Pstat 174 Final Project

Roxanne I. Dimadi

roxannedimadi@ucsb.edu

03/20/2025

Abstract

This project applies Seasonal Autoregressive Integrated Moving Average (SARIMA) modeling to analyze U.S. unemployment rates. The objective was to identify an appropriate model that effectively captures both the trend and seasonal dynamics of the time series. After visually inspecting the data and applying transformations, including Box-Cox and seasonal differencing, various SARIMA models were estimated and compared based on model fit criteria, diagnostic checks, and residual behavior. Spectral analysis and periodograms revealed a lingering spike around frequency 0.2, suggesting potential periodicities not fully captured by the SARIMA structure.

Introduction

This project analyzes the U.S. unemployment rate using monthly data from February 2010 to March 2018, sourced from the Federal Reserve Economic Data (FRED) using time series methods. The primary goal is to identify and model the underlying trend and seasonal patterns in the data. I chose this dataset because unemployment trends offer valuable insights for someone like me, entering the workforce in the next three months. Previous studies have frequently applied ARIMA and SARIMA models to the unemployment series. In this project, we apply seasonal and trend differencing, spectral analysis, and SARIMA modeling to capture the structure of the data. Important findings include the detection of strong annual seasonality and the validation of model adequacy through both residual diagnostics and frequency domain analysis.

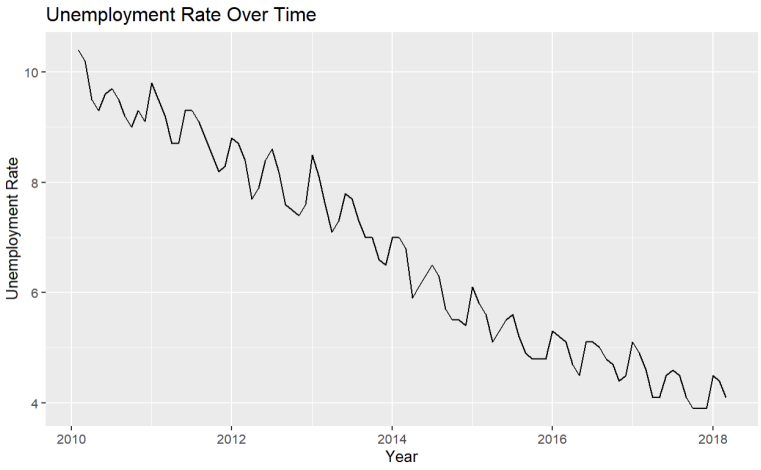
Data

The dataset used in this project contains monthly observations of the U.S. unemployment rate from February 2010 to March 2018, resulting in a total of 98 data points. The frequency of the dataset is monthly, and the values represent the unemployment rate in percentage terms, not seasonally adjusted. This dataset was chosen because I am looking for jobs and would like to know when unemployment is at its peak, indicating poor labor market conditions for individuals searching for jobs. The data was sourced from the Federal Reserve Economic Data (FRED) and can be accessed at https://fred.stlouisfed.org/series/UNRATENSA. The original data was collected by the U.S. Bureau of Labor Statistics (BLS) through the Current Population Survey (CPS), a monthly survey of approximately 60,000 households. This survey measures labor force status, employment, and unemployment. The purpose of studying this dataset is to better understand trends in unemployment and how they may reflect broader macroeconomic dynamics over time. Analyzing this data can help uncover labor market shifts and inform both academic research and policy decisions.

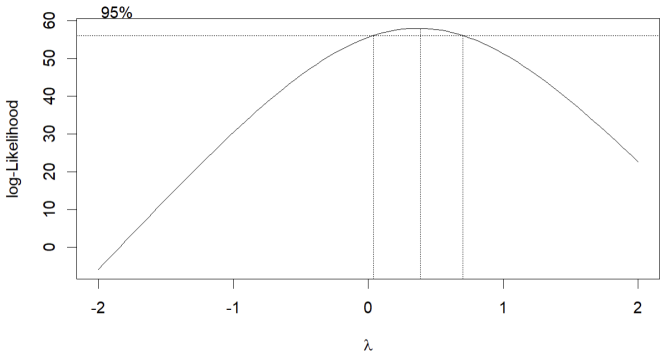
Methodology

This project applies the Box-Jenkins time series modeling framework to analyze monthly U.S. unemployment data. The series was first differenced at lag 12 and lag 1 to remove seasonal and trend components, respectively, and stationarity was confirmed using the Augmented Dickey-Fuller test. Several SARIMA models were then estimated on the differenced data, and model selection was based on AICc values and residual diagnostics, including the Ljung-Box test and residual plots. After identifying a suitable model, spectral analysis was conducted as a post-estimation diagnostic tool to further assess the adequacy of the model. Periodograms and the Kolmogorov-Smirnov test were used to evaluate whether residuals exhibited white noise behavior in the frequency domain, providing additional evidence on the effectiveness of the model in capturing underlying cyclical patterns in the data.

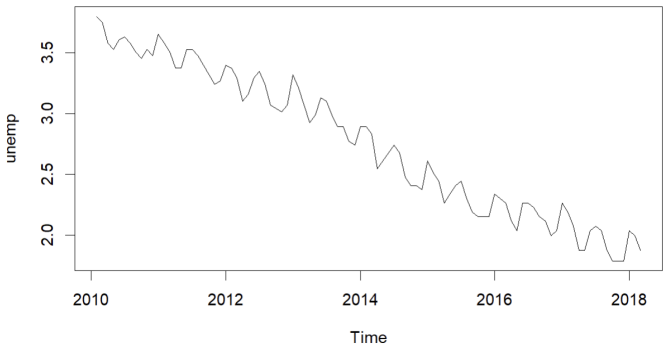
Results



This time series plot shows a clear downward trend in the unemployment rate from 2010 to 2018, with noticeable recurring seasonal fluctuations throughout the period.

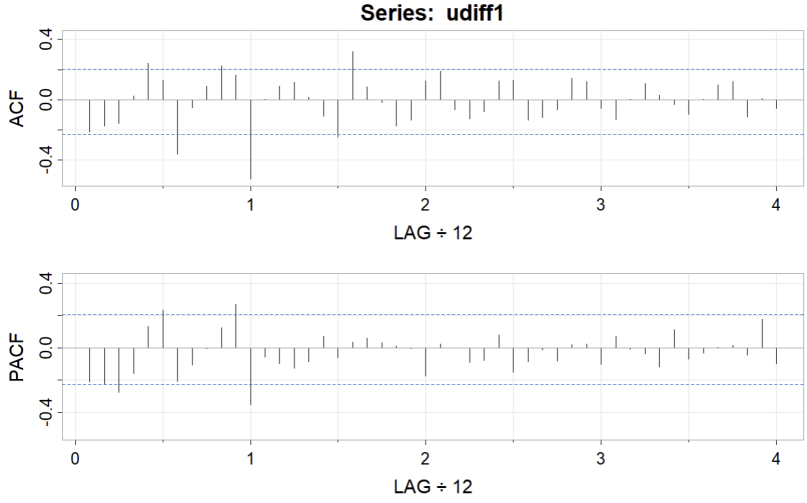


The Box-Cox log-likelihood plot indicates that a transformation around $\lambda = 0$. I applied a log transformation that helps stabilize the variance.



This is what my transformed data looks like. My PACF/ACF plots indicate a significant negative autocorrelation at lag 1 in the ACF and a sharp cutoff in the PACF after lag 1, suggesting a possible MA(1) or AR(1) structure; additionally, a spike at seasonal lag

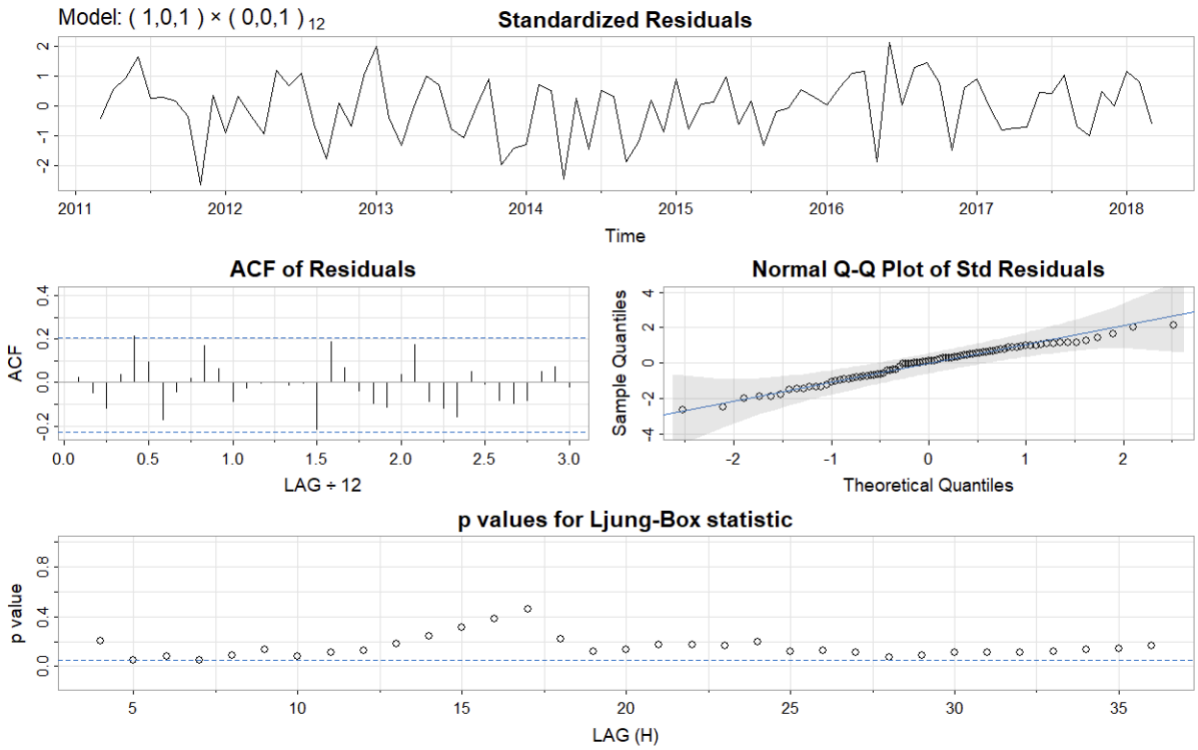
12 hints at seasonal components.



I implemented a multiplicative Seasonal ARIMA model, specifically SARIMA(1,0,1)(0,0,1)[12]. This type of model is designed to capture both short-term patterns and repeating seasonal effects—here, seasonality occurring every 12 months.The general form of the SARIMA model, from the textbook, is: $\Phi_P(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \delta + \Theta_Q(B^s)\theta(B)w_t,$ where ***B*** is the backshift operator and w_t is white noise. Expanding this equation using my model's terms, the SARIMA becomes: [$(1 - \varphi_1 B)\cdot x_t = (1+\theta_1 B)(1+\Theta B^{12})w_t$]. Expanding this product leads to: [$x_t - \varphi_1 x_{t-1} = w_t + \theta_1 w_{t-1} + \Theta_1 w_{t-12} + \theta_1\Theta_1 w_{t-13}$].

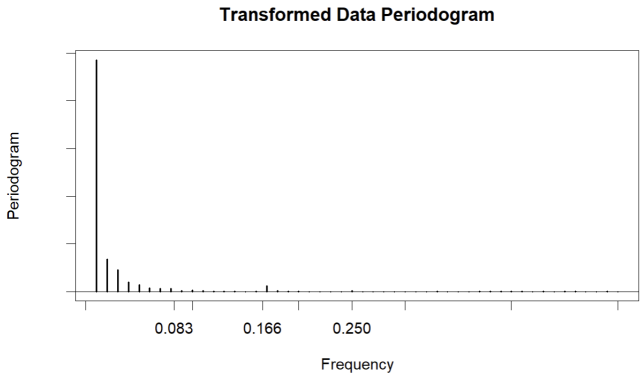
The final selected model is a SARIMA(1,0,1)(0,0,1)[12], which includes a non-seasonal AR(1), MA(1), and a seasonal MA(1) component. Although the AR(1) term is not statistically significant at the 5% level ($p = 0.0712$), it was retained because it noticeably improved model diagnostics—particularly by increasing Ljung-Box p-values and reducing residual autocorrelation, contributing to a better overall model fit. The non-seasonal MA(1) ($p < 0.0001$) and seasonal MA(1) ($p < 0.0001$) terms are both highly statistically significant, indicating that

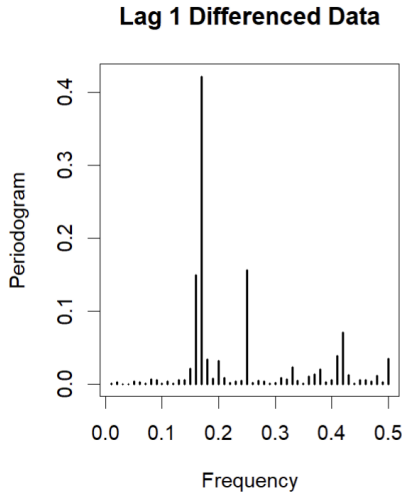
both short-term and seasonal shock structures are well captured by the model. The model achieved an AICc value of -3.0116, supporting its good fit.

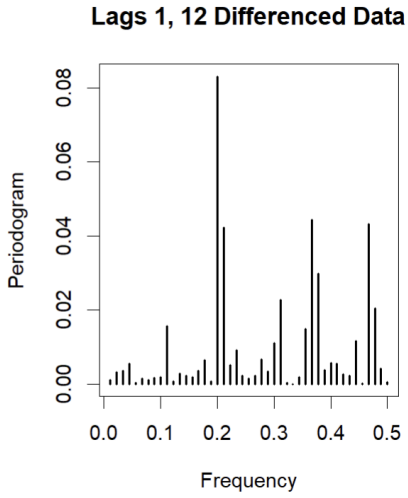


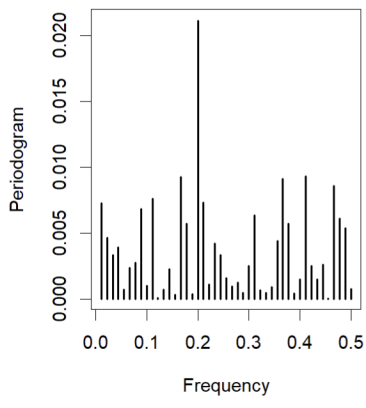
The standardized residuals represent the model's in-samle prediction errors, which is the differences between the observed values and the fitted values from the SARIMA model. They appear to be centered around zero with no obvious trends. The ACF of residuals shows if there is remaining autocorrelation in the residuals. Most spikes lie within the significance bounds, but a few spikes are borderline significant, especially around early lags (lag 1–3), suggesting some mild remaining autocorrelation, but it's not severe enough to reject the model. The Normal Q-Q plot tests whether the residuals are normally distributed. Most of the residuals fall close to the 45-degree line, indicating approximate normality. The Ljung-Box test p-values for this model remain above the 0.05 threshold across most lags, indicating that the residuals are largely free from autocorrelation and resemble white noise, which supports the adequacy of the model fit.

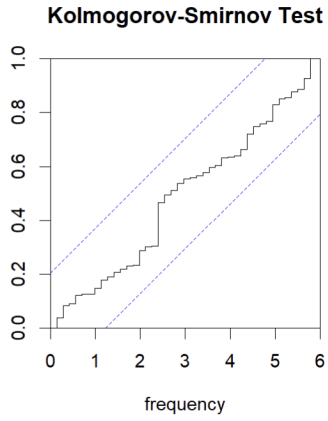
I used spectral analysis to check the periodogram of my residuals and also compare it to my original data. If my model is a good fit, the residuals should be white noise and have no strong peaks. If I see clear spikes in the periodogram of residuals, that's evidence of remaining periodic autocorrelation in my model didn't get removed. As my model residual periodogram shows, the spectral peaks are significantly reduced. I compare the periodogram of original data vs residuals. My original data shows dominant low-frequency or seasonal peaks at frequency = 1/12. My SARIMA model's residuals flattened those peaks (as seen by the y-axis) and ensures that the model has effectively captured the key cycles for both trend and seasonal structure.











Conclusion

This project analyzed the monthly unemployment rate using SARIMA time series modeling, with the aim of capturing both trend and seasonal dynamics in the data. The series showed clear downward movement and strong seasonal fluctuations, which were addressed through appropriate differencing and model specification. After careful diagnostic evaluation and model comparison, the SARIMA(1,0,1)(0,0,1)[12] model was selected based on its AICc value, residuals, and strong performance on the Ljung-Box test. However, a small spike at frequency 0.2 in the residual periodogram suggests the model may not have fully captured all cyclical behavior in the data.

To improve the model further, incorporating Fourier terms could be a valuable strategy. Fourier terms allow for modeling of seasonality and periodic components beyond the rigid integer-lag structure of SARIMA. In particular, when seasonal behavior is not perfectly captured by a seasonal MA or AR term at lag 12, Fourier terms can help approximate more complex or smooth periodic patterns. In this case, the persistent spike at frequency 0.2 suggests the presence of a recurring structure that might not align neatly with seasonal lag operators but could be approximated by sine and cosine waves. If this project were repeated, testing SARIMA models augmented with Fourier terms (e.g., K = 1 or 2) would allow for better periodic structure without overfitting and may better explain the remaining variation in the frequency domain.

References

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: With R examples* (4th ed.). Springer. https://doi.org/10.1007/978-3-319-52452-8

StackExchange. (2015). *SARIMA model equation*. Cross Validated. https://stats.stackexchange.com/questions/129901/sarima-model-equation

Santra, R. (2020, February 7). *Introduction to SARIMA model*. Medium. https://medium.com/@ritusantra/introduction-to-sarima-model-cbb885ceabe8

Appendix

Final Project

Roxanne Dimadi

2025-03-12

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(stats)
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.4.2
```

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

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.4.2
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(astsa)
```

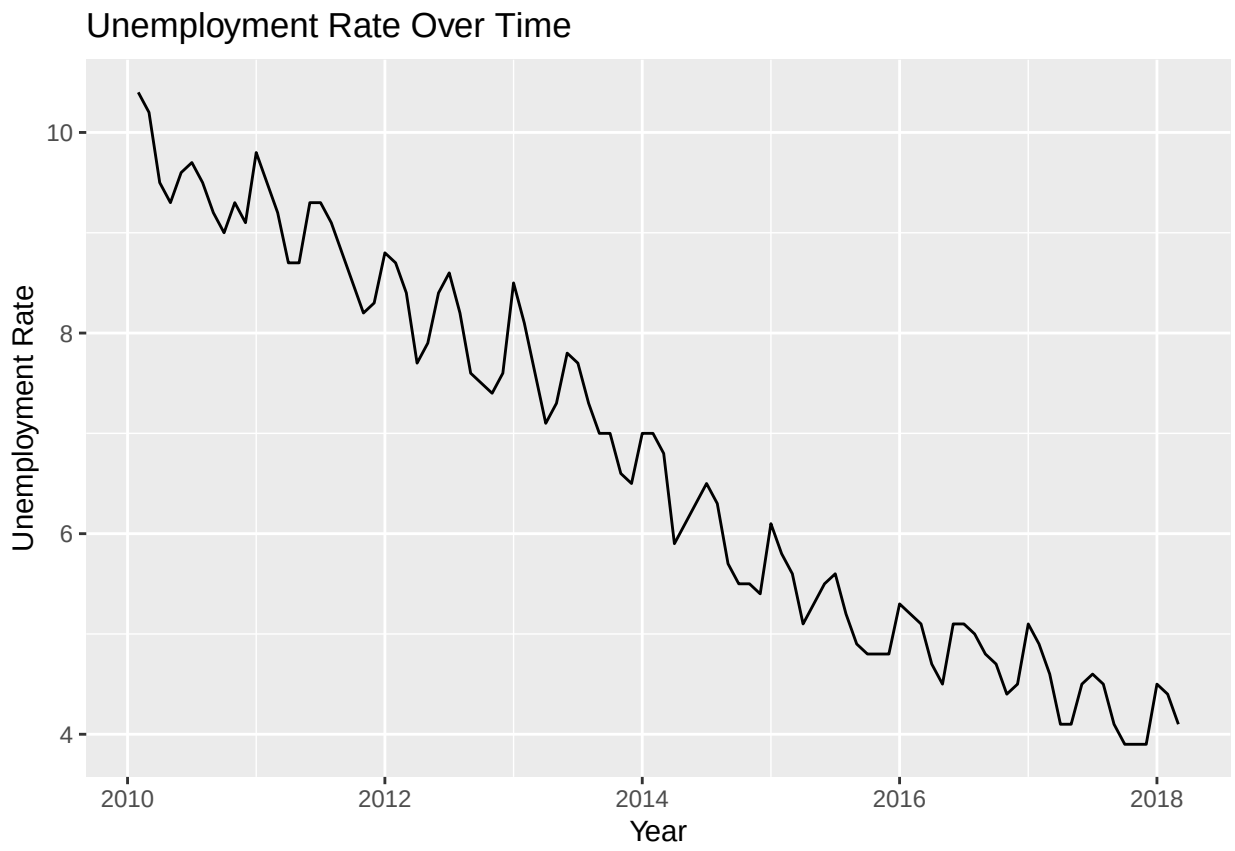
```
##
## Attaching package: 'astsa'
##
## The following object is masked from 'package:forecast':
##
##     gas
```

```
library(ggplot2)
library(readr)

unemp_ds <- read_csv("UNRATENSA.csv")
```

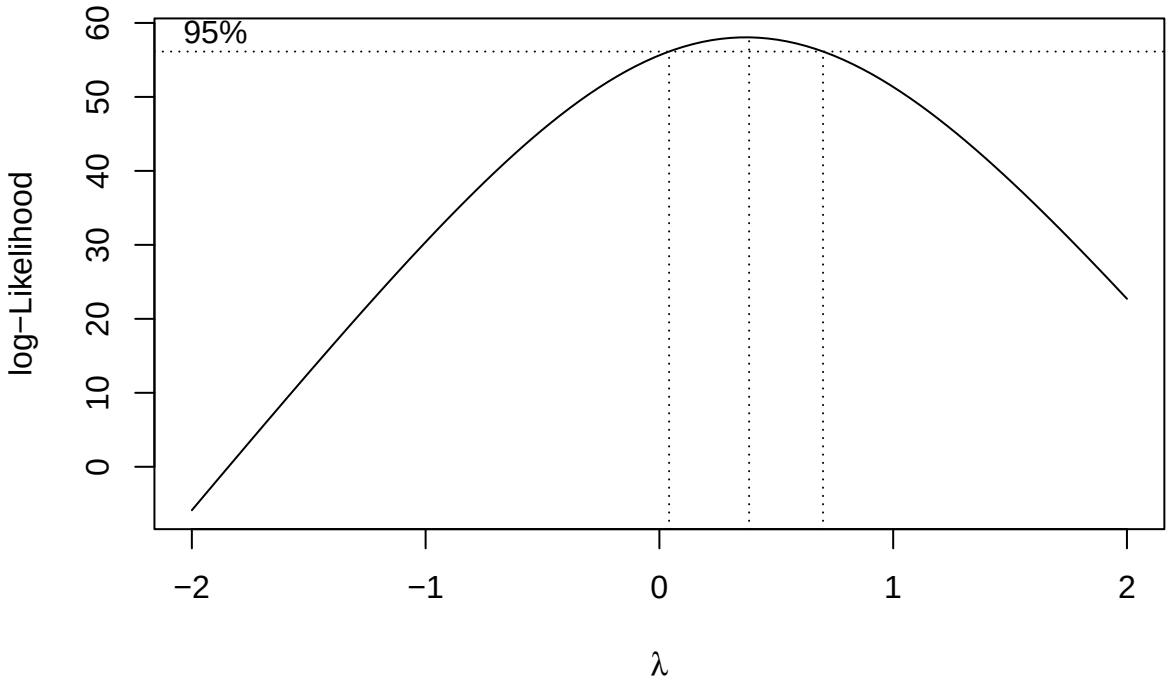
```
## Rows: 98 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): UNRATENSA
## date (1): observation_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Convert the date column to Date format
unemp_ds$observation_date <- as.Date(unemp_ds$observation_date)

# create a time series object (monthly data)
unemp_ts <- ts(unemp_ds$UNRATENSA, start = c(2010, 2), frequency = 12)
```

```
# Basic Time Series Plot
autoplot(unemp_ts) +
  ggtitle("Unemployment Rate Over Time") +
  ylab("Unemployment Rate") +
  xlab("Year")
```




```
# Box-Cox Log Likelihood
time <- 1:length(unemp_ts)
lambda <- boxcox(unemp_ts ~ time, plotit = T)
```

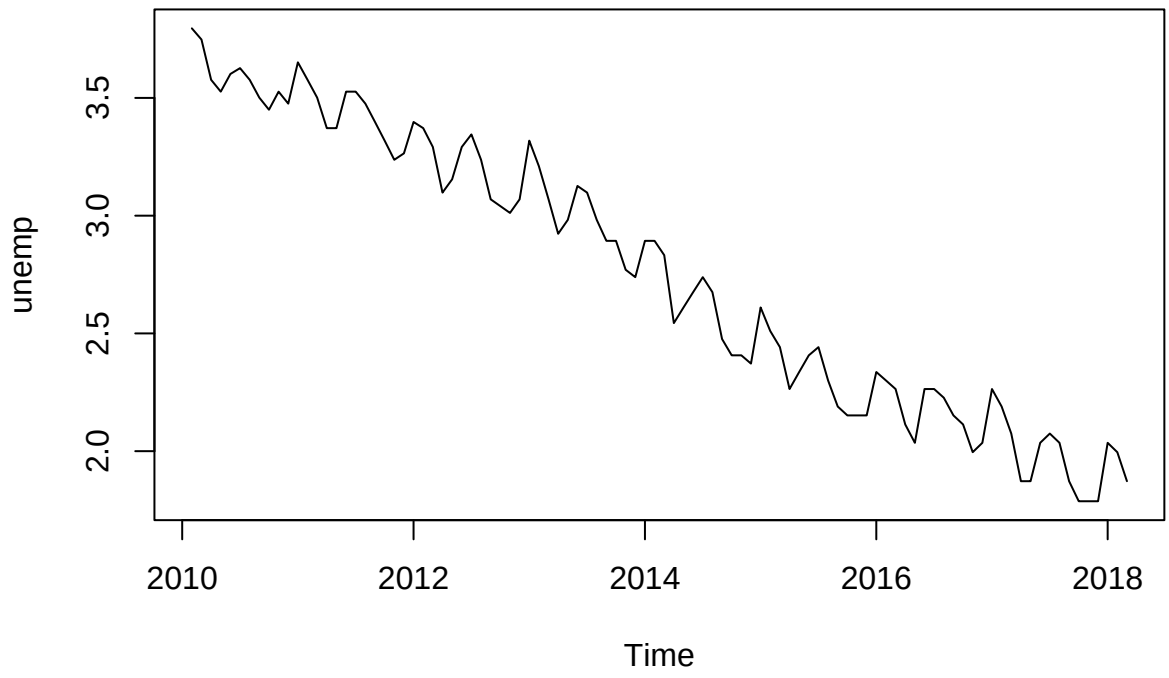



```
lambda.bc <- lambda$x[which.max(lambda$y)]
lambda.bc
```

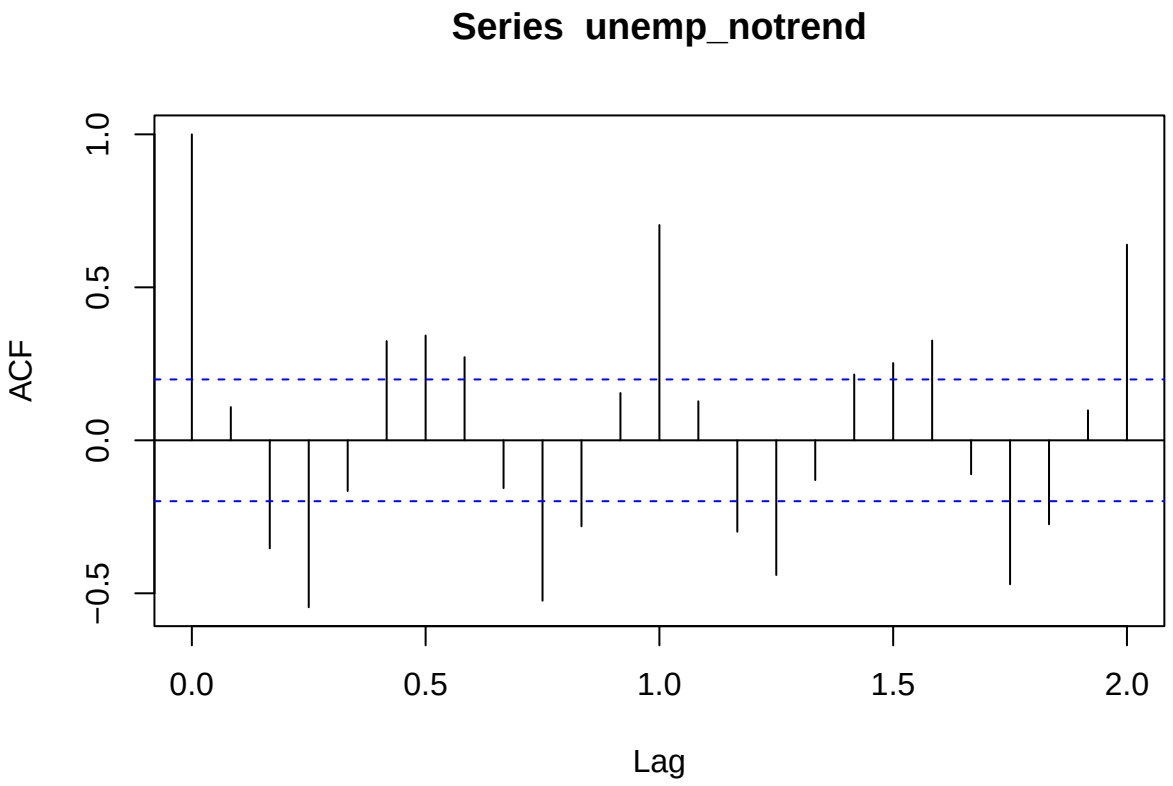
```
## [1] 0.3838384
```

The 95% confidence interval for lambda does not include 1, which means I can statistically reject the idea that no transformation (i.e., raw data) is sufficient.

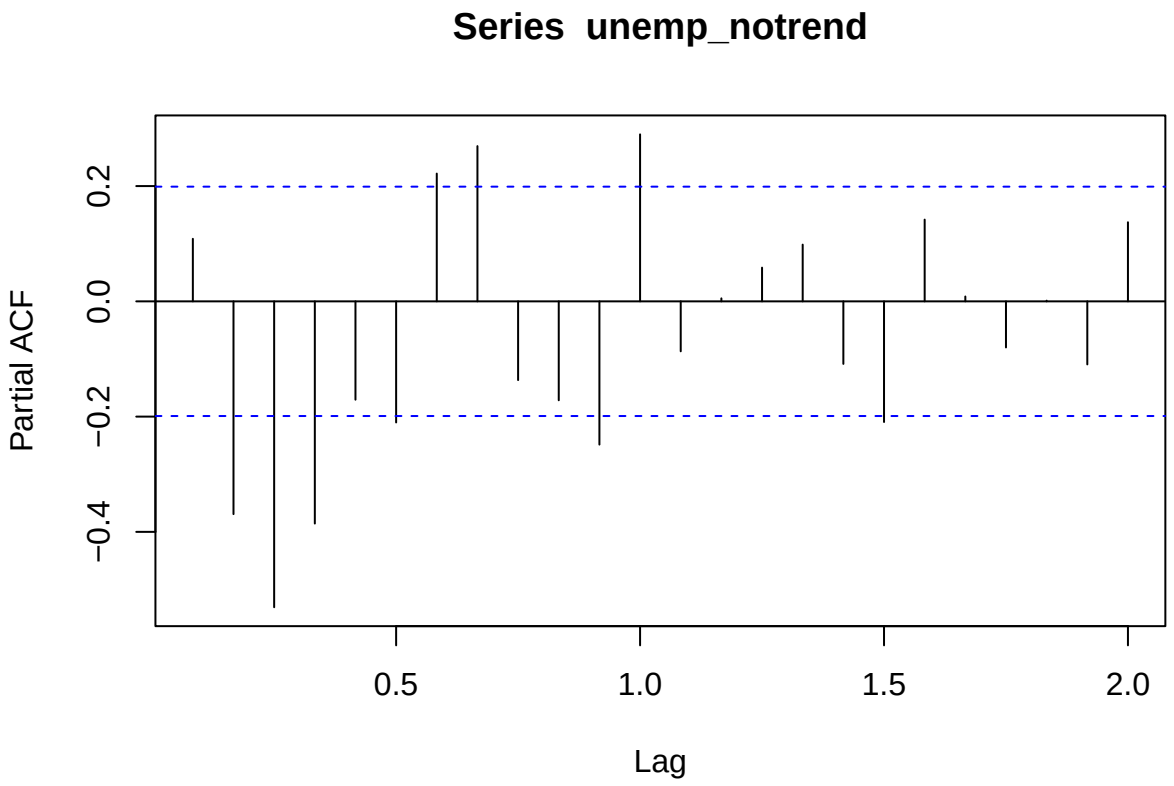
```
# Applying Box Cox transformation

unemp <- (((unemp_ts)^lambda.bc)-1)/ lambda.bc
ts.plot(unemp)
```



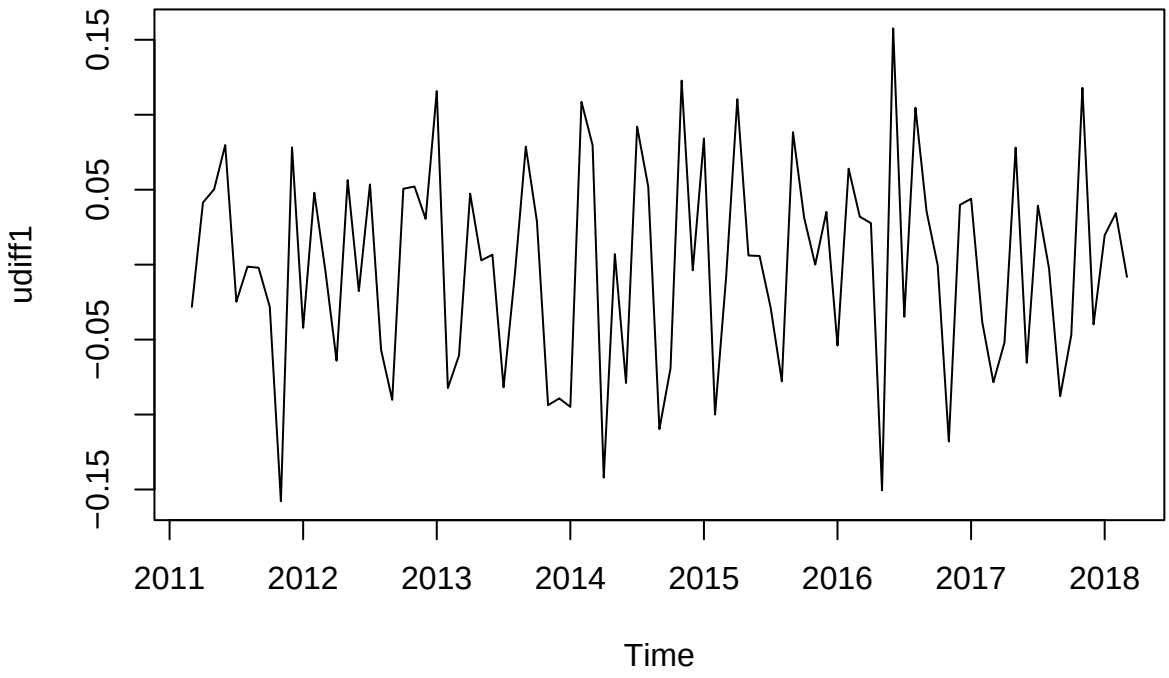

```
# remove seasonality
unemp_seas_diff <- diff(unemp, 12)

# try mid-seasonalality
unemp_midseas_diff <- diff(unemp, 5)

# remove trend
udiff1 <- diff(unemp_seas_diff, 1)

# try removing trend for mid-seasonal difference
udiff2 <- diff(unemp_midseas_diff, 1)

# removing just the trend
unemp_notrend <- diff(unemp, 1)
acf(unemp_notrend, lag.max = 24) # seasonal possible AR process
```



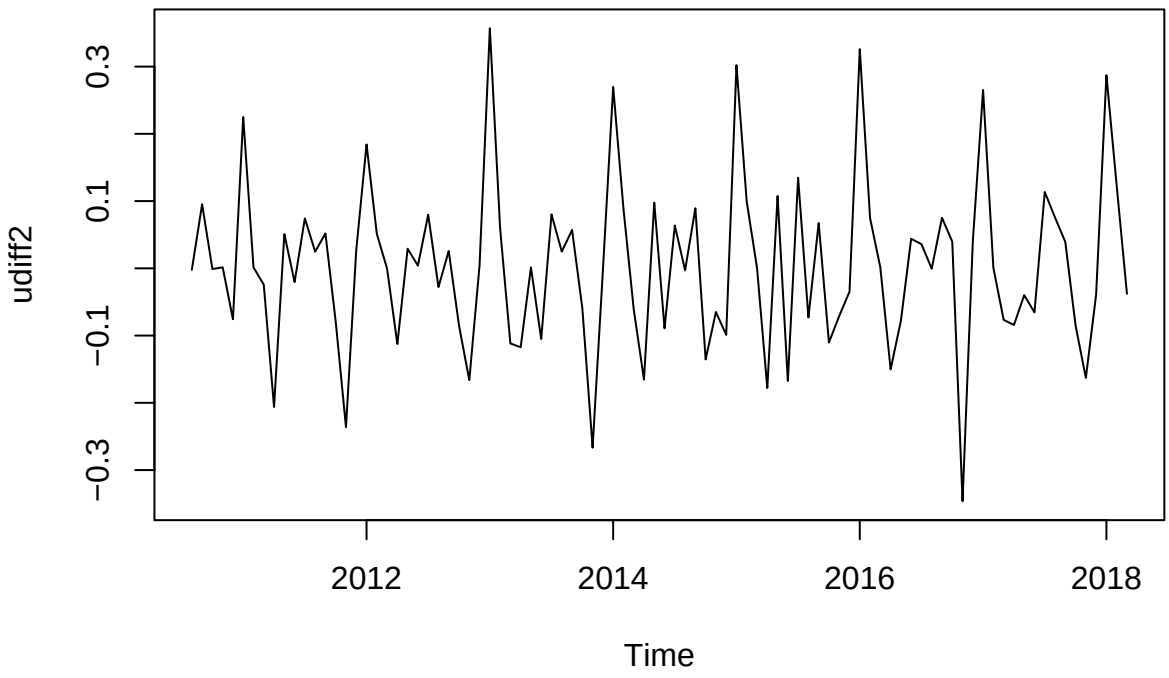

```
pacf(unemp_notrend, lag.max = 24) # probably AR (1) non seasonal
```




```
# plot and check adf

ts.plot(udiff1)
```




```
ts.plot(udiff2)
```




```
adf.test(udiff1)
```

```
## Warning in adf.test(udiff1): p-value smaller than printed p-value
```

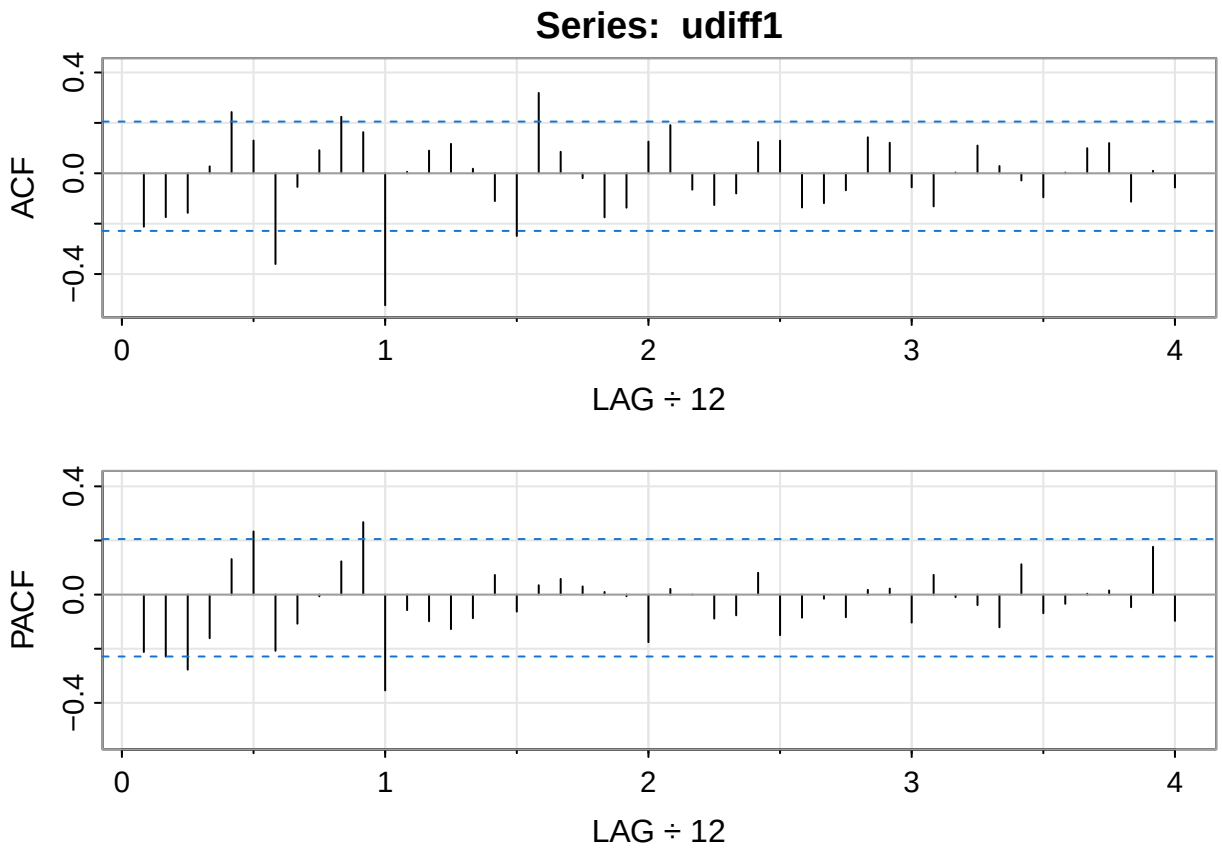
```
##
##  Augmented Dickey-Fuller Test
##
## data:  udiff1
## Dickey-Fuller = -4.9095, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(udiff2)
```

```
## Warning in adf.test(udiff2): p-value smaller than printed p-value
```

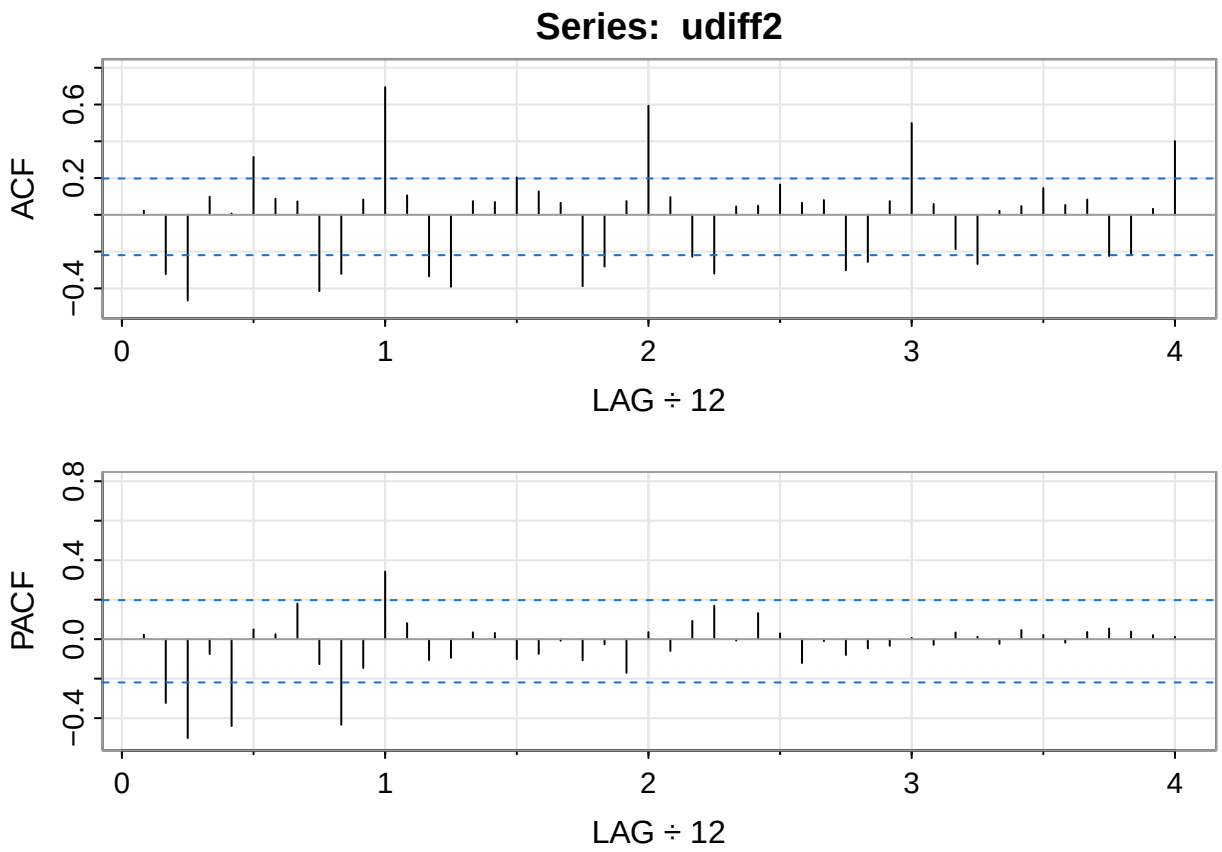
```
##
##  Augmented Dickey-Fuller Test
##
## data:  udiff2
## Dickey-Fuller = -9.7633, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
# let's check and make sure nothing is over-differenced
acf2(udiff1)
```



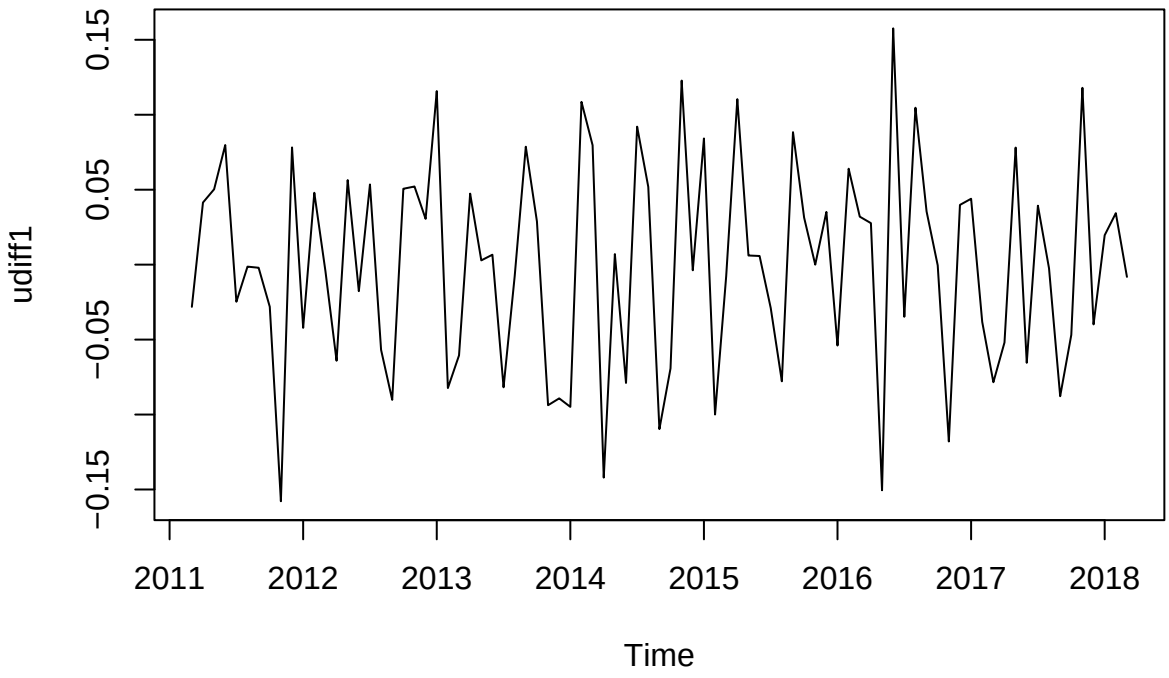

```
##       [,1]  [,2]  [,3]  [,4] [,5] [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.21 -0.17 -0.16  0.03 0.24 0.13 -0.36 -0.05  0.09  0.22  0.16 -0.52
## PACF -0.21 -0.23 -0.28 -0.16 0.13 0.23 -0.21 -0.11 -0.01  0.12  0.27 -0.35
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.01  0.09  0.12  0.02 -0.11 -0.25  0.32  0.09 -0.02 -0.17 -0.14  0.13
## PACF -0.06 -0.10 -0.13 -0.09  0.07 -0.06  0.03  0.06  0.03  0.01 -0.01 -0.18
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF   0.19 -0.07 -0.13 -0.08  0.12  0.13 -0.14 -0.12 -0.07  0.14  0.12 -0.06
## PACF  0.02  0.00 -0.09 -0.08  0.08 -0.15 -0.09 -0.02 -0.08  0.02  0.02 -0.10
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.13  0.00  0.11  0.03 -0.03 -0.10  0.00   0.1  0.12 -0.11  0.01 -0.06
## PACF  0.07 -0.01 -0.04 -0.12  0.11 -0.07 -0.03   0.0  0.02 -0.05  0.18 -0.10
```

```
acf2(udiff2)
```



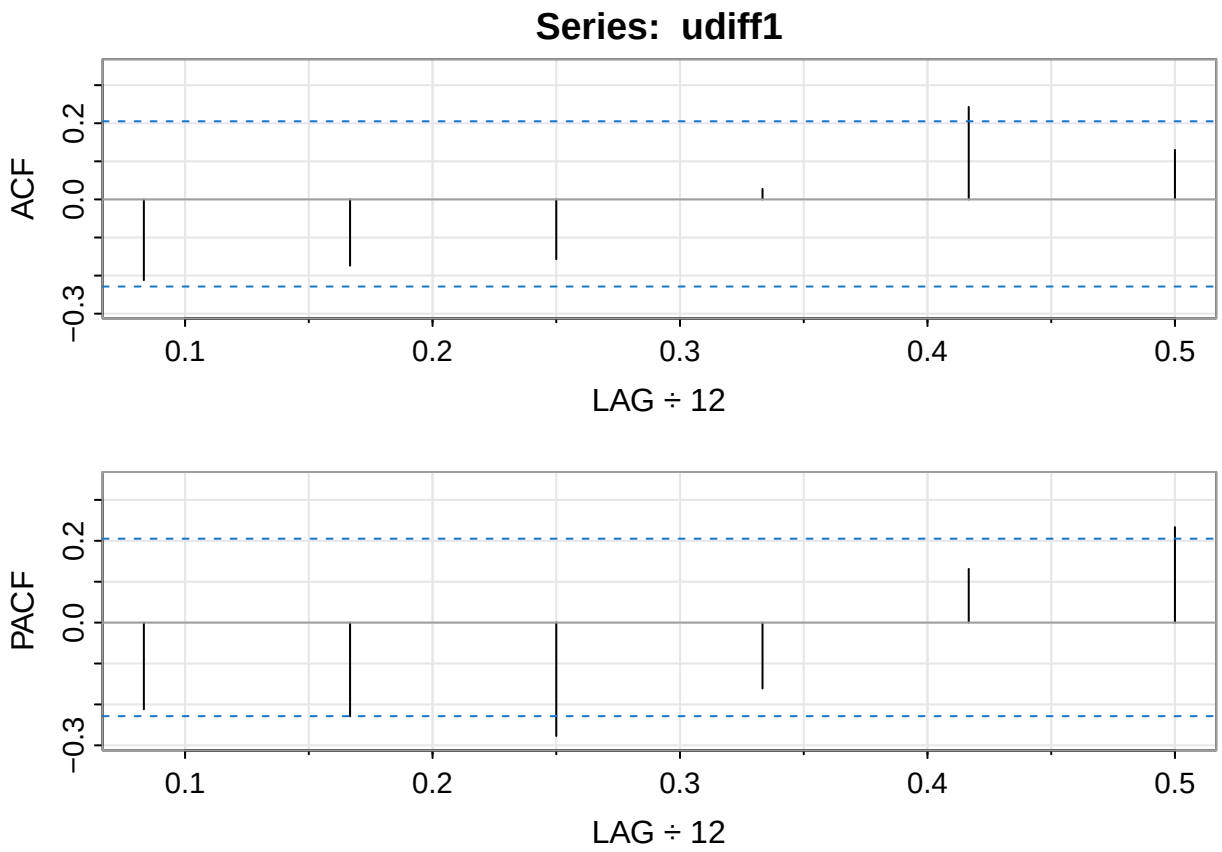

```
##      [,1]  [,2]  [,3]  [,4]  [,5] [,6] [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.02 -0.32 -0.47  0.10  0.01 0.31 0.09 0.07 -0.41 -0.32  0.08  0.69  0.11
## PACF 0.02 -0.32 -0.50 -0.08 -0.44 0.05 0.03 0.18 -0.13 -0.43 -0.15  0.34  0.08
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.33 -0.39  0.07  0.07   0.2  0.13  0.07 -0.39 -0.28  0.07  0.59  0.10
## PACF -0.11 -0.09  0.03  0.03  -0.1 -0.07 -0.01 -0.11 -0.03 -0.17  0.04 -0.06
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF  -0.23 -0.32  0.05  0.05  0.17  0.07  0.08 -0.30 -0.26  0.07  0.50  0.06
## PACF  0.09  0.17 -0.01  0.13  0.03 -0.12 -0.01 -0.08 -0.05 -0.03  0.01 -0.03
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.19 -0.27  0.02  0.05  0.15  0.05  0.08 -0.22 -0.22  0.03  0.40
## PACF  0.03  0.01 -0.02  0.05  0.02 -0.02  0.04  0.05  0.04  0.02  0.01
```

```
# probably better to go with udiff1
# ts.plot(udiff2)
ts.plot(udiff1)
```



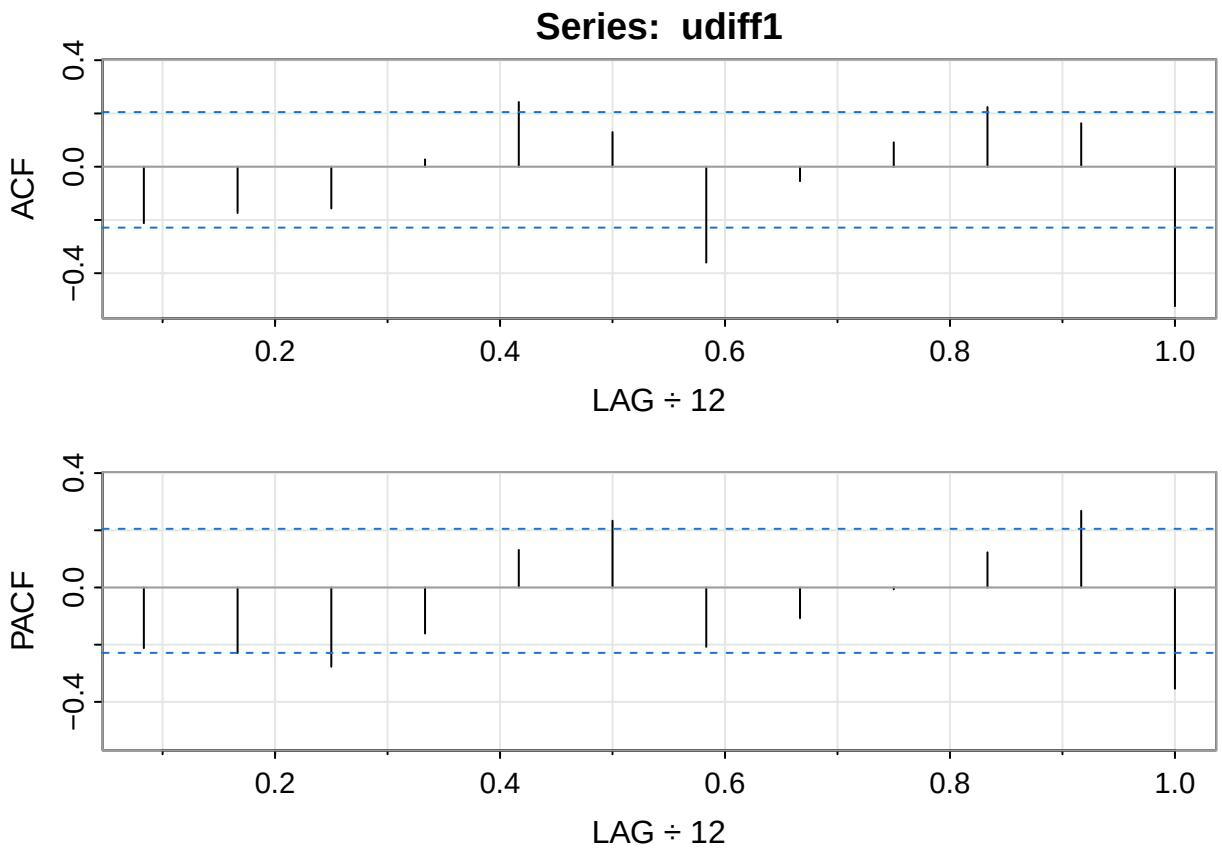
Now, I choose the order of my SARIMA model

```
# use ACF/PACF to pick SARIMA(p,d,q)(P,D,Q) model order
par(mfrow=c(1,2))
acf2(udiff1, max = 6)
```



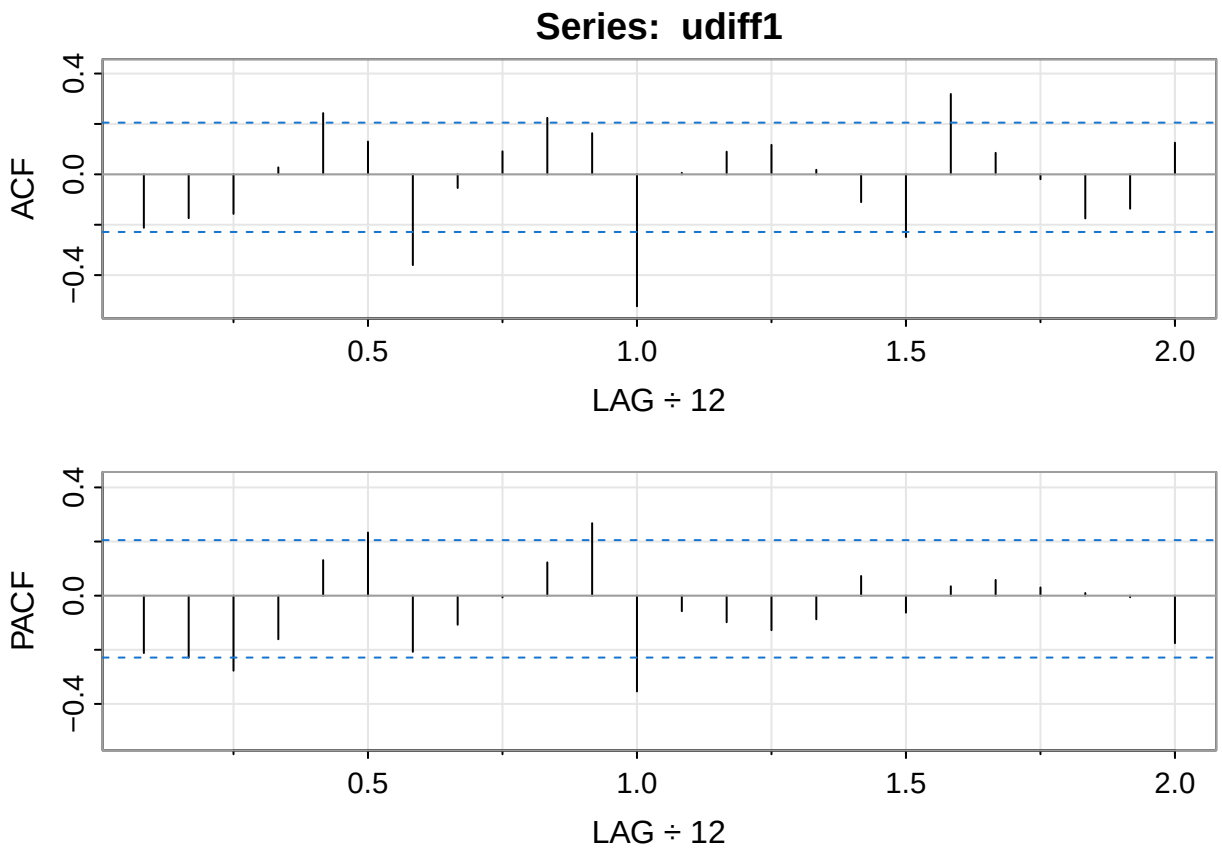

```
##       [,1]  [,2]  [,3]  [,4] [,5] [,6]
## ACF  -0.21 -0.17 -0.16  0.03 0.24 0.13
## PACF -0.21 -0.23 -0.28 -0.16 0.13 0.23
```

```
acf2(udiff1, max = 12)
```



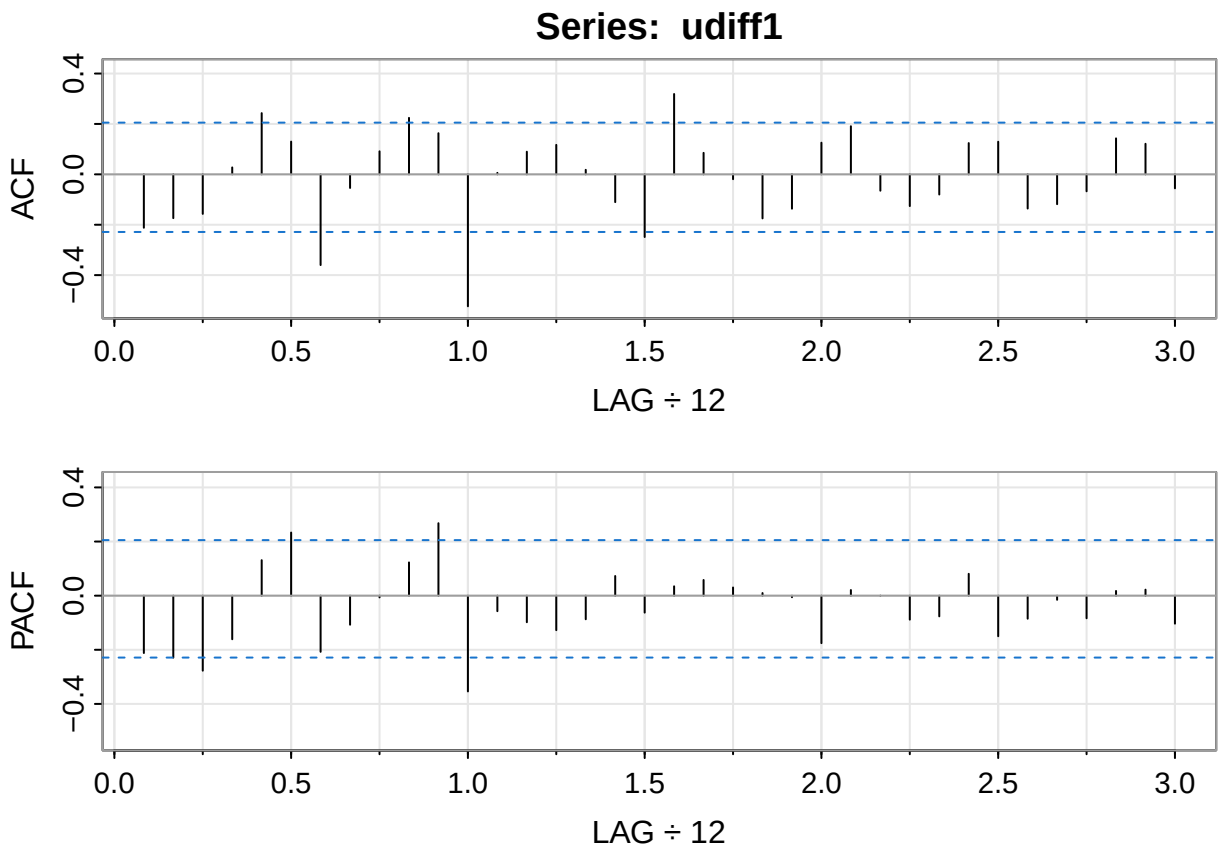

```
##       [,1]  [,2]  [,3]  [,4] [,5] [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.21 -0.17 -0.16  0.03 0.24 0.13 -0.36 -0.05  0.09  0.22  0.16 -0.52
## PACF -0.21 -0.23 -0.28 -0.16 0.13 0.23 -0.21 -0.11 -0.01  0.12  0.27 -0.35
```

```
acf2(udiff1, max = 24)
```



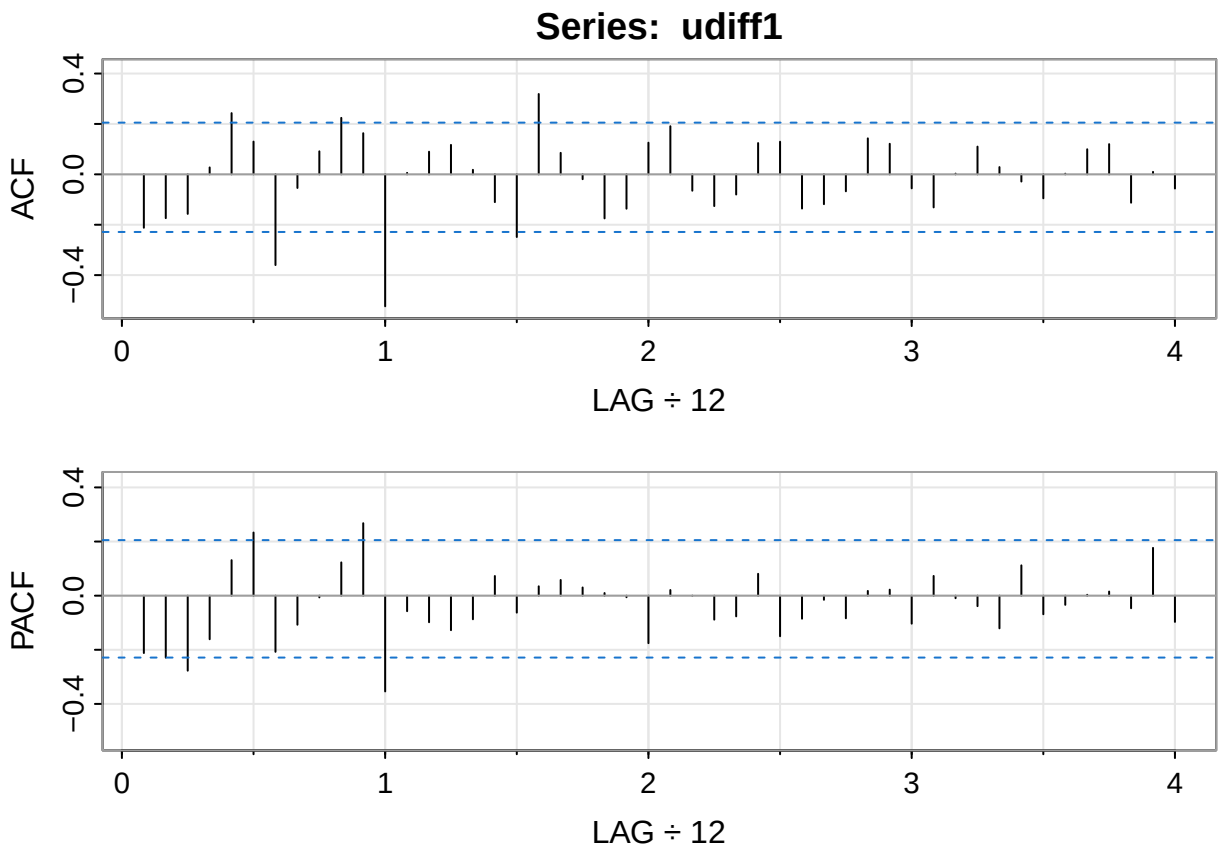

```
##       [,1]  [,2]  [,3]  [,4] [,5] [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.21 -0.17 -0.16  0.03 0.24 0.13 -0.36 -0.05  0.09  0.22  0.16 -0.52
## PACF -0.21 -0.23 -0.28 -0.16 0.13 0.23 -0.21 -0.11 -0.01  0.12  0.27 -0.35
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.01  0.09  0.12  0.02 -0.11 -0.25  0.32  0.09 -0.02 -0.17 -0.14  0.13
## PACF -0.06 -0.10 -0.13 -0.09  0.07 -0.06  0.03  0.06  0.03  0.01 -0.01 -0.18
```

```
acf2(udiff1, max = 36)
```



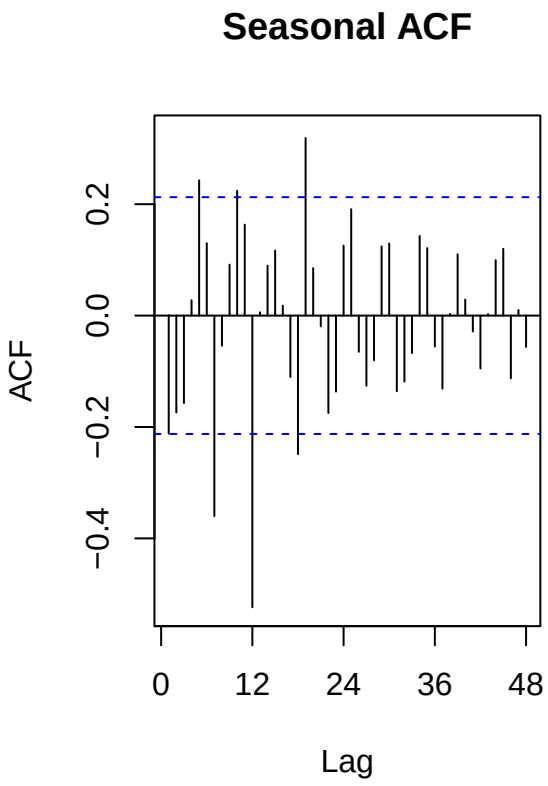

```
##       [,1]  [,2]  [,3]  [,4] [,5] [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.21 -0.17 -0.16  0.03 0.24 0.13 -0.36 -0.05  0.09  0.22  0.16 -0.52
## PACF -0.21 -0.23 -0.28 -0.16 0.13 0.23 -0.21 -0.11 -0.01  0.12  0.27 -0.35
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.01  0.09  0.12  0.02 -0.11 -0.25  0.32  0.09 -0.02 -0.17 -0.14  0.13
## PACF -0.06 -0.10 -0.13 -0.09  0.07 -0.06  0.03  0.06  0.03  0.01 -0.01 -0.18
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF   0.19 -0.07 -0.13 -0.08  0.12  0.13 -0.14 -0.12 -0.07  0.14  0.12 -0.06
## PACF  0.02  0.00 -0.09 -0.08  0.08 -0.15 -0.09 -0.02 -0.08  0.02  0.02 -0.10
```

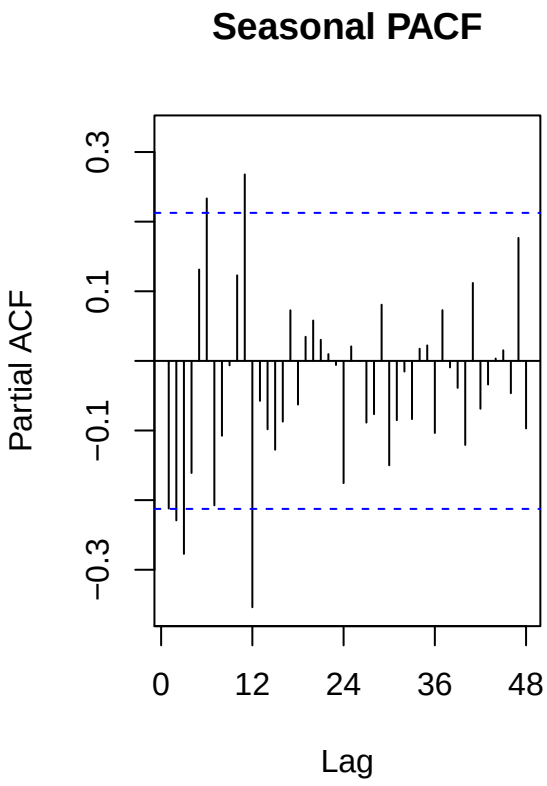
```
acf2(udiff1, max = 48)
```




```
##       [,1]  [,2]  [,3]  [,4] [,5] [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.21 -0.17 -0.16  0.03 0.24 0.13 -0.36 -0.05  0.09  0.22  0.16 -0.52
## PACF -0.21 -0.23 -0.28 -0.16 0.13 0.23 -0.21 -0.11 -0.01  0.12  0.27 -0.35
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.01  0.09  0.12  0.02 -0.11 -0.25  0.32  0.09 -0.02 -0.17 -0.14  0.13
## PACF -0.06 -0.10 -0.13 -0.09  0.07 -0.06  0.03  0.06  0.03  0.01 -0.01 -0.18
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF   0.19 -0.07 -0.13 -0.08  0.12  0.13 -0.14 -0.12 -0.07  0.14  0.12 -0.06
## PACF  0.02  0.00 -0.09 -0.08  0.08 -0.15 -0.09 -0.02 -0.08  0.02  0.02 -0.10
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.13  0.00  0.11  0.03 -0.03 -0.10  0.00   0.1  0.12 -0.11  0.01 -0.06
## PACF  0.07 -0.01 -0.04 -0.12  0.11 -0.07 -0.03   0.0  0.02 -0.05  0.18 -0.10
```

```r
Acf(udiff1, lag.max=48, main="Seasonal ACF")
# AR order Q = 1

Pacf(udiff1, lag.max=48, main="Seasonal PACF")
```



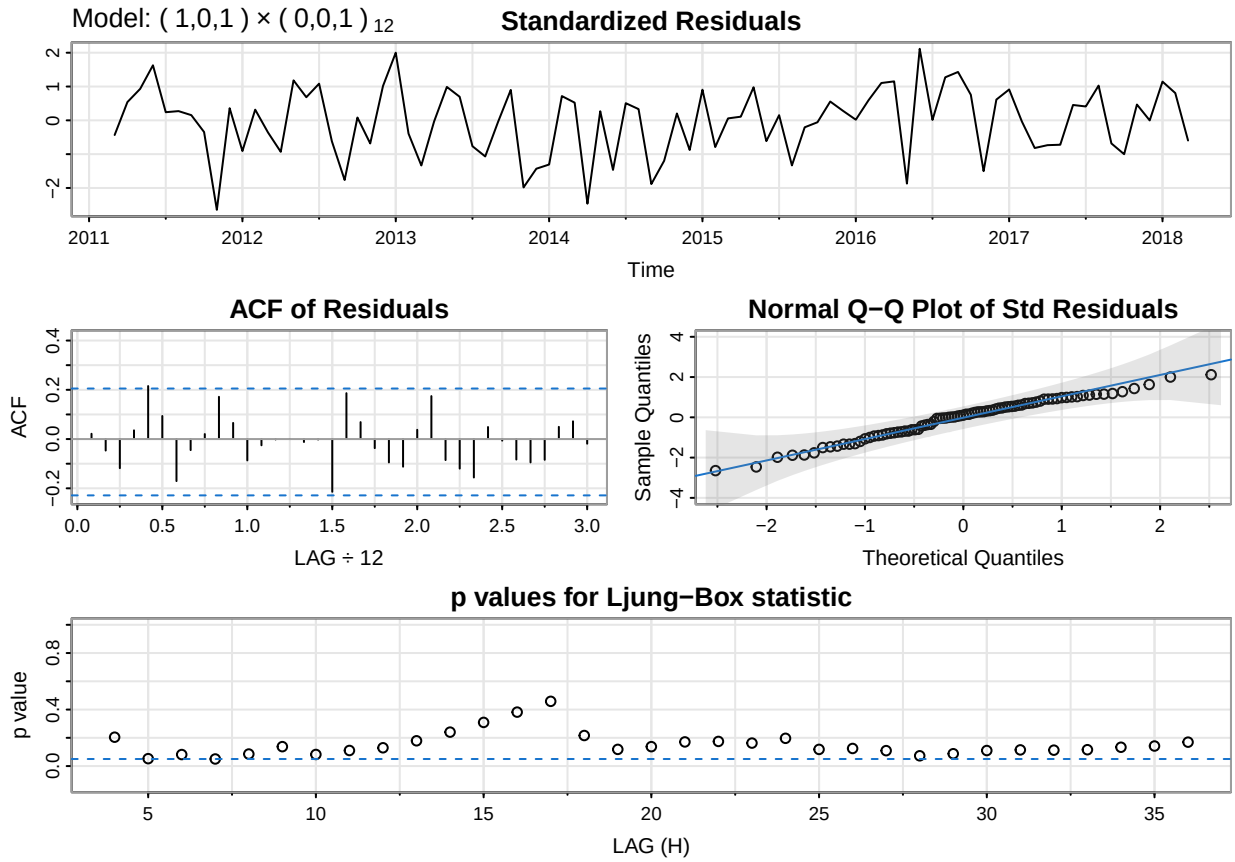



```
# AR order P = 1
```

```
# Candidate 1: SARIMA(1,0,1)(0,0,1)[12]
sarima(udiff1, 1,0,1, 0,0,1, 12)
```

```
## initial  value -2.659603
## iter   2 value -2.877918
## iter   3 value -2.891343
## iter   4 value -2.893402
## iter   5 value -2.894304
## iter   6 value -2.905021
## iter   7 value -2.906495
## iter   8 value -2.909057
## iter   9 value -2.912832
## iter  10 value -2.915580
## iter  11 value -2.916603
## iter  12 value -2.917262
## iter  13 value -2.917871
## iter  14 value -2.918903
## iter  15 value -2.919448
## iter  16 value -2.919631
## iter  17 value -2.919665
## iter  18 value -2.919672
## iter  19 value -2.919673
## iter  19 value -2.919673
```

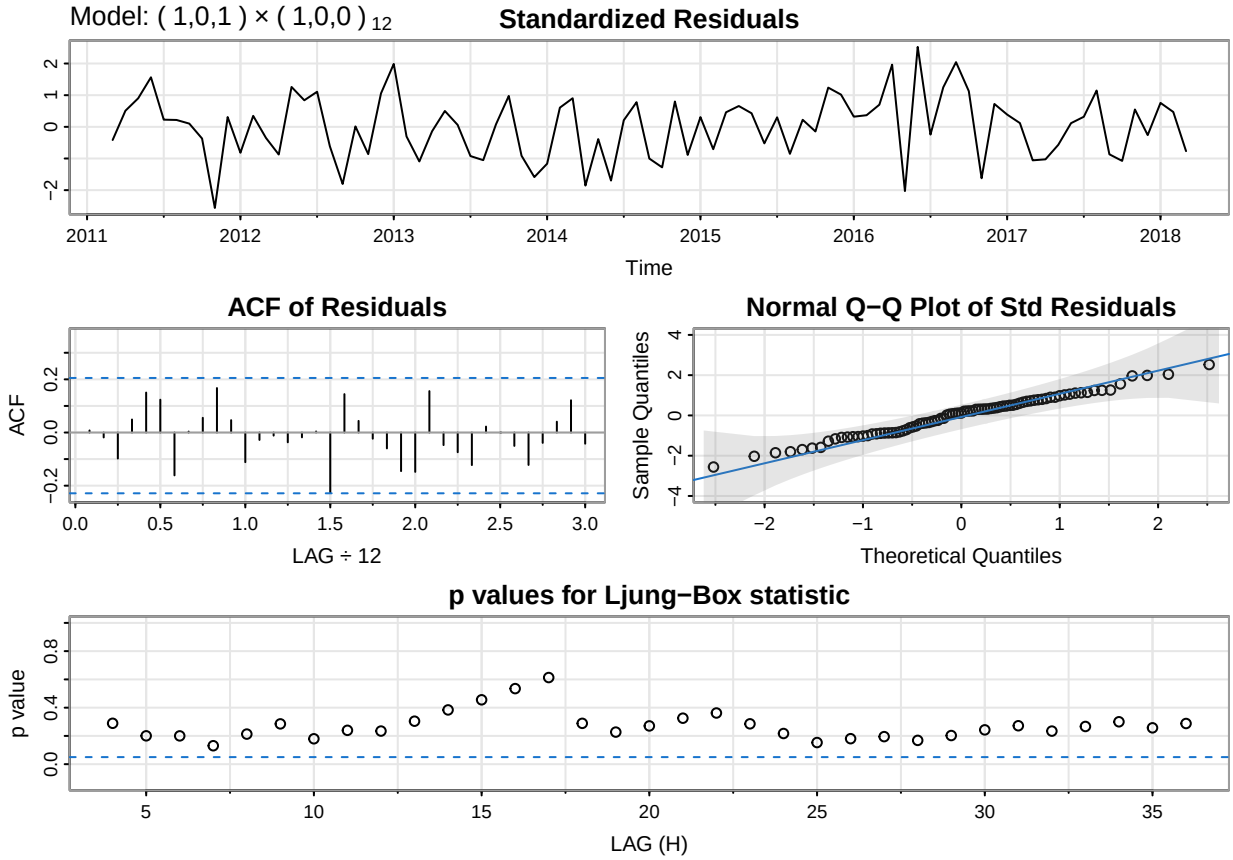
```
## final  value -2.919673
## converged
## initial  value -2.955270
## iter   2 value -2.956956
## iter   3 value -2.972576
## iter   4 value -2.979444
## iter   5 value -2.986336
## iter   6 value -2.986490
## iter   7 value -2.986509
## iter   8 value -2.986514
## iter   9 value -2.986514
## iter  10 value -2.986514
## iter  10 value -2.986514
## final  value -2.986514
## converged
## <><><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ar1     0.3380 0.1849  1.8282  0.0712
## ma1    -0.6684 0.1357 -4.9255  0.0000
## sma1   -0.9998 0.1618 -6.1778  0.0000
## xmean   0.0001 0.0010  0.1422  0.8873
##
## sigma^2 estimated as 0.001891738 on 81 degrees of freedom
##
## AIC = -3.017504  AICc = -3.011622  BIC = -2.873819
##
```




```
# AICc = -3.011622

# Candidate 2: SARIMA(1,0,1)(1,0,0)[12]
sarima(udiff1, 1,0,1, 1,0,0, 12)
```

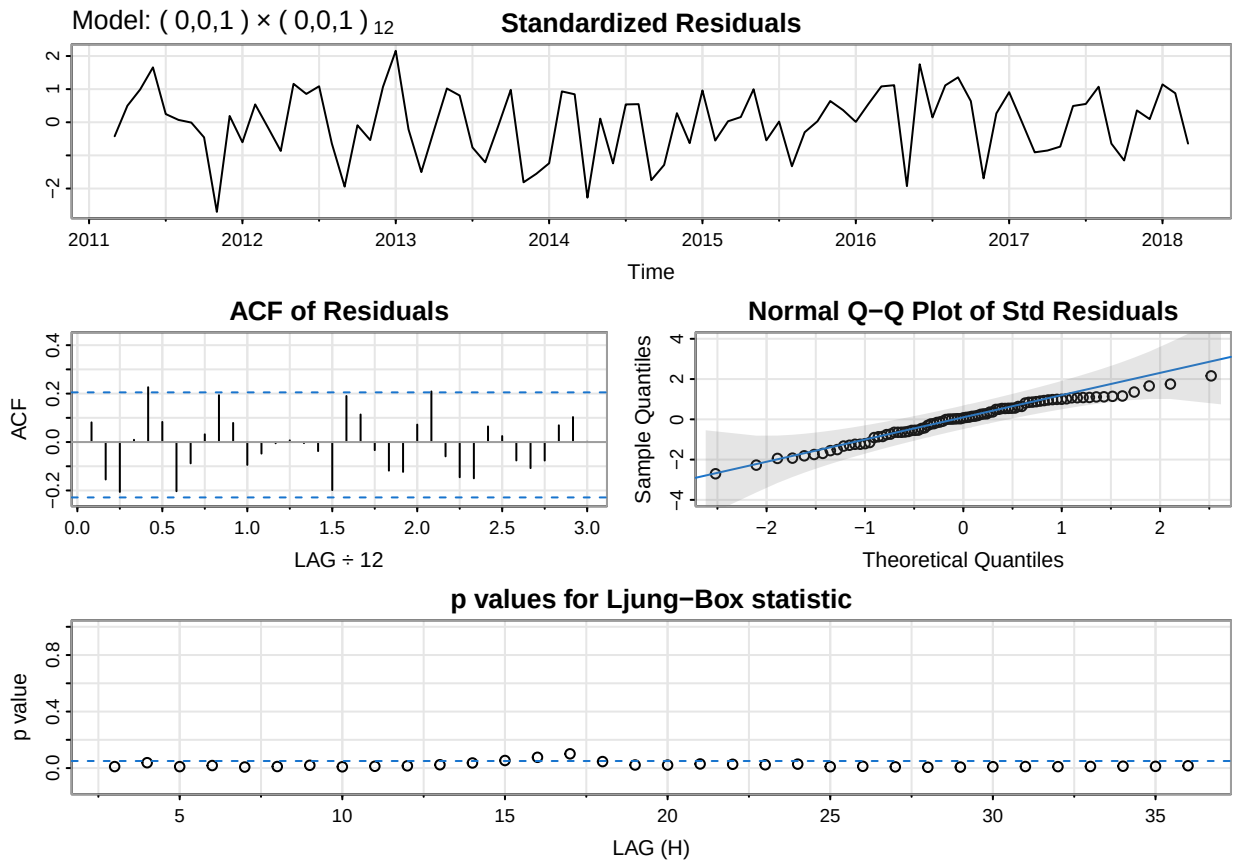
```
## initial  value -2.643373
## iter   2 value -2.895958
## iter   3 value -2.897906
## iter   4 value -2.901262
## iter   5 value -2.906040
## iter   6 value -2.916195
## iter   7 value -2.918404
## iter   8 value -2.921439
## iter   9 value -2.921819
## iter  10 value -2.922127
## iter  11 value -2.922133
## iter  12 value -2.922134
## iter  13 value -2.922136
## iter  14 value -2.922136
## iter  15 value -2.922136
## iter  15 value -2.922136
## iter  15 value -2.922136
## final  value -2.922136
## converged
## initial  value -2.903977
## iter   2 value -2.904130
```

```
## iter   3 value -2.904385
## iter   4 value -2.904529
## iter   5 value -2.904680
## iter   6 value -2.904732
## iter   7 value -2.904742
## iter   8 value -2.904742
## iter   8 value -2.904742
## iter   8 value -2.904742
## final  value -2.904742
## converged
## <><><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ar1     0.2688 0.2022  1.3294  0.1875
## ma1    -0.6070 0.1570 -3.8653  0.0002
## sar1   -0.5624 0.0885 -6.3572  0.0000
## xmean   0.0003 0.0021  0.1361  0.8921
##
## sigma^2 estimated as 0.002836192 on 81 degrees of freedom
##
## AIC = -2.853959  AICc = -2.848077  BIC = -2.710274
##
```



```r
#AICc = -2.848077

# Candidate 3: SARIMA(0,0,1)(0,0,1)[12]
sarima(udiff1, 0,0,1, 0,0,1, 12)
```

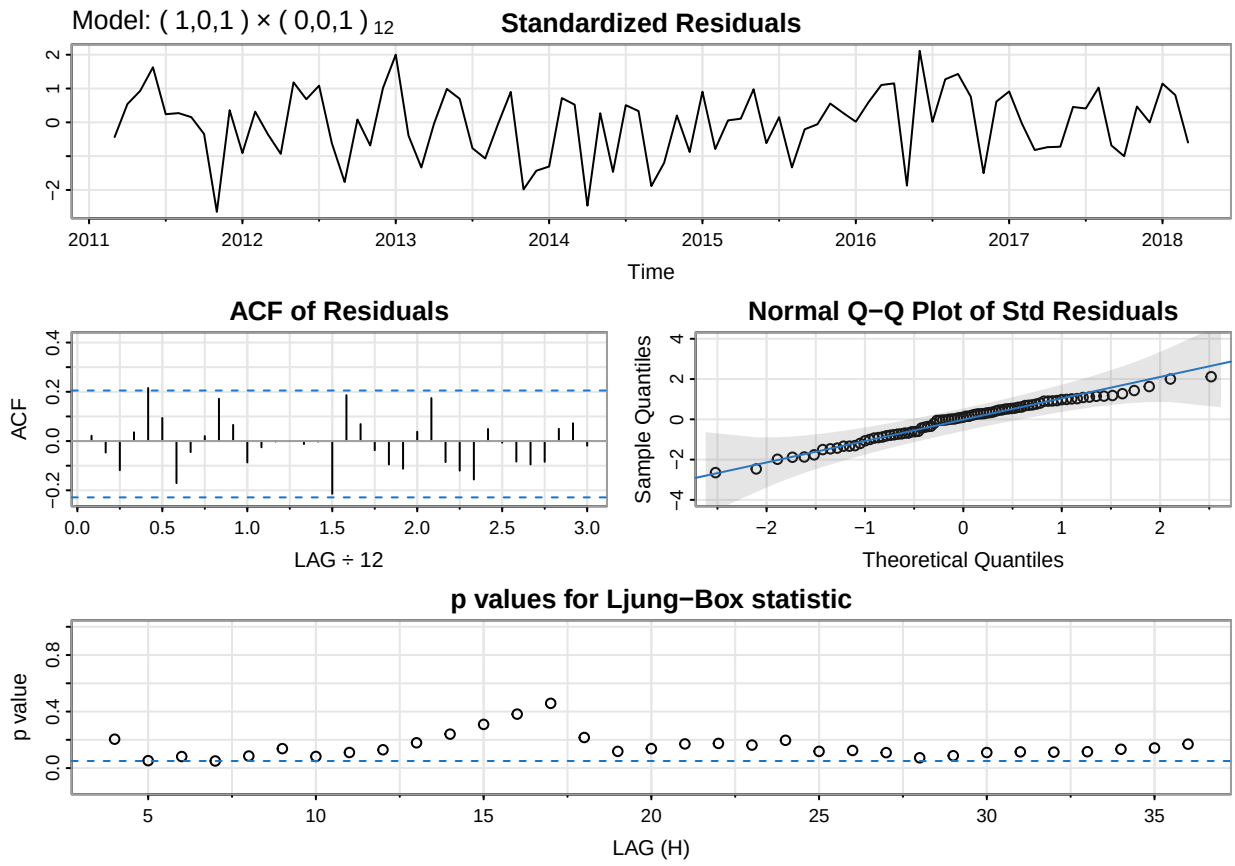
```
## initial  value -2.664545
## iter   2 value -2.891864
## iter   3 value -2.904447
## iter   4 value -2.908016
## iter   5 value -2.910893
## iter   6 value -2.913476
## iter   7 value -2.913736
## iter   8 value -2.913760
## iter   9 value -2.913761
## iter  10 value -2.913761
## iter  11 value -2.913761
## iter  11 value -2.913761
## iter  11 value -2.913761
## final  value -2.913761
## converged
## initial  value -2.938161
## iter   2 value -2.948994
## iter   3 value -2.963702
## iter   4 value -2.967327
## iter   5 value -2.969356
## iter   6 value -2.969385
## iter   7 value -2.969387
## iter   8 value -2.969387
## iter   8 value -2.969387
## final  value -2.969387
## converged
## <><><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ma1    -0.3904 0.1333 -2.9294  0.0044
## sma1   -1.0000 0.1720 -5.8145  0.0000
## xmean   0.0003 0.0012  0.2115  0.8330
##
## sigma^2 estimated as 0.001958415 on 82 degrees of freedom
##
## AIC = -3.00678  AICc = -3.003294  BIC = -2.891832
##
```



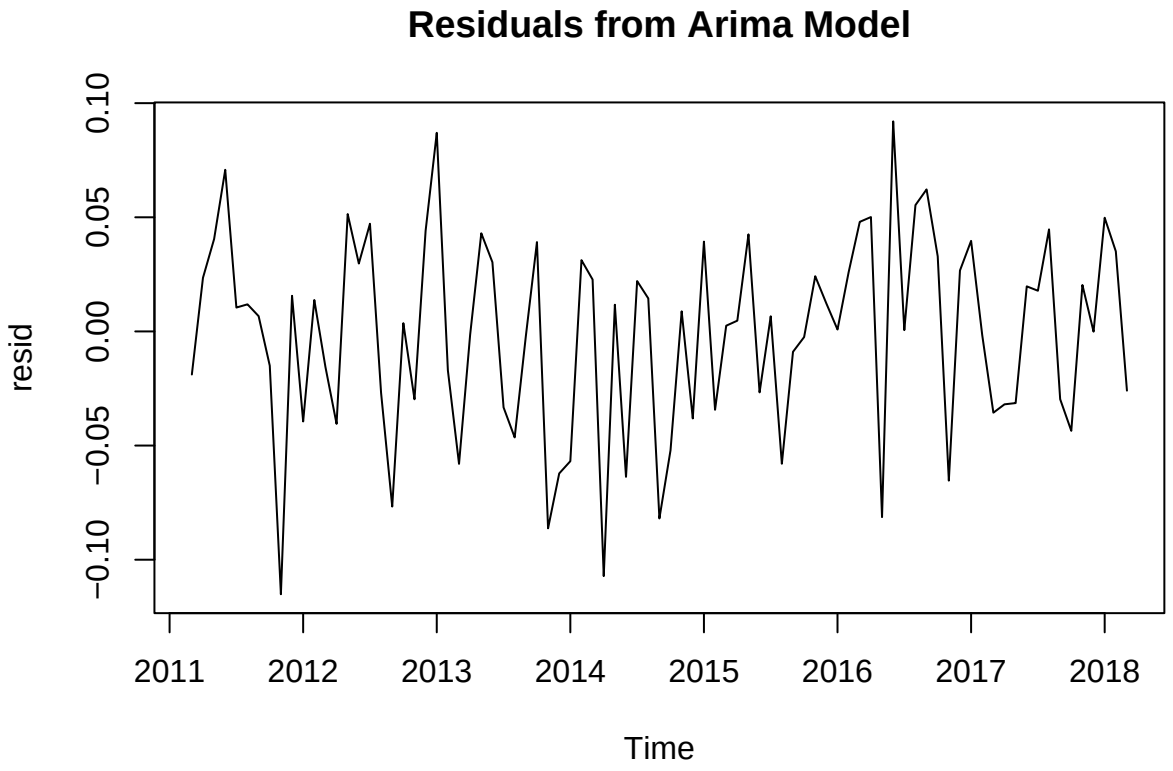

```
#AICc = -3.003294

# Candidate 4: SARIMA(1,0,1)(0,0,1)[12]
sarima(udiff1, 1,0,1, 0,0,1, 12)
```

```
## initial  value -2.659603
## iter   2 value -2.877918
## iter   3 value -2.891343
## iter   4 value -2.893402
## iter   5 value -2.894304
## iter   6 value -2.905021
## iter   7 value -2.906495
## iter   8 value -2.909057
## iter   9 value -2.912832
## iter  10 value -2.915580
## iter  11 value -2.916603
## iter  12 value -2.917262
## iter  13 value -2.917871
## iter  14 value -2.918903
## iter  15 value -2.919448
## iter  16 value -2.919631
## iter  17 value -2.919665
## iter  18 value -2.919672
## iter  19 value -2.919673
## iter  19 value -2.919673
## final  value -2.919673
```

```
## converged
## initial  value -2.955270
## iter   2 value -2.956956
## iter   3 value -2.972576
## iter   4 value -2.979444
## iter   5 value -2.986336
## iter   6 value -2.986490
## iter   7 value -2.986509
## iter   8 value -2.986514
## iter   9 value -2.986514
## iter  10 value -2.986514
## iter  10 value -2.986514
## final  value -2.986514
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate     SE t.value p.value
## ar1     0.3380 0.1849  1.8282  0.0712
## ma1    -0.6684 0.1357 -4.9255  0.0000
## sma1   -0.9998 0.1618 -6.1778  0.0000
## xmean   0.0001 0.0010  0.1422  0.8873
##
## sigma^2 estimated as 0.001891738 on 81 degrees of freedom
##
## AIC = -3.017504  AICc = -3.011622  BIC = -2.873819
##
```




```
#AICc = -3.001705



model_best <- Arima(udiff1, order=c(1,0,1), seasonal=list(order=c(0,0,1), period=12))
resid <- residuals(model_best)
plot(resid, main="Residuals from Arima Model")
```




```
library(TSA)
```

```
## Warning: package 'TSA' was built under R version 4.4.3
```

```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast
```

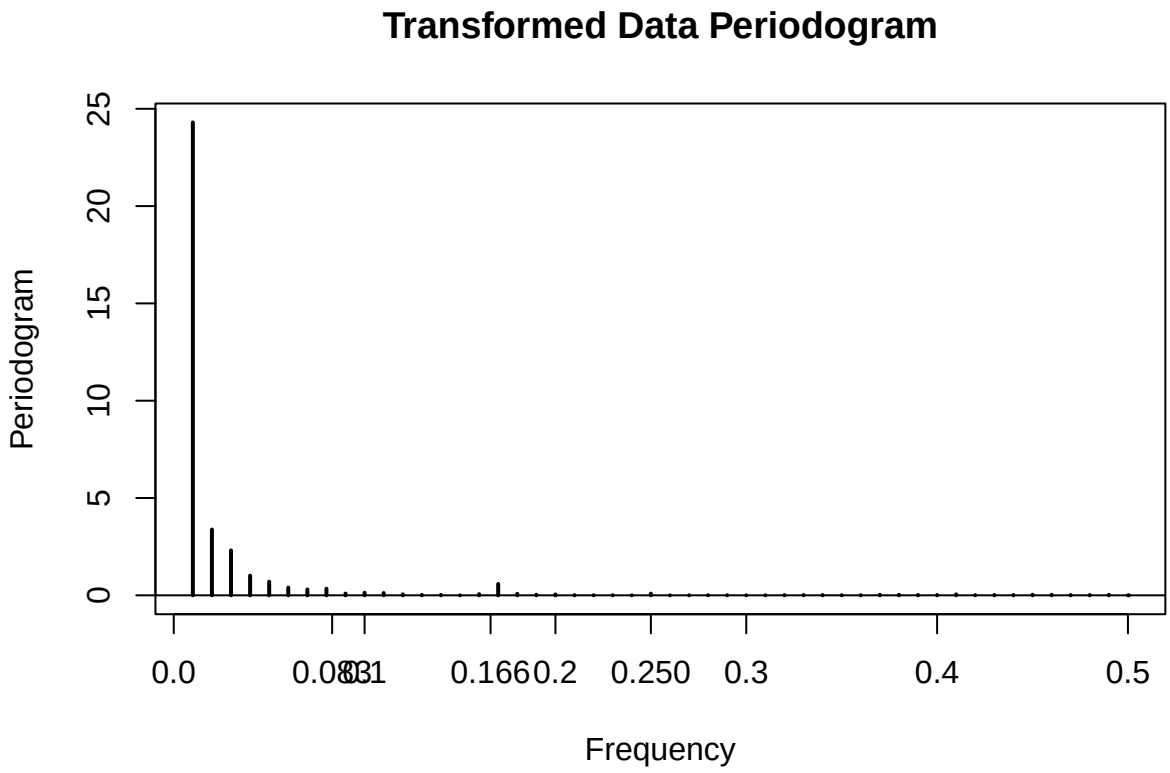
```
##
## Attaching package: 'TSA'
```

```
## The following object is masked from 'package:readr':
##
##     spec
```

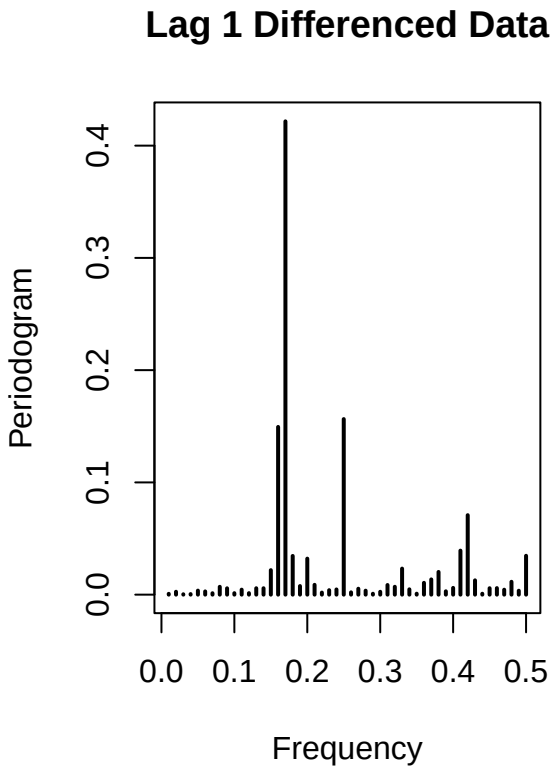
```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

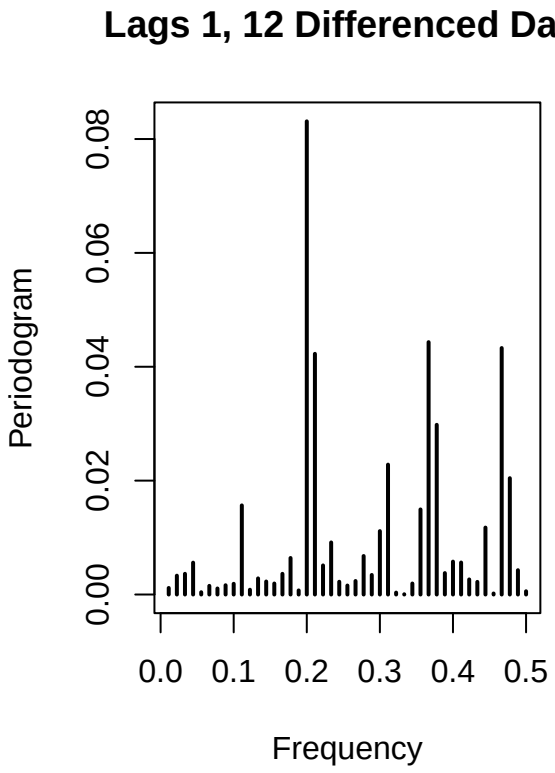
```
## The following object is masked from 'package:utils':
##
##     tar
```

```r
#establish baseline then look at periodogram

# periodogram of Box-Cox transformed data

per <- TSA::periodogram(unemp, main="Transformed Data Periodogram")

abline(h=0)

axis(1,at=c(0.083, 0.166, 0.25))
```

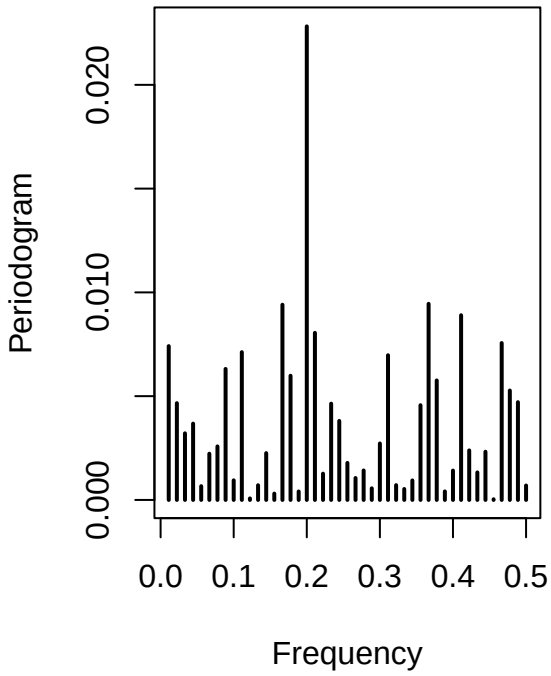


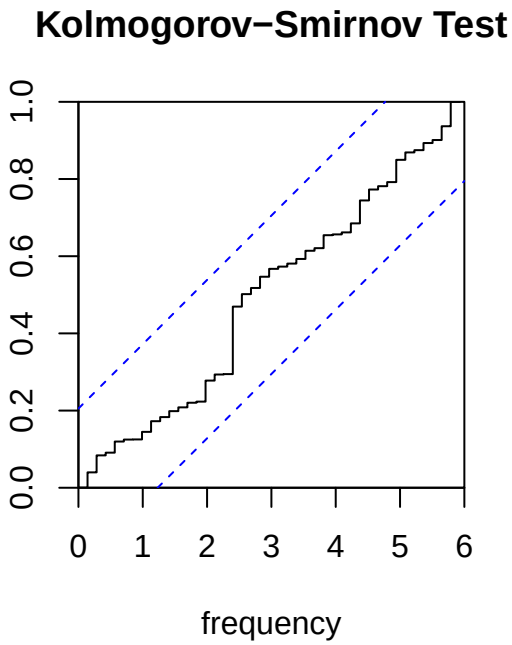

```r
# lag 1, lags 1,12 differenced data periodograms

par(mfrow=c(1,2))

TSA::periodogram(unemp_notrend, main="Lag 1 Differenced Data")

TSA::periodogram(udiff1, main="Lags 1, 12 Differenced Data")
```






```
# residual periodogram and Kolmogorov-Smirnov test

par(mfrow=c(1,2))

TSA::periodogram(resid)

cpgram(resid,main="Kolmogorov-Smirnov Test")
```


Periodogram
0.000
0.010
0.020
0.0
0.1
0.2
0.3
0.4
0.5
Frequency


Kolmogorov−Smirnov Test
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
frequency