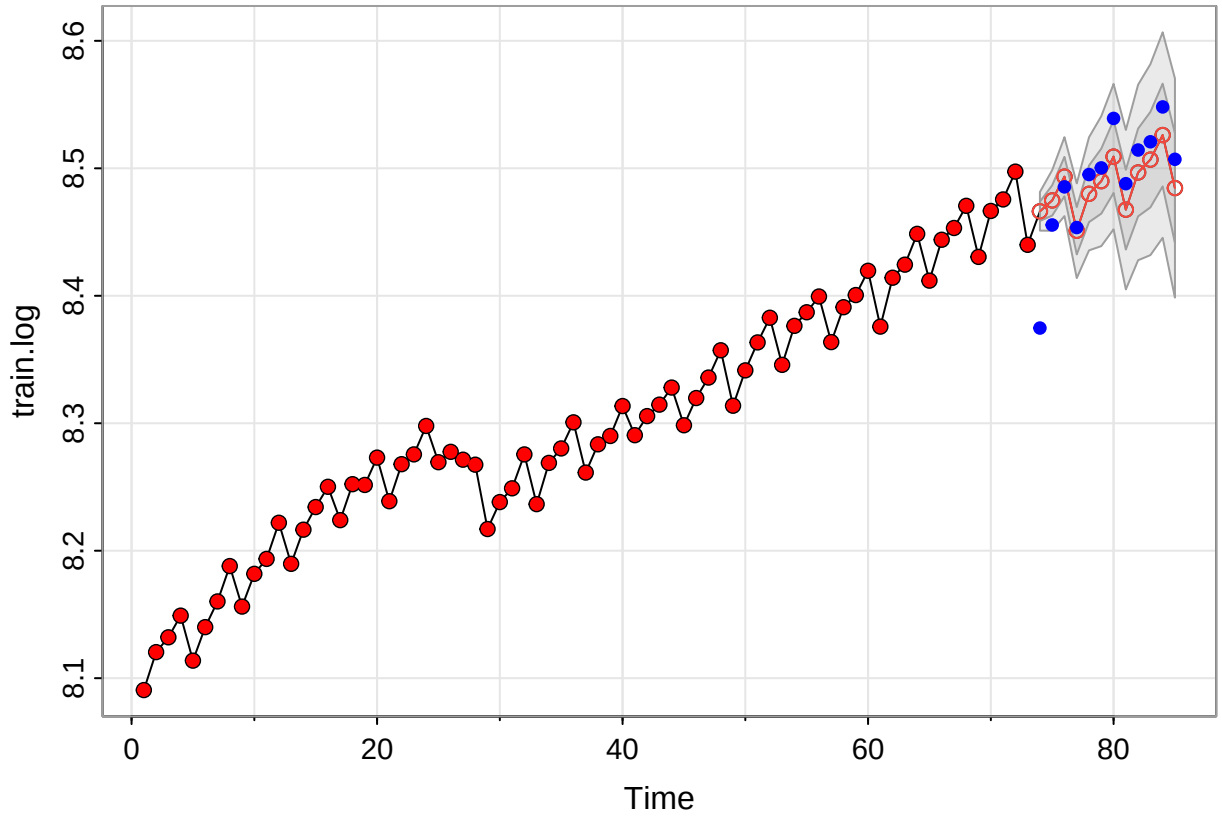
SARIMA $(p = 0, d = 1, q = 4) \times (P = 1, D = 1, Q = 1)s{=}4$

```
x <- sarima.for(xdata = train.log, n.ahead = 12, p = 0, d = 1, q = 4, P = 1, D = 1, Q = 1, S = 4)
points(1:length(train), train.log, col = "red", pch = 19, cex = 0.8)
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```



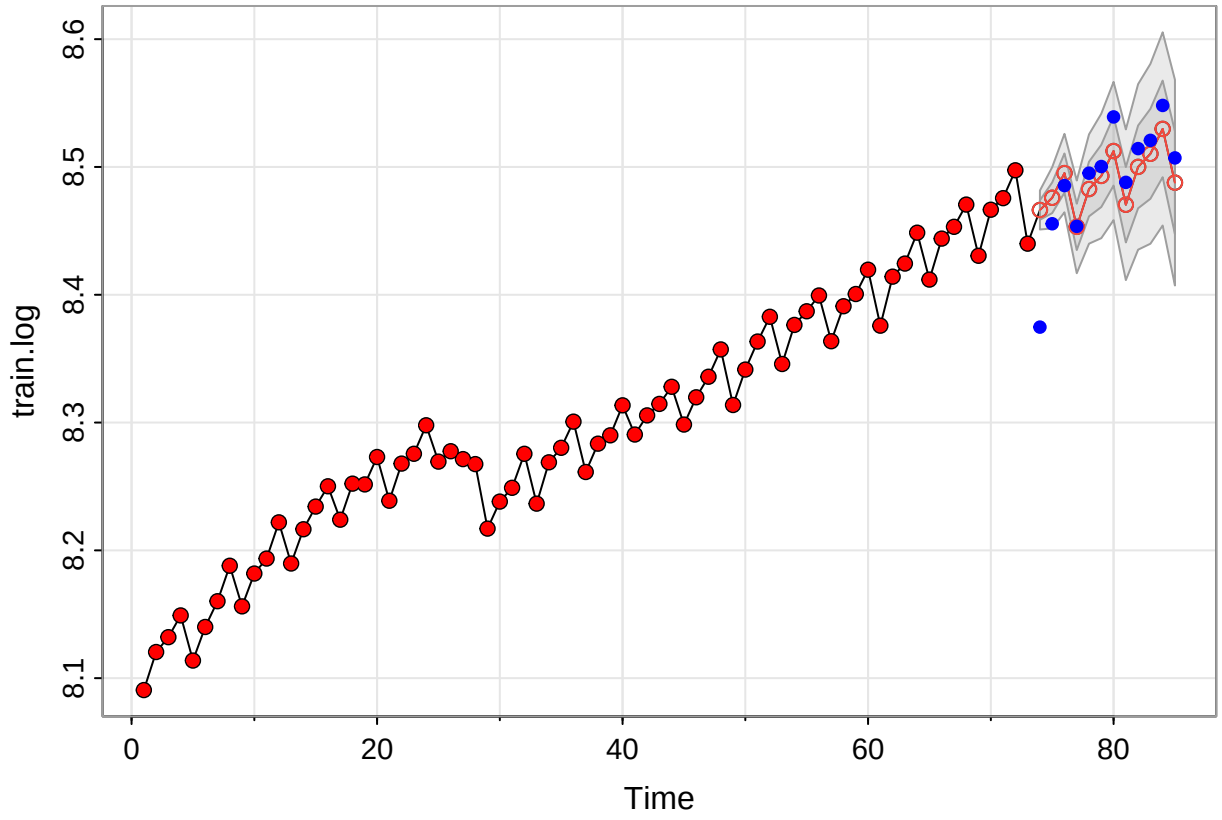
SARIMA $(p = 1, d = 1, q = 1) \times (P = 1, D = 1, Q = 1)s{=}4$

```
x <- sarima.for(xdata = train.log, n.ahead = 12, p = 1, d = 1, q = 1, P = 1, D = 1, Q = 1, S = 4)
points(1:length(train), train.log, col = "red", pch = 19, cex = 0.8)
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```



SARIMA $(p = 1, d = 1, q = 0) \times (P = 0, D = 1, Q = 1)s{=}4$

```
x <- sarima.for(xdata = train.log, n.ahead = 12, p = 1, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 4)
points(1:length(train), train.log, col = "red", pch = 19, cex = 0.8)
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```



We can see that all the above models presents fairly accurate forecasts. Suggesting that more than one model may be appropriate in forecasting U.S. Real GDP.

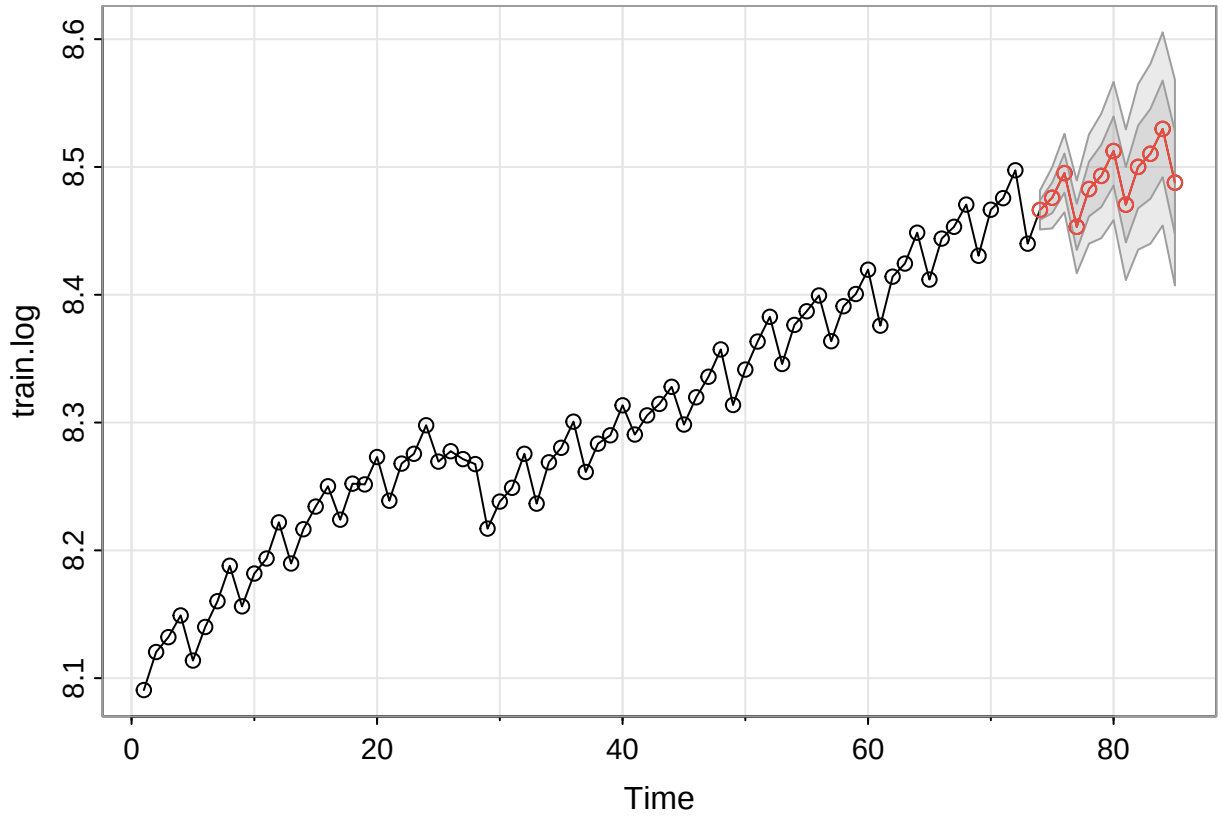
It is not clear which model best captures the trend and trajectory. This is reasonable since the original data is not Gaussian due to volatility during the Great Recession (December 2007 – June 2009) and the Pandemic (April 2020).

The non-Gaussian like behavior may cause the forecasting values to shift up and down depending on when the volatility occurred in historical data.

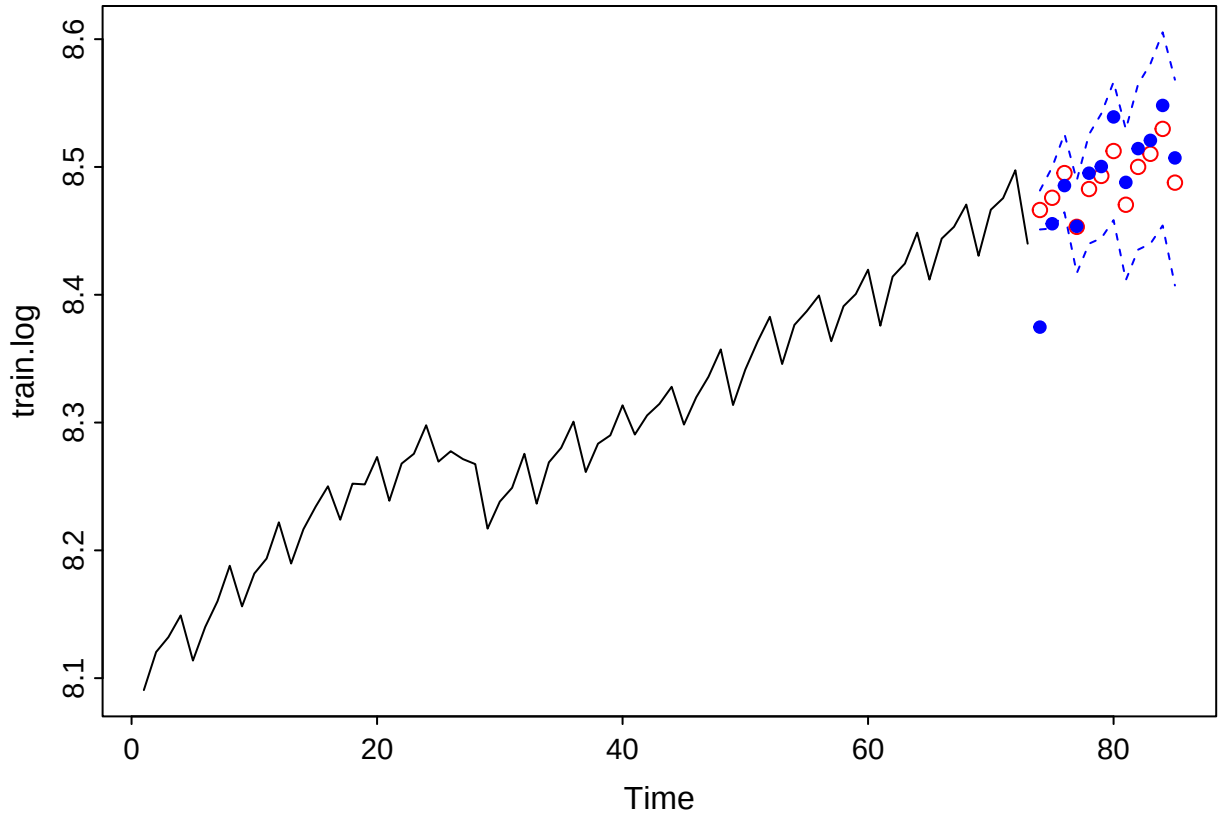
For example, the forecast produced by our final model may appear to be overestimating. This is because there was a recent crash in economy that decreases U.S. Real GDP drastically and is slowly recovering.

Forecast of transformed data using model A

```
pred.tr <- sarima.for(xdata = train.log, n.ahead = 12, p = 1, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 4)
```

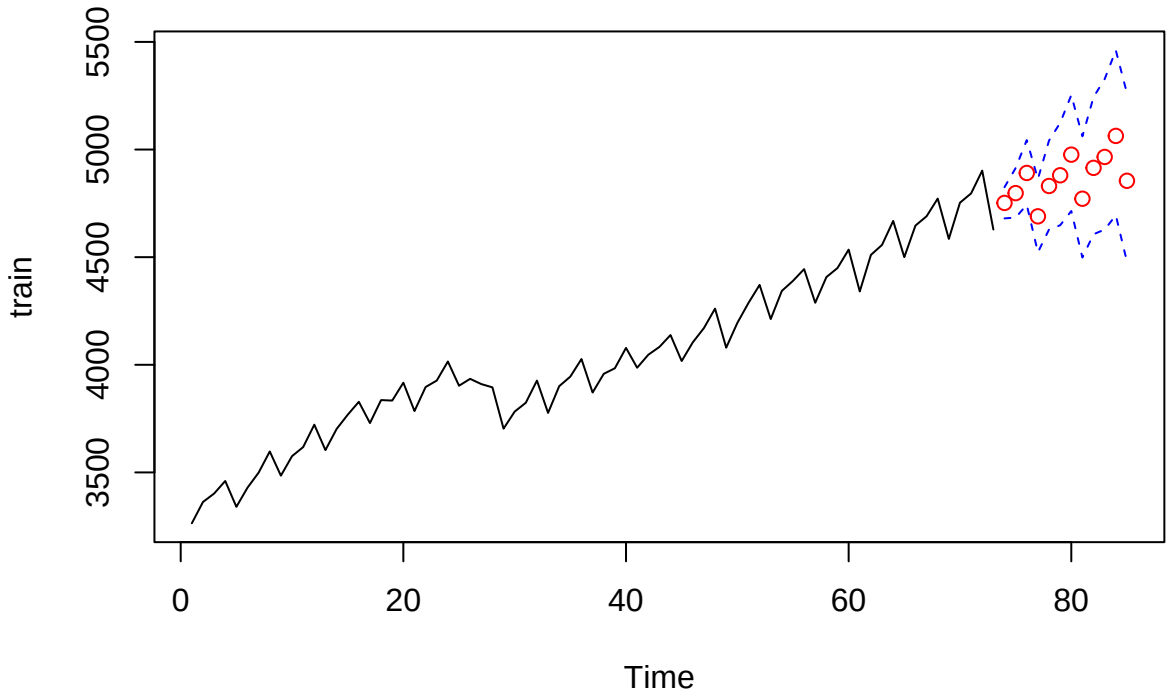



```
U.tr <- pred.tr$pred + 2 * pred.tr$se
L.tr <- pred.tr$pred - 2 * pred.tr$se
ts.plot(train.log, xlim = c(1, length(train.log) + 12), ylim = c(min(train.log), max(U.tr)))
lines(U.tr, col = "blue", lty = "dashed")
lines(L.tr, col = "blue", lty = "dashed")
points((length(train.log) + 1):(length(train.log) + 12), pred.tr$pred, col = "red")
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```



Forecast of original data using model A

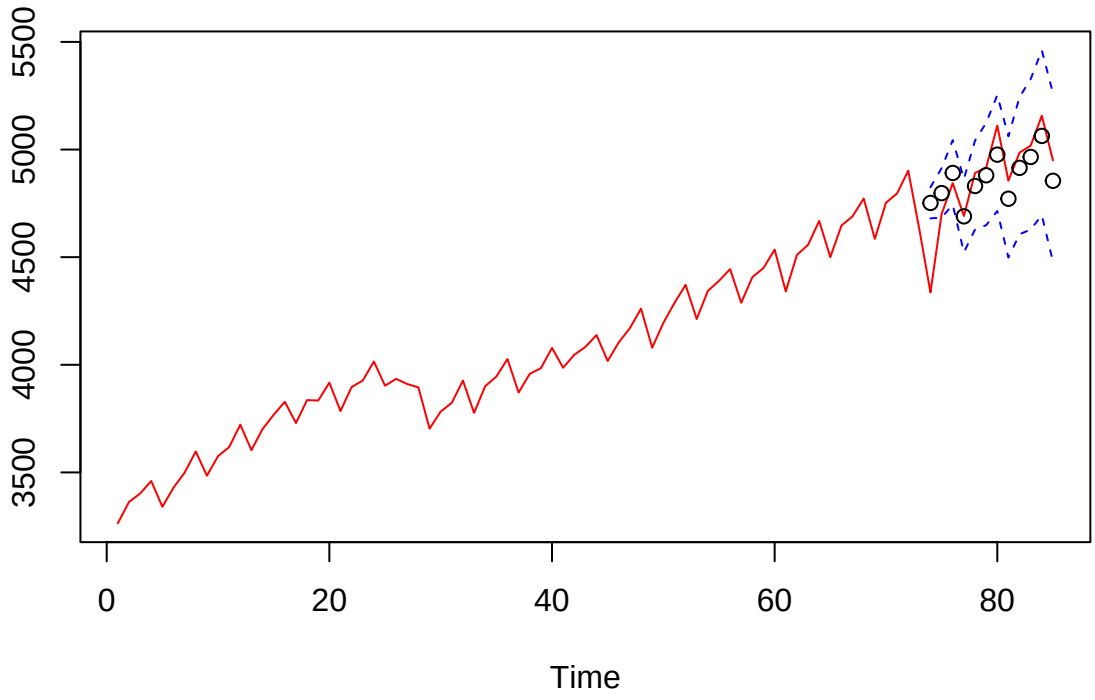
```
pred.orig <- exp(pred.tr$pred)
U <- exp(U.tr)
L <- exp(L.tr)
ts.plot(train, xlim = c(1, length(train) + 12), ylim = c(min(train), max(U)))
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(train) + 1):(length(train) + 12), pred.orig, col = "red")
```



Forecasts - Black circles

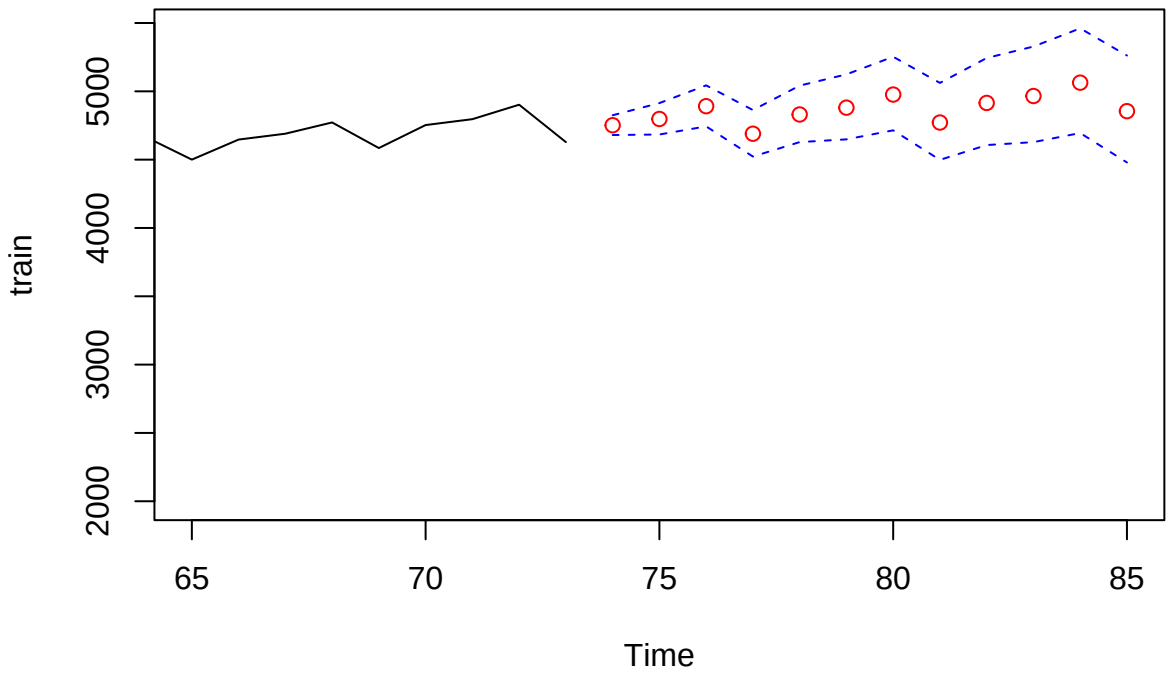
Original data - Red line

```
ts.plot(gdp.csv, xlim = c(1, length(train) + 12), ylim = c(min(train), max(U)), col = "red")
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(train) + 1):(length(train) + 12), pred.orig, col = "black")
```

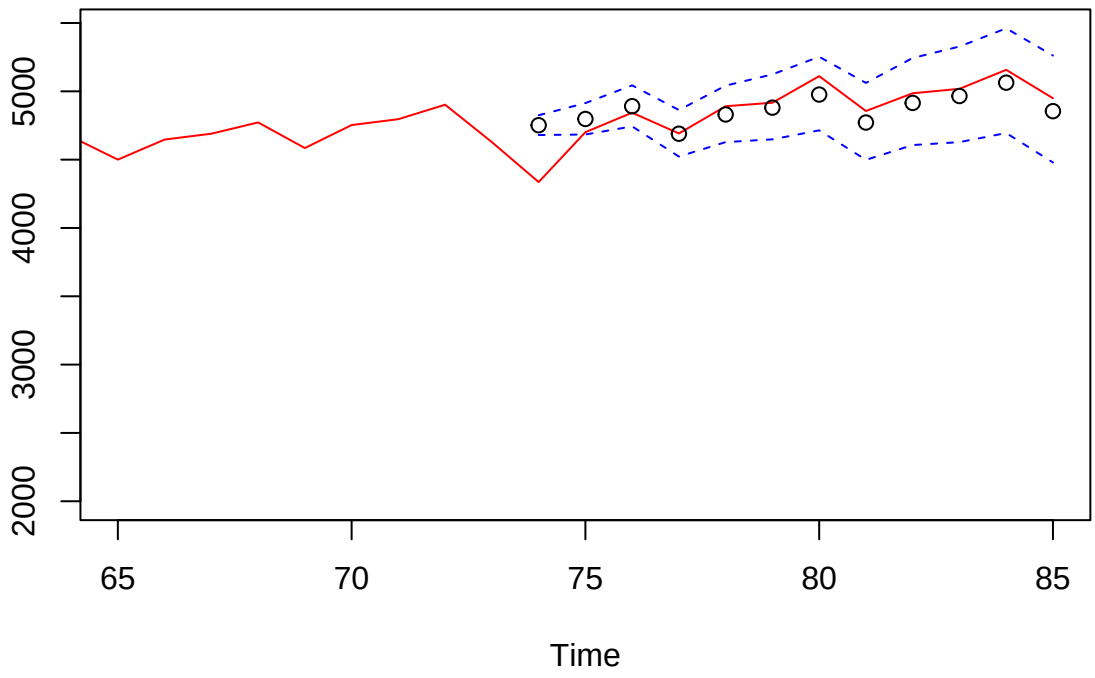


Zoomed Forecast of original Data using model A

```
ts.plot(train, xlim = c(65, length(train) + 12), ylim = c(2000, max(U)))
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(train) + 1):(length(train) + 12), pred.orig, col = "red")
```




```
ts.plot(gdp.csv, xlim = c(65, length(train) + 12), ylim = c(2000, max(U)), col = "red")
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(train) + 1):(length(train) + 12), pred.orig, col = "black")
```



We can see that the Test set is within prediction intervals except for the data point at April 2020 (pandemic).

Conclusion:

The results of the forecasting models revealed valuable insights into the behavior of U.S. real GDP over the studied period. The conclusions drawn from the project indicate that the selected models demonstrated effectiveness in capturing and predicting the dynamics of U.S. real GDP. The forecasts provided valuable information for decision-making, enabling policymakers, economists, and businesses to anticipate future economic trends and make informed decisions. The project contributes to a better understanding of the U.S. economy and provides insights into the factors driving U.S. real GDP growth.

Acknowledgments:

I would like to express my gratitude to Dr.Feldman for her guidance, support, and valuable insights throughout the project. Her expertise and feedback greatly enhanced the quality of the analysis and interpretation.

I would like to acknowledge the sources of the data used in this project, the U.S. Bureau of Economic Analysis (BEA) and the Federal Reserve Economic Data (FRED) database. Their comprehensive and reliable datasets served as the foundation for this analysis.

APPENDIX

Appendix A: Data Preparation

The following code snippet illustrates the process of loading and examining the GDP data set:

```
gdp.csv <- read.table("data/ND000334Q.csv", sep = ",", header = FALSE, skip = 1, nrows = 85)
head(gdp.csv)
tail(gdp.csv)
```

Appendix B: Data Description

Please note that the provided GDP data set is not Gaussian distributed due to the presence of volatility during significant economic events such as the Great Recession (December 2007 – June 2009) and the COVID-19 pandemic (April 2020).

Appendix C: Data Examination

Partition dataset to two parts for model training and model validation; work with training set:

```
gdp <- ts(gdp.csv[, 2], start = c(2002, 1), frequency = 4)
train <- gdp[1:73]
test <- gdp[74:85]
par(mfrow = c(1, 2))
ts.plot(gdp, ylab = "Quarterly U.S. GDP (Billions of Chained 2012 Dollars)")
plot.ts(train)

fit <- lm(train ~ as.numeric(1:length(train)))
abline(fit, col = "red")
abline(h = mean(train), col = "blue")
hist(train, col = "light blue", xlab = "", main = "Histogram: GDP data")
acf(train, lag.max = 40, main = "ACF of the GDP Data")
```

The histogram indicates a symmetric or bell-shaped distribution, although there may be slight right skewness. The autocorrelation function (ACF) plot shows large correlation values at various lags.

Appendix D: Transformation Comparison

The following code snippet compares different transformations applied to the GDP data:

```
bcTransform <- boxcox(train ~ as.numeric(1:length(train)))
bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda <- bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
train.bc <- (1 / lambda) * (train^lambda - 1)
test.bc <- (1 / lambda) * (test^lambda - 1)
train.log <- log(train)
train.sqrt <- sqrt(train)

op <- par(mfrow = c(2, 2))
```

```r
hist(train, col = "light blue", xlab = "", main = "Histogram: U_t")
hist(train.log, col = "light blue", xlab = "", main = "Histogram: ln(U_t)")
hist(train.bc, col = "light blue", xlab = "", main = "Histogram: bc(U_t)")
hist(train.sqrt, col = "light blue", xlab = "", main = "Histogram: sqrt(U_t)")

# Compare transforms
op <- par(mfrow = c(2, 2))
ts.plot(train, main = "Original Time Series")
ts.plot(train.bc, main = "Box-Cox Transform")
ts.plot(train.log, main = "Log Transform")
ts.plot(train.sqrt, main = "Square Root Transform")
```

In this project, the provided GDP data set was examined and preprocessed using various transformations. The histogram analysis suggests a symmetric or bell-shaped distribution, although with slight right skewness.

To address the goals of the project, several transformations were compared, including the Box-Cox transform, logarithmic transform, and square root transform. These transformations aimed to improve the distributional properties and reduce variance.

Appendix E: Seasonality and Trend Analysis

The following code snippet demonstrates the decomposition of the natural logarithm of the GDP data, $ln(U_t)$, to identify seasonality and trend components:

```r
y <- ts(as.ts(train.log), frequency = 4)
decomp <- decompose(y)
plot(decomp)
```

Appendix F: Differencing Analysis

The following code snippet showcases the process of differencing the $ln(U_t)$ time series at different lags:

```r
var(train.log)
plot.ts(train.log, main = "Ln(U_t)")

train.log_4 <- diff(train.log, lag = 4)
plot.ts(train.log_4, main = "Ln(U_t) differenced at lag 4")
var(train.log_4)
fit <- lm(train.log_4 ~ as.numeric(1:length(train.log_4)))
abline(fit, col = "red")
mean(train.log_4)
abline(h = mean(train.log_4), col = "blue")

train.log_4_1 <- diff(train.log_4, lag = 1)
plot.ts(train.log_4_1, main = "Ln(U_t) differenced at lags 4 and then 1")
var(train.log_4_1)
```

```r
fit <- lm(train.log_4_1 ~ as.numeric(1:length(train.log_4_1)))
abline(fit, col = "red")
mean(train.log_4_1)
abline(h = mean(train.log_4_1), col = "blue")
```

The first differencing at lag 4, denoted as $ln(U_t)_{\Delta 4}$, is performed to remove the seasonality component. The resulting time series shows reduced seasonality and potentially a stationary behavior. The second differencing at lag 1, denoted as $\Delta 4 \Delta 1 ln(U_t)$, is applied to address any remaining trend component. The resulting time series exhibits a relatively constant mean and variance.

Appendix G: Identification of Candidate SARIMA Models

The following analysis focuses on identifying candidate SARIMA$(p, d, q) \times (P, D, Q)_s$ models for the $ln(U_t)$ time series.

1. ACF and PACF Analysis:

```r
par(mfrow = c(1, 3))
acf(train.log, lag.max = 40, main = "ACF: log(U_t)")
acf(train.log_4, lag.max = 40, main = "ACF: log(U_t)_4")
acf(train.log_4_1, lag.max = 40, main = "ACF: log(U_t)_4_1")
```

The ACF and PACF plots provide insights into the stationarity and seasonality of the time series. The following observations are made:

- ACF of $ln(U_t)$: Slower decay indicates non-stationarity, and the presence of seasonality is evident.
- ACF of $ln(U_t)$ differenced at lag 4: Seasonality is no longer apparent, but non-stationarity remains.
- ACF of $ln(U_t)$ differenced at lags 4 & 1: The ACF decay corresponds to a stationary process, indicating a potential candidate model.

2. Histogram Analysis:

```r
par(mfrow = c(1, 2))
hist(train.log_4_1, col = "light blue", xlab = "", main = "ln(U_t) differenced at lags 4 & 1")
hist(train.log_4_1, density = 20, breaks = 20, col = "blue", xlab = "", main = "Density", prob = TRUE)
m <- mean(train.log_4_1)
std <- sqrt(var(train.log_4_1))
```

The histogram of $\nabla 1 \nabla 4 ln(U_t)$ appears symmetric and almost Gaussian, indicating a potential candidate for a stationary process.

3. ACF and PACF Analysis of $\nabla 1 \nabla 4 ln(U_t)$:

```r
par(mfrow = c(1, 2))
acf(train.log_4_1, lag.max = 20, main = "ACF of ln(U_t)_4_1")
pacf(train.log_4_1, lag.max = 20, main = "PACF of ln(U_t)_4_1")
```

The ACF and PACF plots of $\nabla 1 \nabla 4 ln(U_t)$ provide insights into the selection of the AR and MA components for the candidate SARIMA models.

Based on the analyses performed, the following candidate models are considered:

1. SARIMA$(p = 1, d = 1, q = 1) \times (P = 1, D = 1, Q = 1)s = 4$
2. SARIMA$(p = 0, d = 1, q = 1) \times (P = 1, D = 1, Q = 1)s = 4$
3. SARIMA$(p = 0, d = 1, q = 0) \times (P = 0, D = 1, Q = 1)s = 4$
4. SARIMA$(p = 1, d = 1, q = 0) \times (P = 0, D = 1, Q = 1)s = 4$

These models are selected based on the observed characteristics in the ACF, PACF, and histogram plots.

We have identified potential candidate SARIMA models for the $ln(U_t)$ time series. The selected models will be further evaluated for their accuracy and performance in modeling and forecasting the U.S. real GDP.

Appendix H: Evaluation of Candidate Models

```
arima(train.log, order = c(4, 1, 4), seasonal = list(order = c(4, 1, 4), period = 4), method = "ML")
AICc(arima(train.log, order = c(4, 1, 4), seasonal = list(order = c(4, 1, 4), period = 4), method = "ML
```

```
arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML")
AICc(arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML
```

```
arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 4), fixed = c(NA, 0, 
AICc(arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 1), period = 4), fixed = c(NA
```

```
arima(train.log, order = c(1, 1, 0), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML")
AICc(arima(train.log, order = c(1, 1, 0), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML
```

```
arima(train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML")
AICc(arima(train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML
```

```
arima(train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1), period = 4), fixed = c(NA, NA,
AICc(arima(train.log, order = c(0, 1, 4), seasonal = list(order = c(1, 1, 1), period = 4), method = "ML
```

```
arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML")
AICc(arima(train.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML
```

```
arima(train.log, order = c(0, 1, 0), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML")
AICc(arima(train.log, order = c(0, 1, 0), seasonal = list(order = c(0, 1, 1), period = 4), method = "ML
```

Appendix I: Fitting Selected Model

```
fit.i <- sarima(xdata = train.log, p = 1, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 4)

print("Coefficients")
fit.i$fit$coef
```

```r
# Residual plots:
res <- fit.i$fit$residuals
mean(res)
var(res)

# layout(matrix(c(1, 1, 2, 3), 2, 2, byrow=T))
par(mfrow = c(1, 1))
ts.plot(res, main = "Fitted Residuals")
t <- 1:length(res)
fit.res <- lm(res ~ t)
abline(fit.res)
abline(h = mean(res), col = "red")

var(fit.res$residuals)
```

```r
# ACF and PACF:
par(mfrow = c(1, 2))
acf(res, main = "Autocorrelation")
pacf(res, main = "Partial Autocorrelation")
```

Appendix J: Model Diagnostic

```r
# Test for independence of residuals
Box.test(res, lag = 9, type = c("Box-Pierce"), fitdf = 2)
```

The Box-Pierce test is used to assess the independence of residuals in the SARIMA model. The test is applied with a lag of 9, considering our sample size of approximately 81. The fitdf parameter is set to 2, which represents the sum of the AR and MA orders (p + q) in the model.

```r
Box.test(res, lag = 9, type = c("Ljung-Box"), fitdf = 2)
```

The Ljung-Box test is another method to evaluate the independence of residuals. Similar to the Box-Pierce test, we use a lag of 9 and fitdf = 2.

```r
Box.test(res^2, lag = 9, type = c("Ljung-Box"), fitdf = 0)
```

In addition to the Ljung-Box test for the residuals themselves, we can also perform the test on the squared residuals. This helps to assess whether there is any remaining autocorrelation after applying the SARIMA model.

```r
# Test for normality of residuals
shapiro.test(res)
```

The Shapiro-Wilk test is used to examine the normality of the residuals. In our case, the test indicates whether the residuals follow a normal distribution. The null hypothesis is that the residuals are normally distributed. The test is rejected at a significance level of 0.006, suggesting a departure from normality.

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

The Yule-Walker method is employed to estimate the autoregressive (AR) coefficients of the residuals. The "ar" function is used with the "aic = TRUE" option to select the optimal order based on the Akaike Information Criterion (AIC). The order.max parameter is set to NULL to consider all possible orders, and the method is specified as "yule-walker."

These diagnostic tests provide insights into the adequacy of the selected SARIMA model. While Model A passes most of the tests for independence of residuals, it is rejected by the Shapiro-Wilk test, indicating non-normality in the residuals at a significance level of 0.006. This suggests a departure from the assumption of normality in the model's errors.

Appendix K: Visualizing Residuals

```r
# Histogram and QQ-plot:
par(mfrow = c(1, 2))
hist(res, main = "Histogram")
qqnorm(res)
qqline(res, col = "blue")

par(mfrow = c(1, 2))
hist(res, col = "light blue", xlab = "", main = "Histogram of Residuals")
hist(res, density = 20, breaks = 20, col = "blue", xlab = "", main = "Density of Residuals", prob = TRU
m <- mean(res)
std <- sqrt(var(res))
curve(dnorm(x, m, std), add = TRUE)
```

The histograms and QQ-plot provide visual representations of the residuals. The histogram shows that the residuals are skewed to the left, indicating a departure from a symmetric Gaussian distribution. The QQ-plot suggests that the majority of the residuals align along the diagonal line, but there are a few deviations from normality.

```r
# Access the coefficients of the SARIMA model
coefficients <- coef(fit.i$fit)

# Print the coefficients
coefficients

# Access the variance
var <- fit.i$fit$sigma2
var
```

Appendix L: Forecast Candidate Models

```r
train.log = ts(train.log) # Convert train.log to time series object
```

SARIMA $(p = 0, d = 1, q = 0) \times (P = 0, D = 1, Q = 4)s{=}4$

```
x <- sarima.for(xdata = train.log, n.ahead = 12, p = 0, d = 1, q = 0, P = 0, D = 1, Q = 4, S = 4)
points(1:length(train), train.log, col = "red", pch = 19, cex = 0.8)
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```

SARIMA $(p = 0, d = 1, q = 4) \times (P = 1, D = 1, Q = 1)s{=}4$

```
x <- sarima.for(xdata = train.log, n.ahead = 12, p = 0, d = 1, q = 4, P = 1, D = 1, Q = 1, S = 4)
points(1:length(train), train.log, col = "red", pch = 19, cex = 0.8)
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```

SARIMA $(p = 1, d = 1, q = 1) \times (P = 1, D = 1, Q = 1)s{=}4$

```
x <- sarima.for(xdata = train.log, n.ahead = 12, p = 1, d = 1, q = 1, P = 1, D = 1, Q = 1, S = 4)
points(1:length(train), train.log, col = "red", pch = 19, cex = 0.8)
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```

SARIMA $(p = 1, d = 1, q = 0) \times (P = 0, D = 1, Q = 1)s{=}4$

```
x <- sarima.for(xdata = train.log, n.ahead = 12, p = 1, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 4)
points(1:length(train), train.log, col = "red", pch = 19, cex = 0.8)
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```

We can see that all the above models presents fairly accurate forecasts. Suggesting that more than one model may be appropriate in forecasting U.S. Real GDP.

It is not clear which model best captures the trend and trajectory. This is reasonable since the original data is not Gaussian due to volatility during the Great Recession (December 2007 – June 2009) and the Pandemic (April 2020).

The non-Gaussian like behavior may cause the forecasting values to shift up and down depending on when the volatility occurred in historical data.

For example, the forecast produced by our final model may appear to be overestimating. This is because there was a recent crash in economy that decreases U.S. Real GDP drastically and is slowly recovering.

Appendix M: Forecast Selected Model Using Transformed Data

Forecast of transformed data using model A

```
pred.tr <- sarima.for(xdata = train.log, n.ahead = 12, p = 1, d = 1, q = 0, P = 0, D = 1, Q = 1, S = 4)
U.tr <- pred.tr$pred + 2 * pred.tr$se
L.tr <- pred.tr$pred - 2 * pred.tr$se
ts.plot(train.log, xlim = c(1, length(train.log) + 12), ylim = c(min(train.log), max(U.tr)))
lines(U.tr, col = "blue", lty = "dashed")
lines(L.tr, col = "blue", lty = "dashed")
points((length(train.log) + 1):(length(train.log) + 12), pred.tr$pred, col = "red")
points(length(train) + 1:length(test), log(test), col = "blue", pch = 19, cex = 0.8)
```

Appendix N: Forecast Selected Model Using Original Data

Forecast of original data using model A

```
pred.orig <- exp(pred.tr$pred)
U <- exp(U.tr)
L <- exp(L.tr)
ts.plot(train, xlim = c(1, length(train) + 12), ylim = c(min(train), max(U)))
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(train) + 1):(length(train) + 12), pred.orig, col = "red")
```

Forecasts - Black circles

Original data - Red line

```
ts.plot(gdp.csv, xlim = c(1, length(train) + 12), ylim = c(min(train), max(U)), col = "red")
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(train) + 1):(length(train) + 12), pred.orig, col = "black")
```

Zoomed Forecast of original Data using model A

```
ts.plot(train, xlim = c(65, length(train) + 12), ylim = c(2000, max(U)))
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(train) + 1):(length(train) + 12), pred.orig, col = "red")

ts.plot(gdp.csv, xlim = c(65, length(train) + 12), ylim = c(2000, max(U)), col = "red")
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points((length(train) + 1):(length(train) + 12), pred.orig, col = "black")
```

We can see that the Test set is within prediction intervals except for the data point at April 2020 (pandemic).