

Time Series Analysis Project

Bill Sendewicz

December 8, 2020

Data Series 1: Wick Airport Rainfall

My first data series represents monthly rainfall measured at Wick John O' Groats Airport in Wick, Scotland. Estimated data is marked with an asterisk after the value. Missing data (more than 2 days missing in month) is marked by —.

In this case, none of the data are missing. However, there are a handful of estimated values (nine to be exact) in the 1269 months of data in the series; the asterisks associated with these estimates will have to be removed from the dataset.

Inspecting the data and preparing the data for modeling

```
series1 = read.table("wickairportdata.txt", skip=11, sep=" ", fill=T, strip.white=T,
                    dec=".", header=T, na.strings="NA", as.is=T, stringsAsFactors = F)

series1 <- series1[1:6] # remove last column

series1 <- series1[!(series1$yyyy == "Provisional"),] # remove rows with "Provisional"

series1 <- series1 %>% slice(-c(1)) # remove second row

series1 <- select(series1, -c(3:5)) # remove unwanted columns

colnames(series1) <- c("Year", "Month", "Rain") # rename columns

series1$Rain <- gsub("\\*", "", series1$Rain) # remove values with asterisks

series1$Year <- as.numeric(series1$Year) # convert Year column to numeric
series1$Month <- as.numeric(series1$Month) # convert Month column to numeric
series1$Rain <- as.numeric(series1$Rain) # convert Rain column to numeric

str(series1)
```

```
## 'data.frame':   1269 obs. of  3 variables:
##  $ Year : num  1914 1914 1914 1914 1914 ...
##  $ Month: num   1  2  3  4  5  6  7  8  9 10 ...
##  $ Rain : num  48.3 66 76.7 27.9 61 21.1 91.9 40.6 58.9 48.8 ...
```

```
head(series1)
```

```
##   Year Month Rain
## 1 1914     1 48.3
## 2 1914     2 66.0
## 3 1914     3 76.7
## 4 1914     4 27.9
## 5 1914     5 61.0
## 6 1914     6 21.1
```

```
tail(series1)
```

```
##   Year Month Rain
## 1264 2019     4 32.8
## 1265 2019     5 51.4
## 1266 2019     6 63.6
## 1267 2019     7 62.6
## 1268 2019     8 92.4
## 1269 2019     9 52.4
```

```
table(series1$Year) # to ensure that no data are missing
```

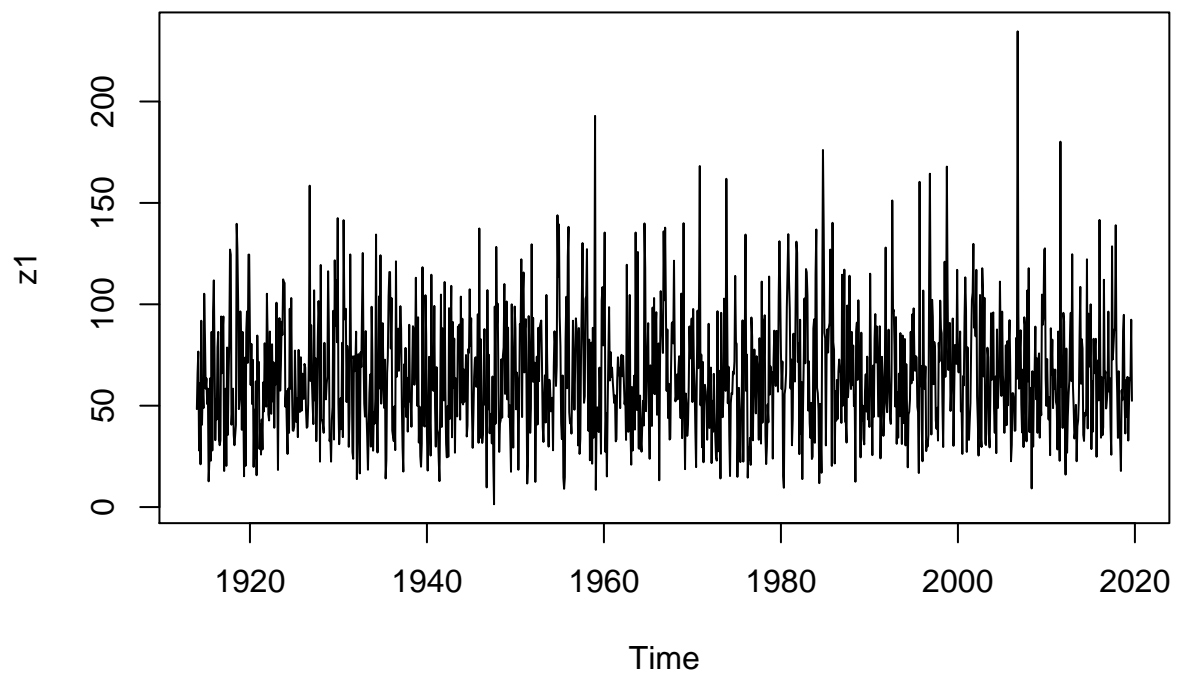
```
##
## 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929
##   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12
## 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945
##   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12
## 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961
##   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12
## 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977
##   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12
## 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993
##   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12
## 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009
##   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12
## 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
##   12   12   12   12   12   12   12   12   12   9
```

No data are missing in this series.

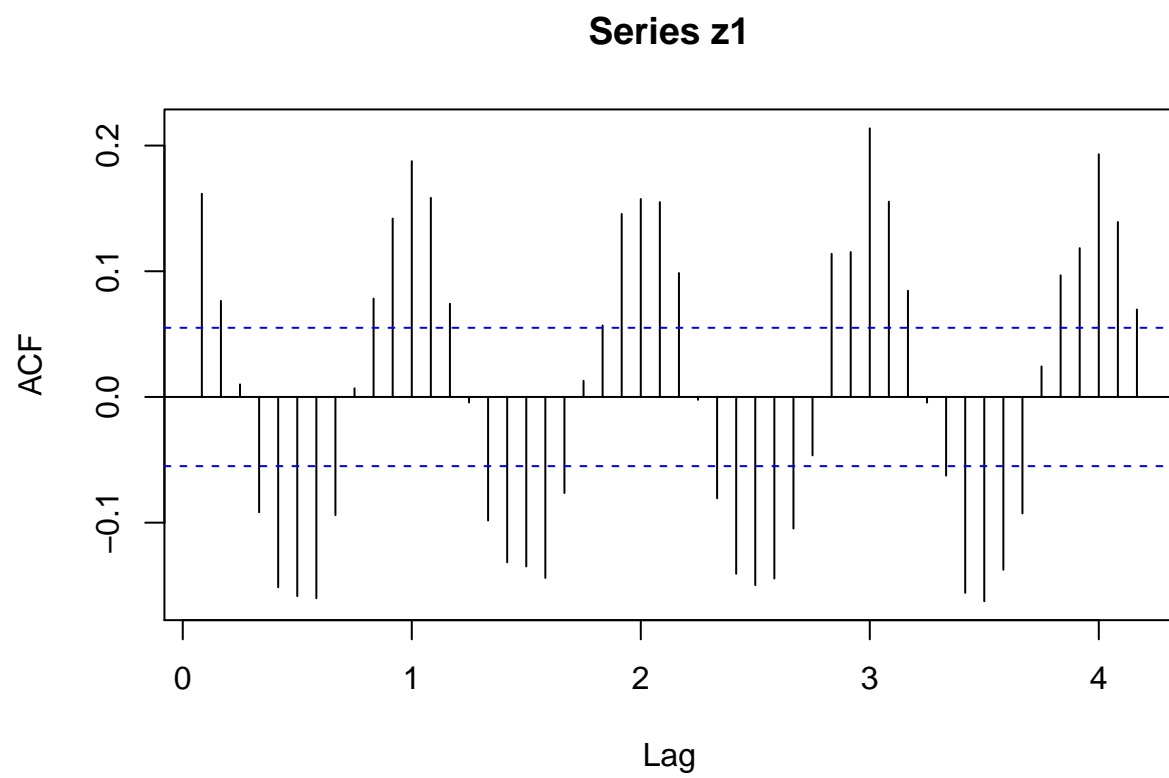
Let's examine the time series, its plot, autocorrelations plot and partial autocorrelations plot.

```
z1 <- ts(series1$Rain, frequency=12, start=c(series1$Year[1], series1$Month[1]))
```

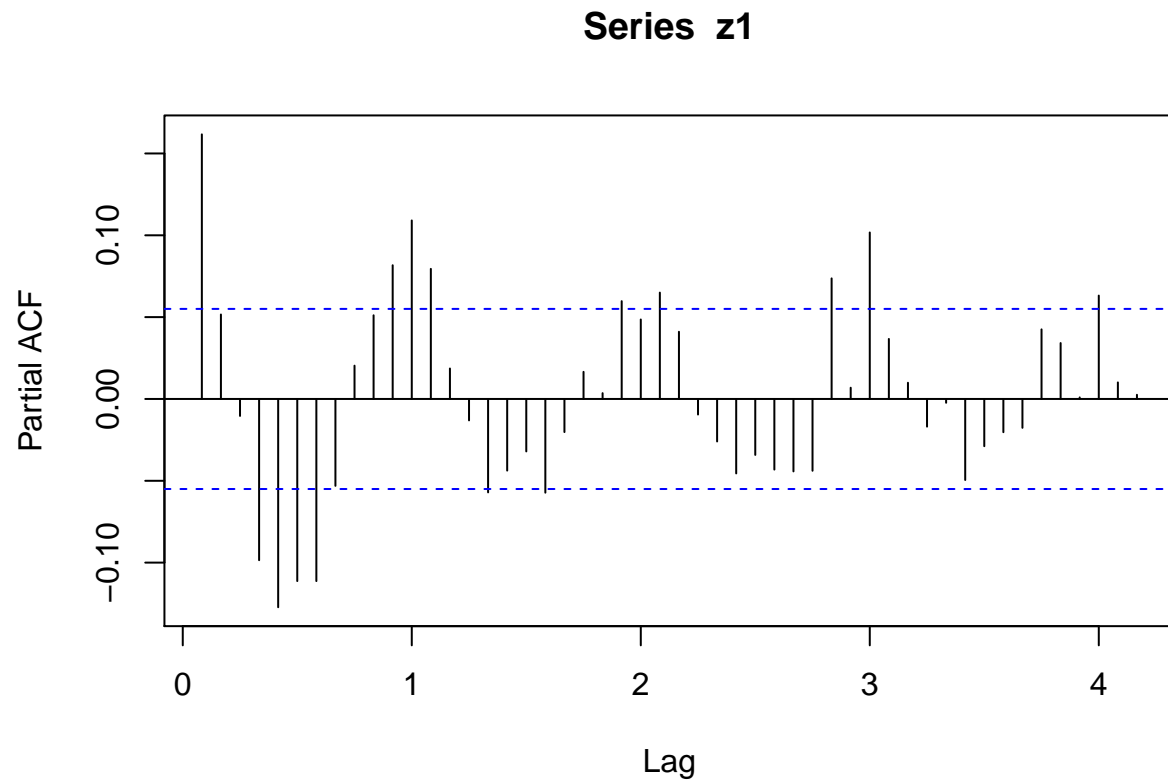
```
plot(z1)
```



```
acf(z1, 50)
```



```
pacf(z1, 50)
```



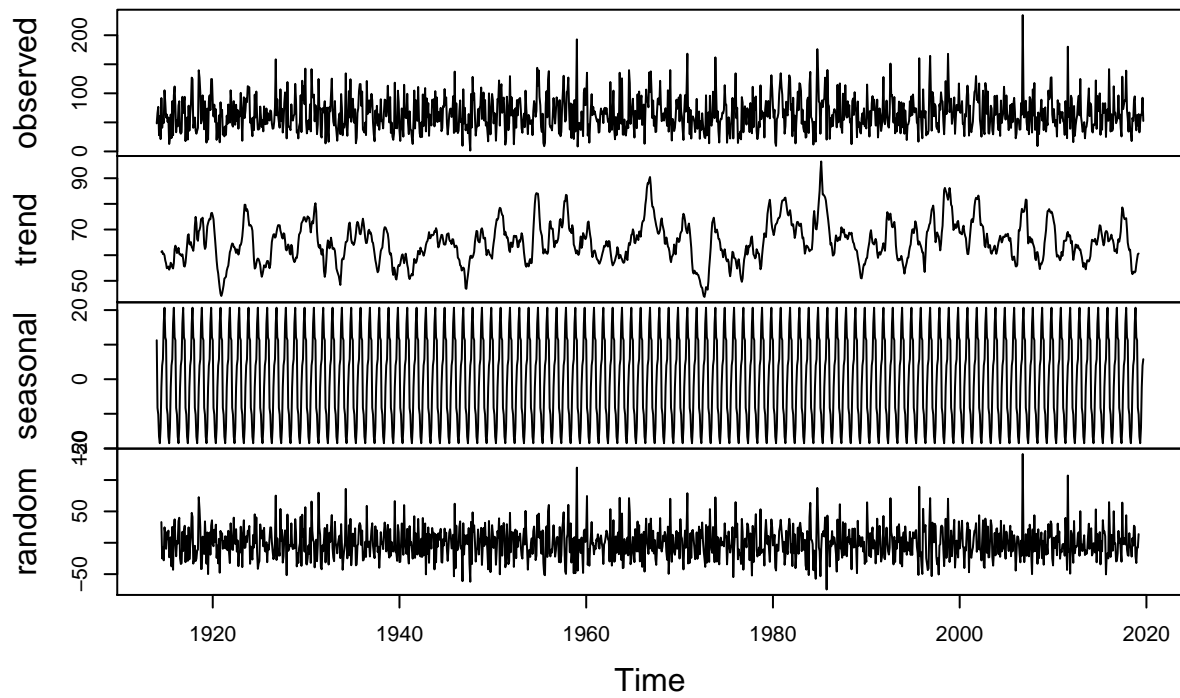
Three things jump out at me from the time series plot and the ACF and PACF plots:

1. The original series is not stationary.
2. There is obvious seasonality in the data.
3. There does not appear to be any trend in the data.

Let's confirm these three observations using the `decompose` function:

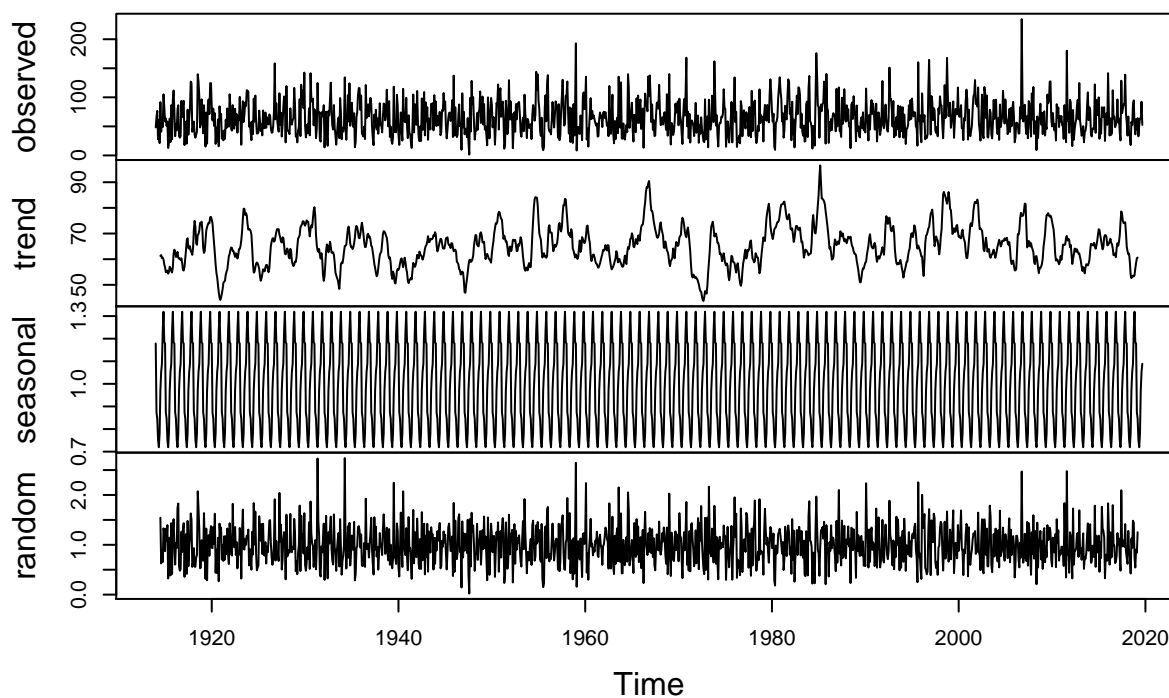
```
plot(decompose(z1, type = "additive", filter = NULL))
```

Decomposition of additive time series



```
plot(decompose(z1, type = "multiplicative", filter = NULL))
```

Decomposition of multiplicative time series

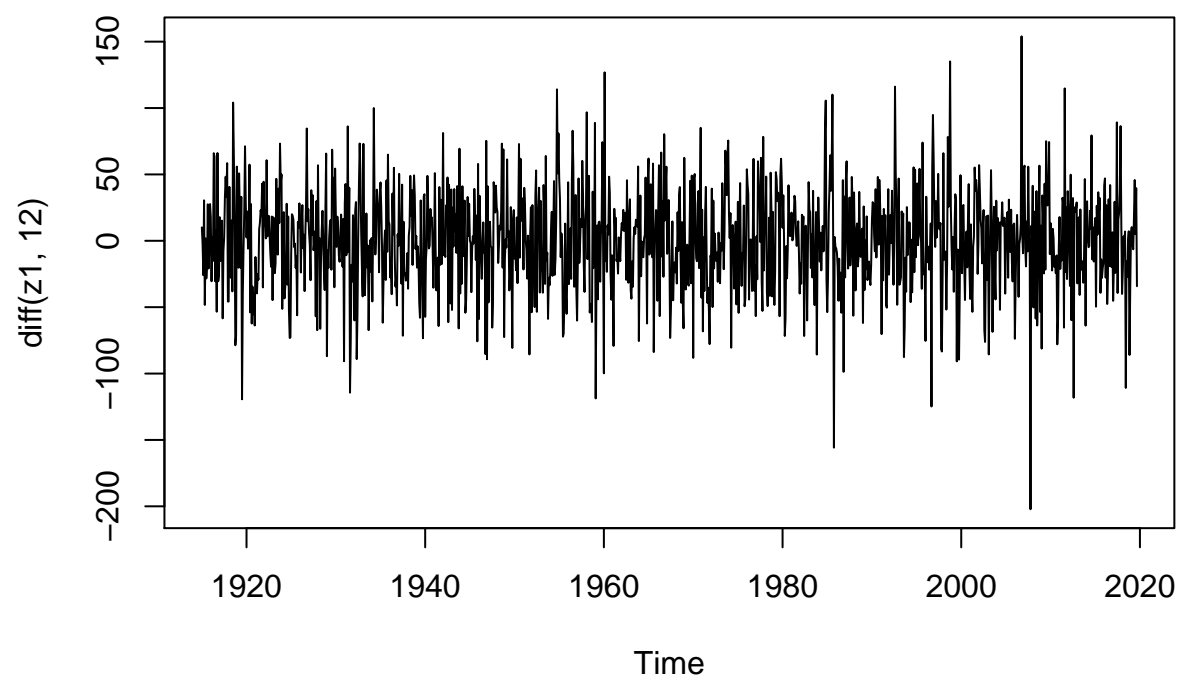


These two plots confirm that there is no trend in the data but there is yearly seasonality. It makes sense that there is no trend in rainfall data but that there is a seasonal aspect to rainfall.

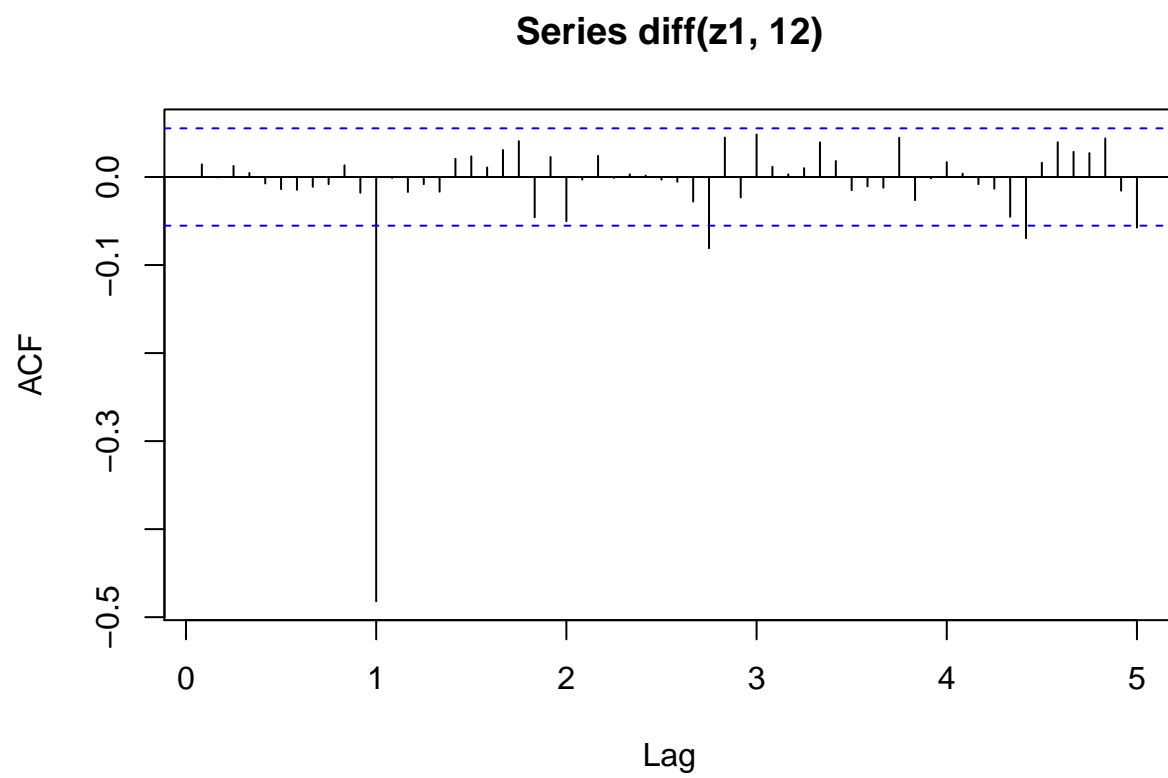
My first objective is to transform the data so they are stationary.

Because the data are monthly, and peaks and troughs in the data repeat every 12 lags, there is no need to take an ordinary difference, but taking a seasonal difference is necessary. Let's try taking a seasonal difference ($s = 12$).

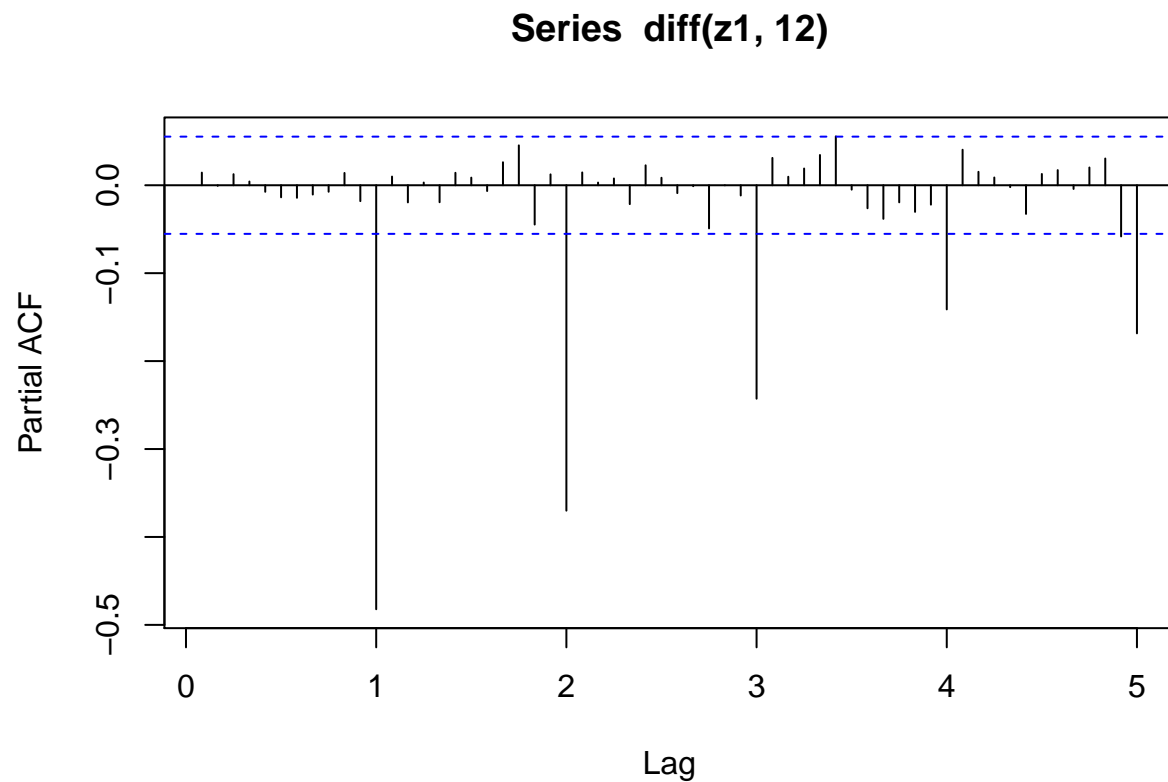
```
plot(diff(z1, 12))
```



```
acf(diff(z1, 12), 60)
```

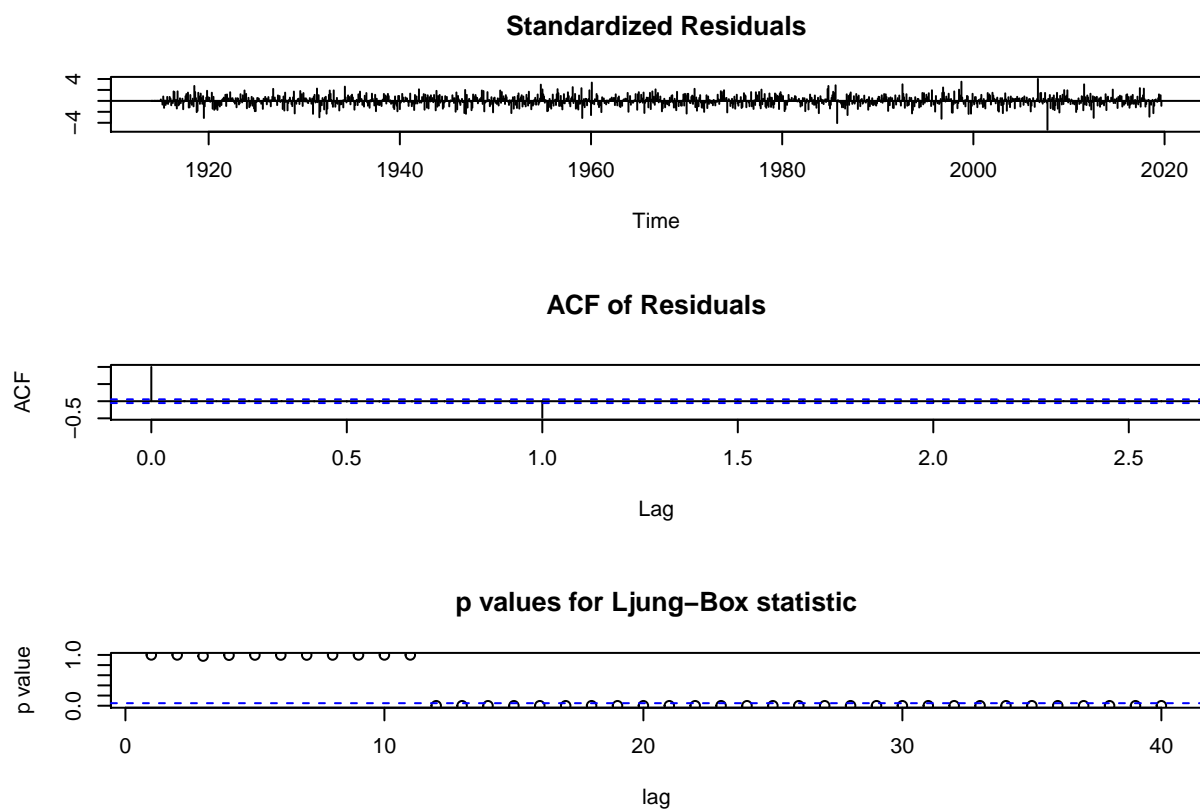



```
pacf(diff(z1, 12), 60)
```

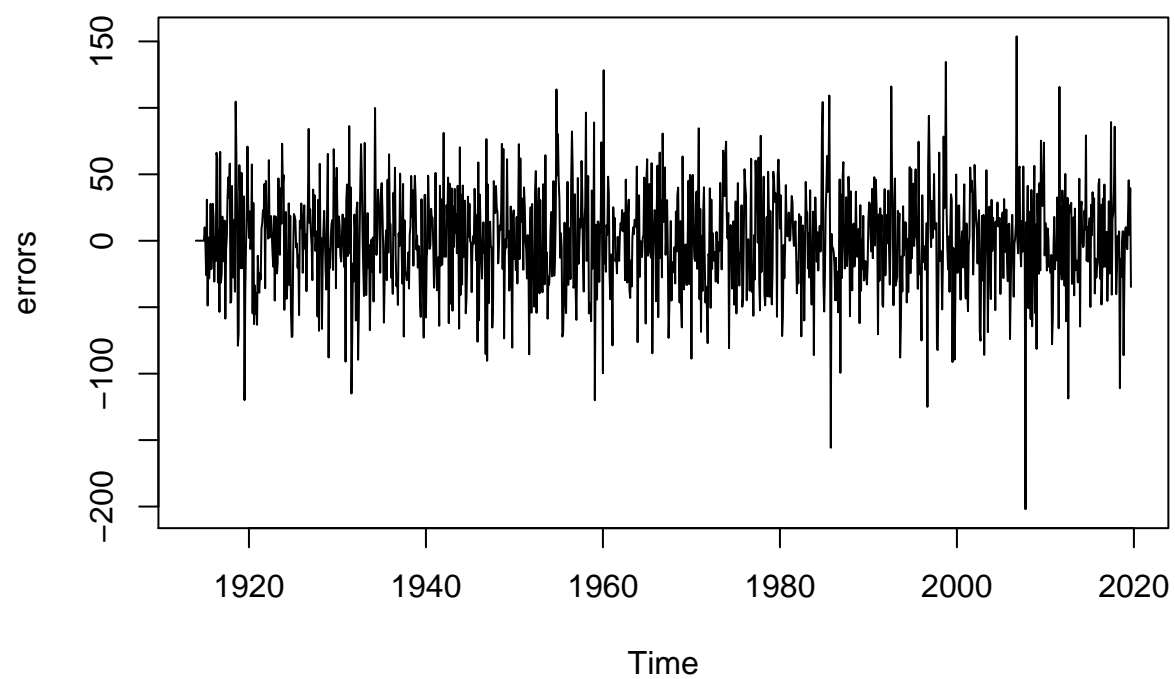


Now the differenced time series looks stationary. Furthermore, all but one lag in the ACF plot is within the 95% confidence interval. That tells me that MA(1) for low order terms looks to be a good candidate model. Let's start with an $ARIMA(0, 0, 1)x(0, 1, 0)_{12}$ model:

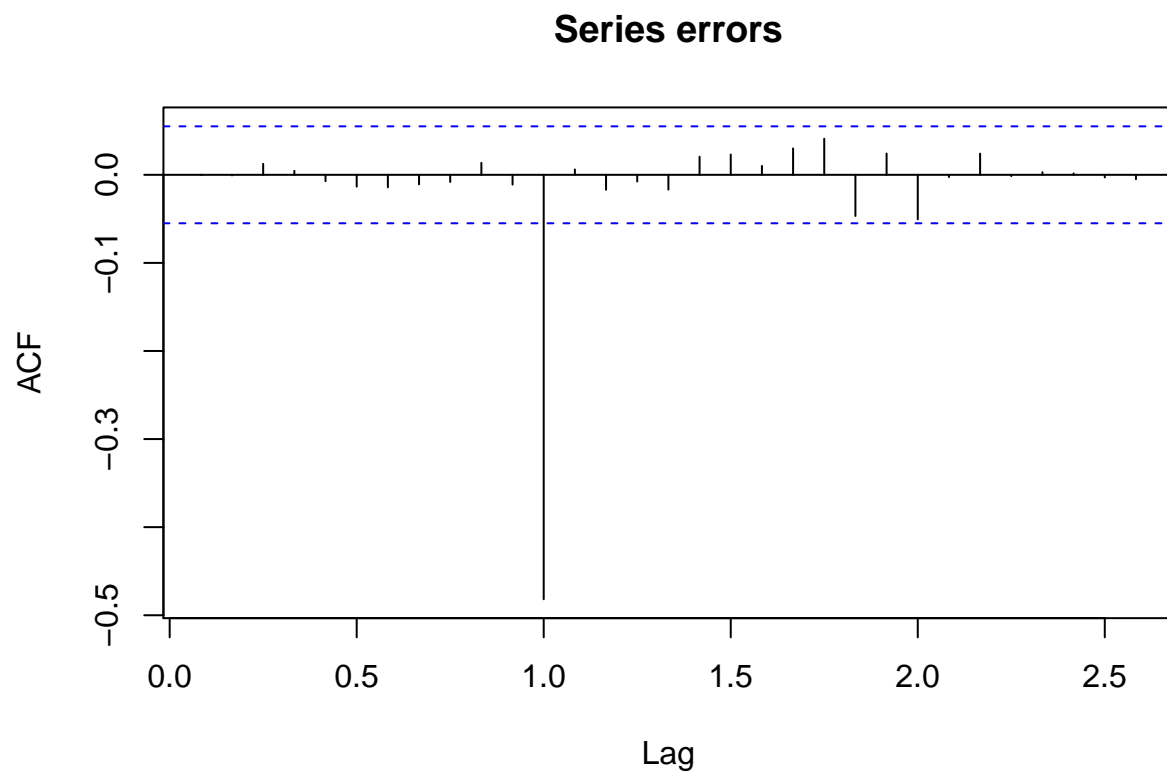
```
z1ARIMA001x010_12model <- arima(z1, order=c(0,0,1), seasonal = list(order=c(0,1,0), period=12))  
tsdiag(z1ARIMA001x010_12model, 40)
```



```
errors <- residuals(z1ARIMA001x010_12model)
plot(errors)
```



```
acf(errors)
```



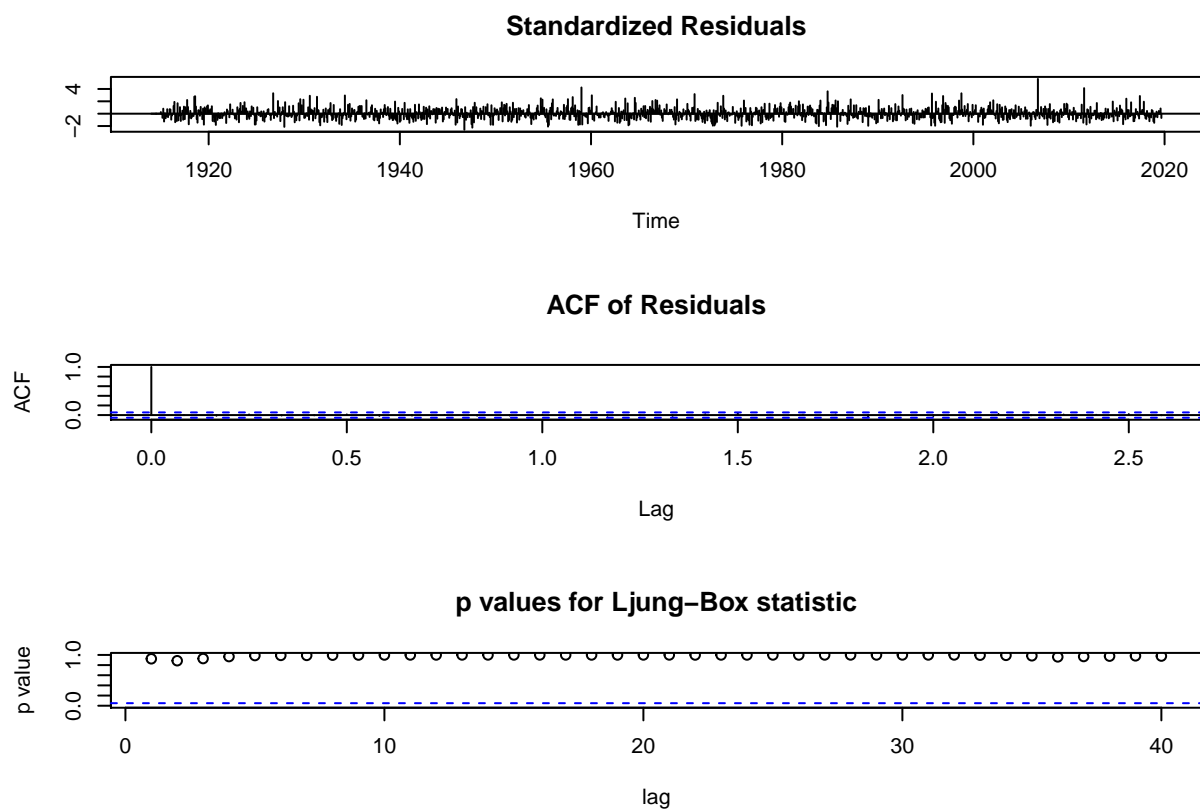
This model is not suitable based on the plot of Ljung-Box test values even though the residuals appear to be a white noise process.

Looking at the autocorrelations plot, lag 12 is outside of the 95% confidence interval.

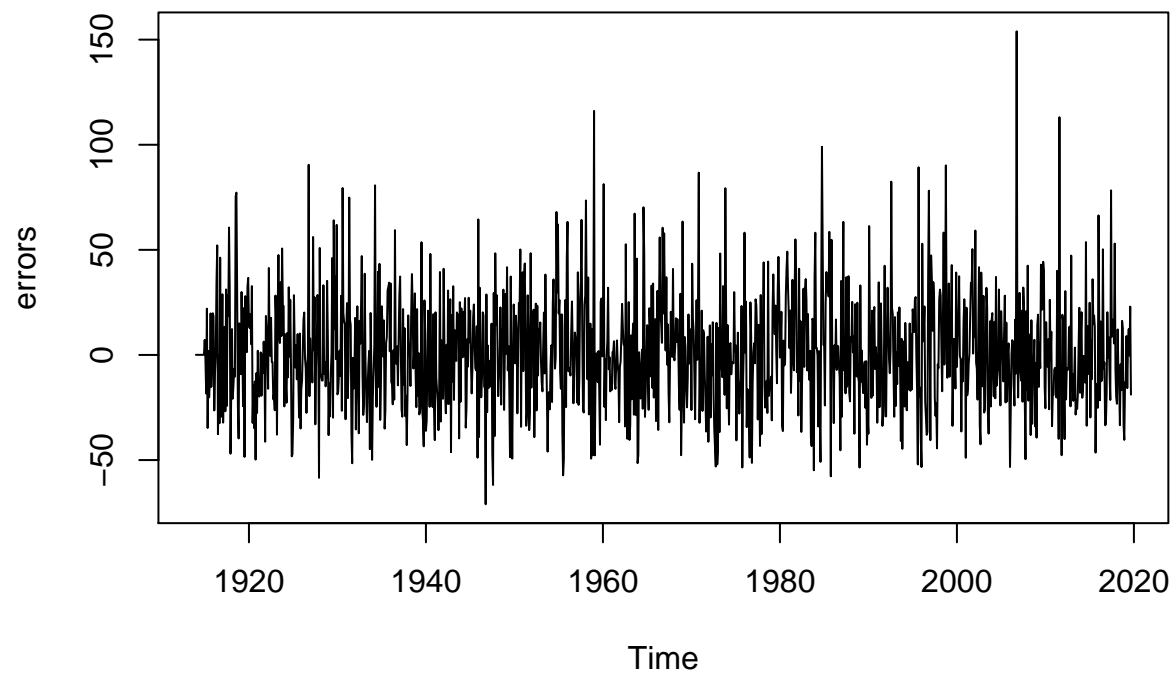
Combining these two pieces of information, it is clear that an $ARIMA(0, 0, 1)x(0, 1, 0)_{12}$ model is not suitable for predictions.

Let's try an $ARIMA(0, 0, 1)x(0, 1, 1)_{12}$ model, since the plot of autocorrelations of residuals above suggests additional information from lag 12 should be included in the model.

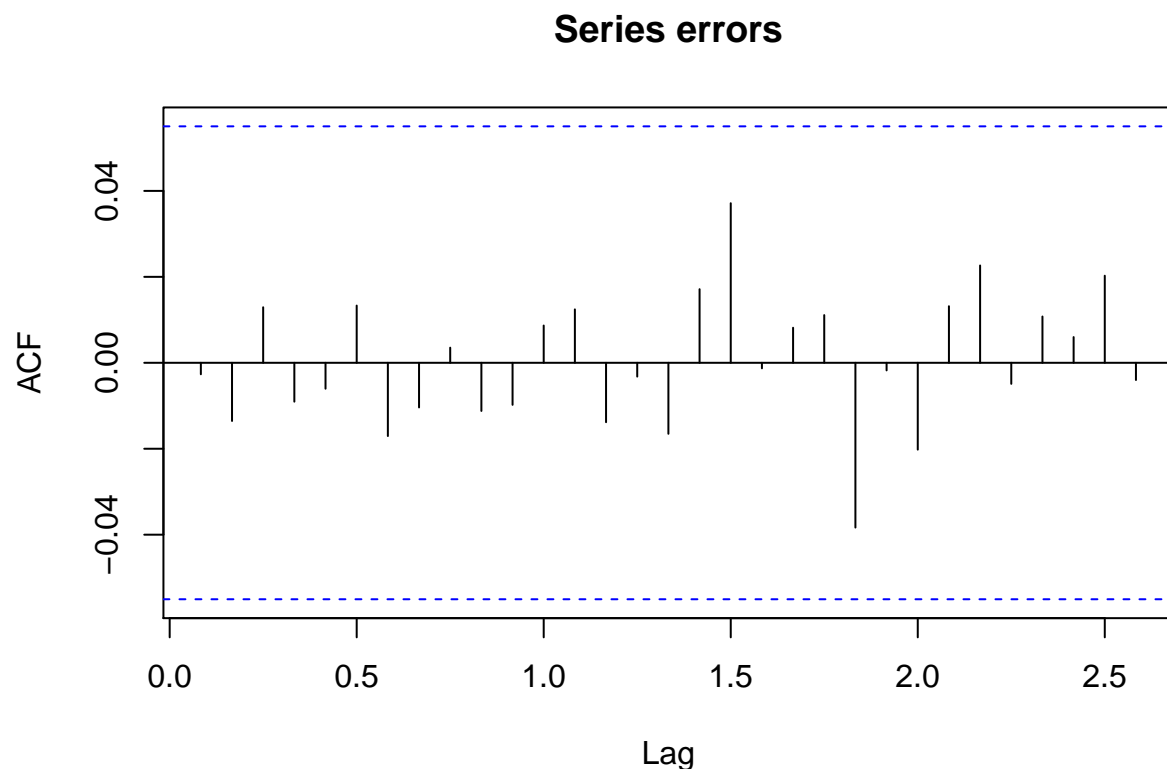
```
z1ARIMA001x011_12model <- arima(z1, order=c(0,0,1), seasonal = list(order=c(0,1,1), period=12))
tsdiag(z1ARIMA001x011_12model, 40)
```



```
errors <- residuals(z1ARIMA001x011_12model)
plot(errors)
```



```
acf(errors)
```



This looks like a strong model based on the plot of Ljung-Box test values. Additionally, the plot of residuals looks like a white noise process. Furthermore, the plot of autocorrelations of residuals shows that they are all within the 95% confidence interval.

Let's run the Ljung-Box test to be extra sure.

```
Box.test(errors, 60, type = "Ljung-Box", fitdf=2)
```

```
##
## Box-Ljung test
##
## data: errors
## X-squared = 43.572, df = 58, p-value = 0.9203
```

Since the p-value = 0.9203 is well above 0.05 for group size 60, we can retain the null hypothesis that this model can be used for predictions.

Let's examine the model parameters in greater detail now that we know it is a good model to use for making predictions.

```
z1ARIMA001x011_12model
```

```
##
## Call:
## arima(x = z1, order = c(0, 0, 1), seasonal = list(order = c(0, 1, 1), period = 12))
##
```



```
## Coefficients:
##          ma1      sma1
##          0.0197 -0.9911
## s.e.    0.0286   0.0225
##
## sigma^2 estimated as 750.1:  log likelihood = -5967.61,  aic = 11939.22
```

The AIC value for this model is 11,941.22.

The equation of an $ARIMA(0,0,1)x(0,1,1)_{12}$ model is the following:

$$(1 - B^{12})Z_t = (1 + \theta_1 B)(1 + \Theta_1 B^{12})A_t$$

This is equivalent to

$$Z_t = Z_{t-12} + A_t + \theta_1 A_{t-1} + \Theta_1 A_{t-12} + \theta_1 \Theta_1 A_{t-13}$$

$\theta_1 = 0.0197$ and $\Theta_1 = -0.9911$.

Thus the equation of this model is:

$$Z_t = Z_{t-12} + A_t + 0.0197A_{t-1} - 0.9911A_{t-12} - 0.0195A_{t-13}$$

Because the Ljung-Box value for group size 60 for this model is very close to 1, residuals resemble a white noise process and autocorrelations of residuals are all within the 95% confidence interval, I consider this model to be quite strong and thus, do not need to continue searching for better models.

Now, let's predict the next four values.

```
predict(z1ARIMA001x011_12model, 4)
```

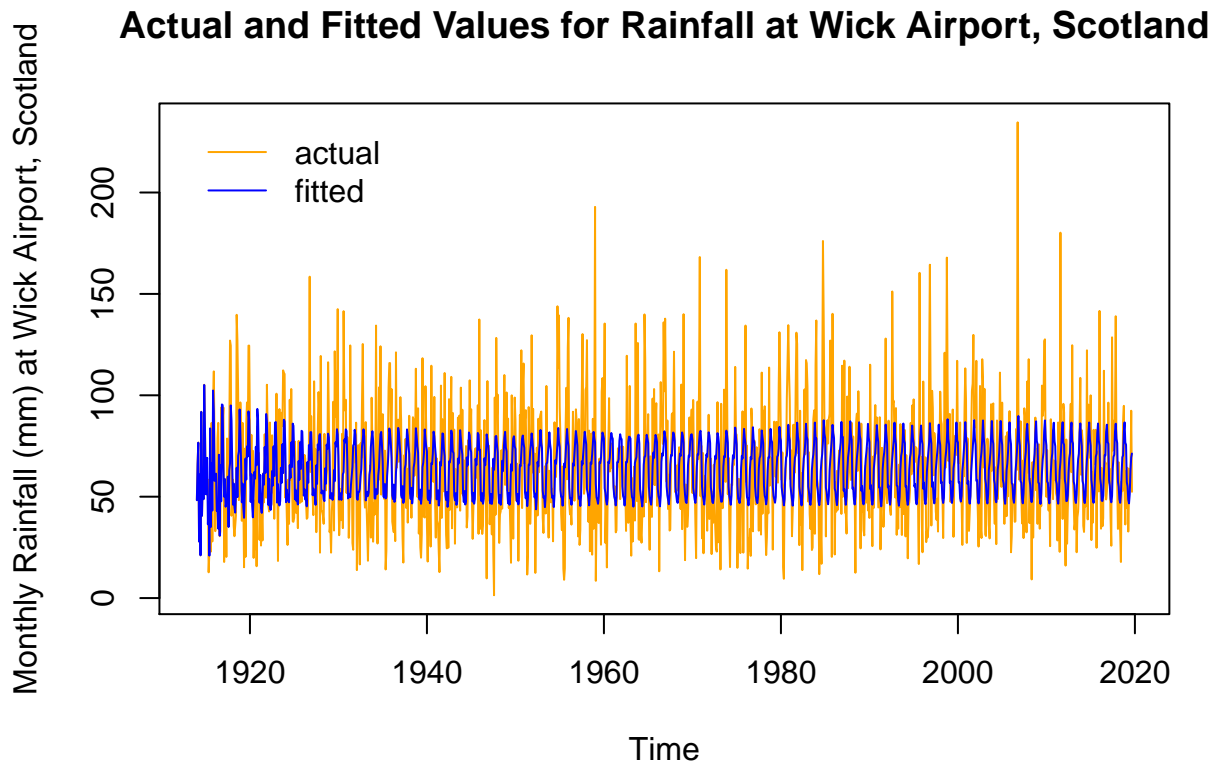
```
## $pred
##           Jan Feb Mar Apr May Jun Jul Aug Sep           Oct           Nov           Dec
## 2019                                81.63528 86.06017 76.95285
## 2020 76.22586
##
## $se
##           Jan Feb Mar Apr May Jun Jul Aug Sep           Oct           Nov           Dec
## 2019                                27.43132 27.43665 27.43665
## 2020 27.43574
```

The following is a graph of the actual and fitted values for all observations in the time series:

```
plot(z1, col="orange", main="Actual and Fitted Values for Rainfall at Wick Airport, Scotland",
     ylab="Monthly Rainfall (mm) at Wick Airport, Scotland")

lines(fitted(z1ARIMA001x011_12model), col="blue")

legend("topleft", inset=.02, legend=c("actual","fitted"), col=c("orange","blue"),lty=1, box.lty=0)
```



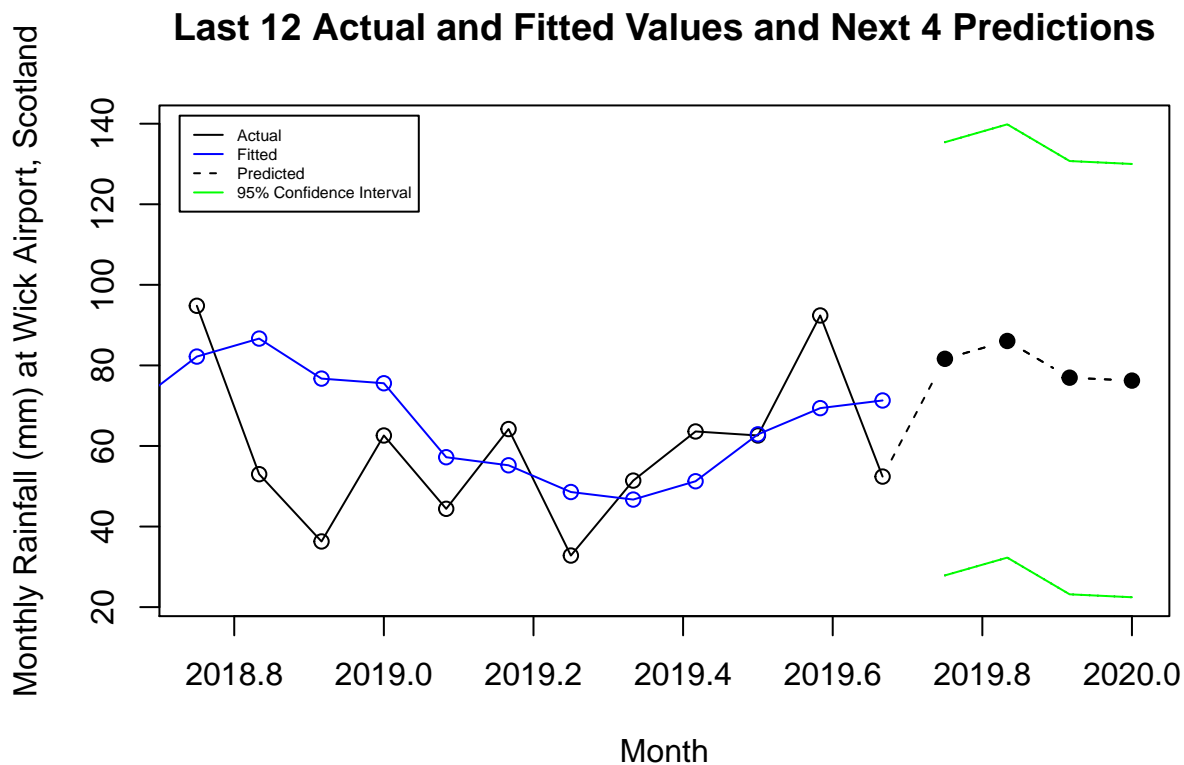
The 12 most recent actual and fitted values, and predicted values with 95% prediction intervals:

```
# last 12 actual values and next four predictions with 95% confidence intervals
pred1 <- plot(z1ARIMA001x011_12model, n.ahead=4,
             main="Last 12 Actual and Fitted Values and Next 4 Predictions",
             ylab='Monthly Rainfall (mm) at Wick Airport, Scotland',
             xlab='Month', n1=c(2018, 10), pch=19)

# add fitted values
points(fitted(z1ARIMA001x011_12model), col="blue")
lines(fitted(z1ARIMA001x011_12model), col="blue")

lines(pred1$lpi, col="green")
lines(pred1$upi, col="green")

legend("topleft", inset=0.02,
      legend=c("Actual", "Fitted", "Predicted", "95% Confidence Interval"),
      col=c("black", "blue", "black", "green"), cex=.5, lty=c(1,1,2,1))
```



Now let's find the best model using exponential smoothing and the Holt-Winters method.

Holt-Winters smoothing:

```
### exponential smoothing:
expSmoothing <- HoltWinters(z1, beta=F, gamma=F)

### Holt's method (no seasonality):
### HoltWinters(x, gamma=F)

### Holt-Winter's multiplicative:
HWmult <- HoltWinters(z1, seasonal="multiplicative")

### Holt-Winter's additive:
HWadd <- HoltWinters(z1, seasonal="additive")
```

Holt's method assumes no seasonality, which is not appropriate here, so this method will be omitted.

The original time series z_1 does not pass through zero, so Holt-Winters multiplicative is appropriate in this situation.

```
exp_measures <- measures(expSmoothing)
HW_add_measures <- measures(HWadd)
HW_mult_measures <- measures(HWmult)

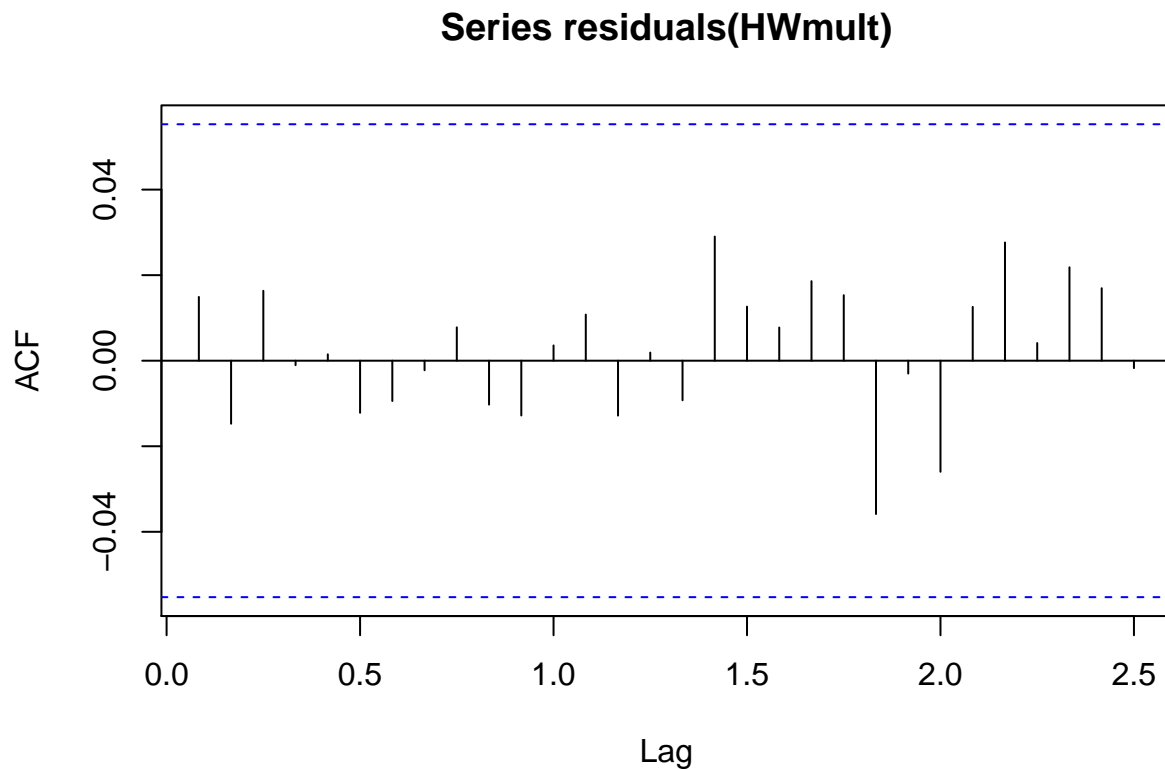
data.frame(rbind(exp_measures, HW_add_measures, HW_mult_measures))
```

```
##                MAD      RMSE      MAPE
## exp_measures    23.96028 30.27062 0.5535743
## HW_add_measures 22.55753 28.74327 0.5074540
## HW_mult_measures 22.38589 28.53296 0.5052846
```

The Holt-Winters multiplicative model is the best of the three using MAD, RMSE and MAPE measures.

Let's examine the autocorrelations of residuals of the Holt-Winters multiplicative model.

```
acf(residuals(HWmult))
```



All residuals are within the 95% confidence interval.

The 12 most recent actual and fitted values and predicted values for Holt-Winters multiplicative model and $ARIMA(0,0,1)x(0,1,0)_{12}$ model:

```
# actual values
last_12_actual <- series1$Rain[c((nrow(series1)-11):nrow(series1))]
last_12_actual <- ts(last_12_actual, start=c(2018,10), frequency = 12)

# Holt-Winters fitted
HWmult_resid <- residuals(HWmult)
HWmult_resid <- HWmult_resid[1246:1257]
HWresiduals <- ts(HWmult_resid, start=c(2018,10), frequency = 12)
last_12_fitted_HW <- last_12_actual - HWresiduals

# Holt-Winters and ARIMA predictions
```

```

predictions_holt <- predict(HWmult, 4)
predictionsSARIMA <- predict(z1ARIMA001x011_12model, 4)

# SARIMA fitted
SARIMA_resid <- residuals(z1ARIMA001x011_12model)
SARIMA_resid <- SARIMA_resid[1246:1257]
last_12_fitted_SARIMA <- last_12_actual - SARIMA_resid

# Actual values
plot(last_12_actual, ylim = c(0, 110), xlim=c(2018.75,2020), col="blue",
      ylab="Monthly Rainfall (mm) at Wick Airport",
      main="Actual, Fitted and Predicted Values, HW Mult and SARIMA Models")
points(last_12_actual, col="blue")

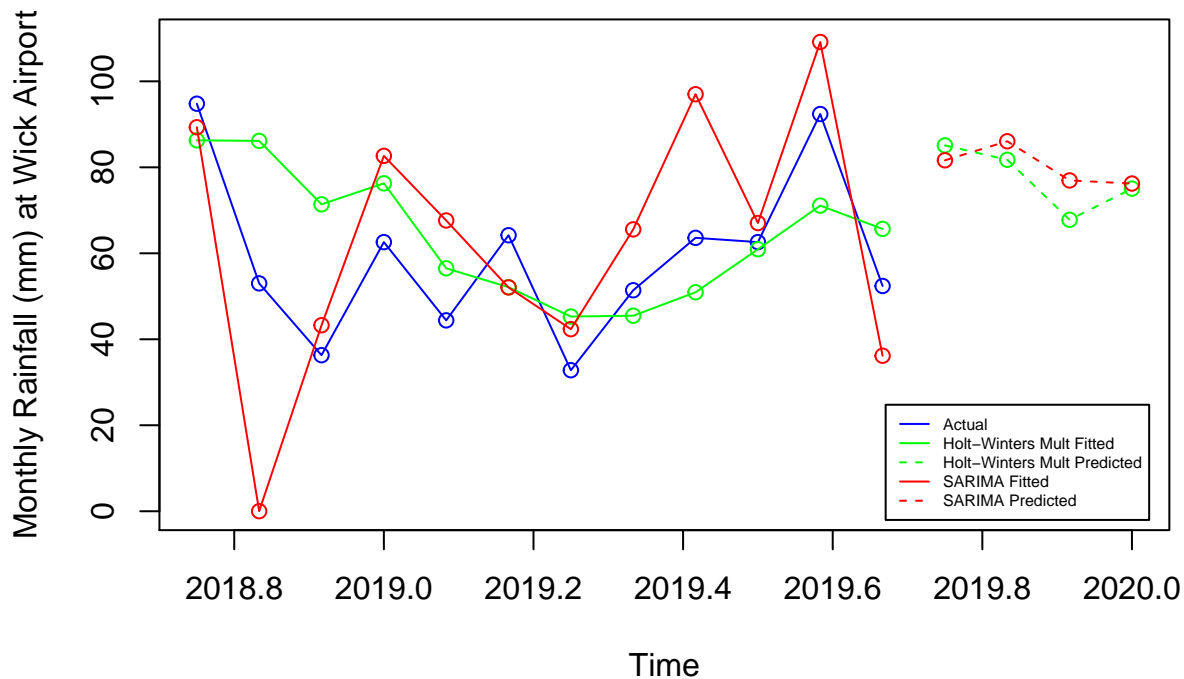
# Holt-Winters multiplicative model fitted and predicted values
lines(last_12_fitted_HW, col="green")
points(last_12_fitted_HW, col="green")
lines(predictions_holt, col="green", lty=2)
points(predictions_holt, col="green", lty=2)

# SARIMA model fitted and predicted values
lines(last_12_fitted_SARIMA, col="red")
points(last_12_fitted_SARIMA, col="red")
lines(predictionsSARIMA$pred, col="red", lty=2)
points(predictionsSARIMA$pred, col="red", lty=2)

legend("bottomright", inset=.02,
      legend=c("Actual", "Holt-Winters Mult Fitted",
               "Holt-Winters Mult Predicted", "SARIMA Fitted", "SARIMA Predicted"),
      col=c("blue","green","green","red", "red"), cex=.5, lty=c(1,1,2,1,2))

```

Actual, Fitted and Predicted Values, HW Mult and SARIMA Models



Data Series 2: GF027: State Budget Tax Revenues by Year and Month

My second data series represents state budget tax revenues in Estonia by month and year in thousands of euros.

Inspecting the data and preparing the data for modeling

```
series2 <- read.table("GF027s_SOCIAL.csv", skip=2, sep=";", dec=".",
                      header=T, na.strings="..", as.is=T)

setDT(series2)

months <- c("January", "February", "March", "April", "May", "June",
            "July", "August", "September", "October", "November", "December")

series2 <- series2 %>%
  select(-1) %>%
  slice(-1) %>%
  rename(Year = X.1) %>%
  melt(id.vars = "Year", value.name = "Tax_Revenue") %>%
  rename(Month = variable) %>%
  mutate(Month_num = match(Month, months)) %>%
```

```

select(Year, Month_num, Tax_Revenue) %>%
  arrange(Year, Month_num)

series2 <- series2[1:237] # remove first row and last seven rows

str(series2)

## Classes 'data.table' and 'data.frame':  237 obs. of  3 variables:
##  $ Year      : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
##  $ Month_num  : int   1  2  3  4  5  6  7  8  9 10 ...
##  $ Tax_Revenue: num  52498 46486 51801 51249 55072 ...
##  - attr(*, ".internal.selfref")=<externalptr>

head(series2)

##      Year Month_num Tax_Revenue
## 1: 2000           1    52498.37
## 2: 2000           2    46486.27
## 3: 2000           3    51800.84
## 4: 2000           4    51248.71
## 5: 2000           5    55071.84
## 6: 2000           6    57101.03

tail(series2)

##      Year Month_num Tax_Revenue
## 1: 2019           4    264880.4
## 2: 2019           5    278206.0
## 3: 2019           6    277579.8
## 4: 2019           7    292217.7
## 5: 2019           8    284670.6
## 6: 2019           9    276019.7

table(series2$Year) # to ensure that no data are missing

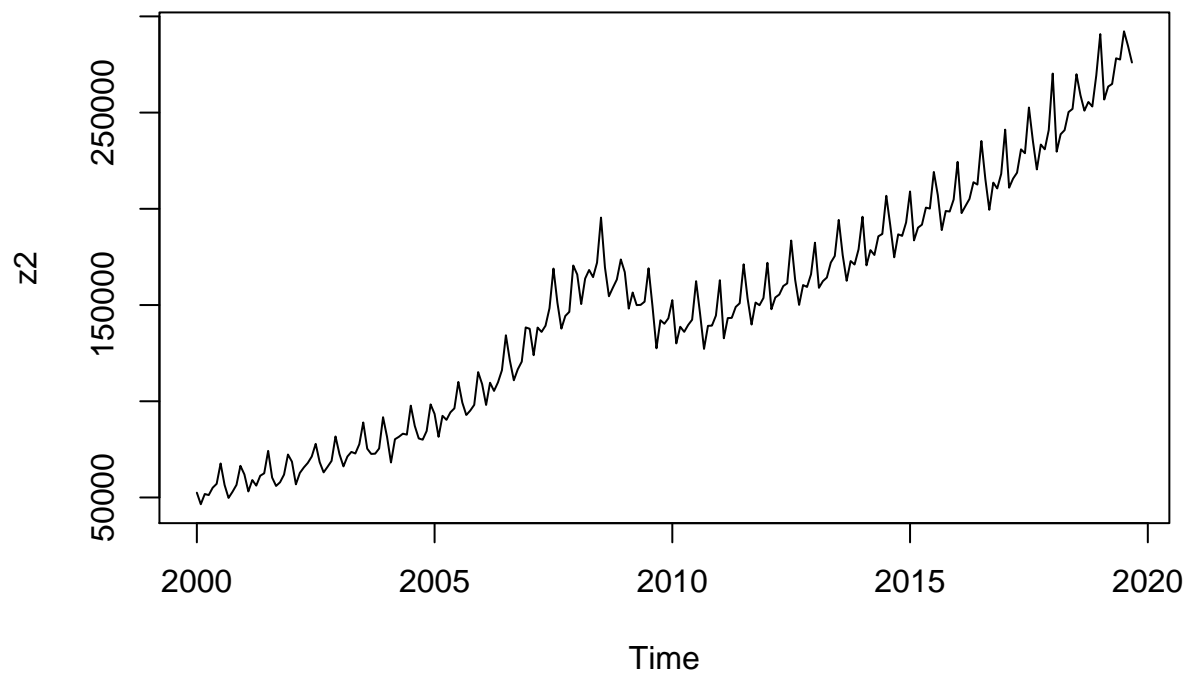
##
## 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
##   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12   12
## 2016 2017 2018 2019
##   12   12   12    9

No data are missing in this series.

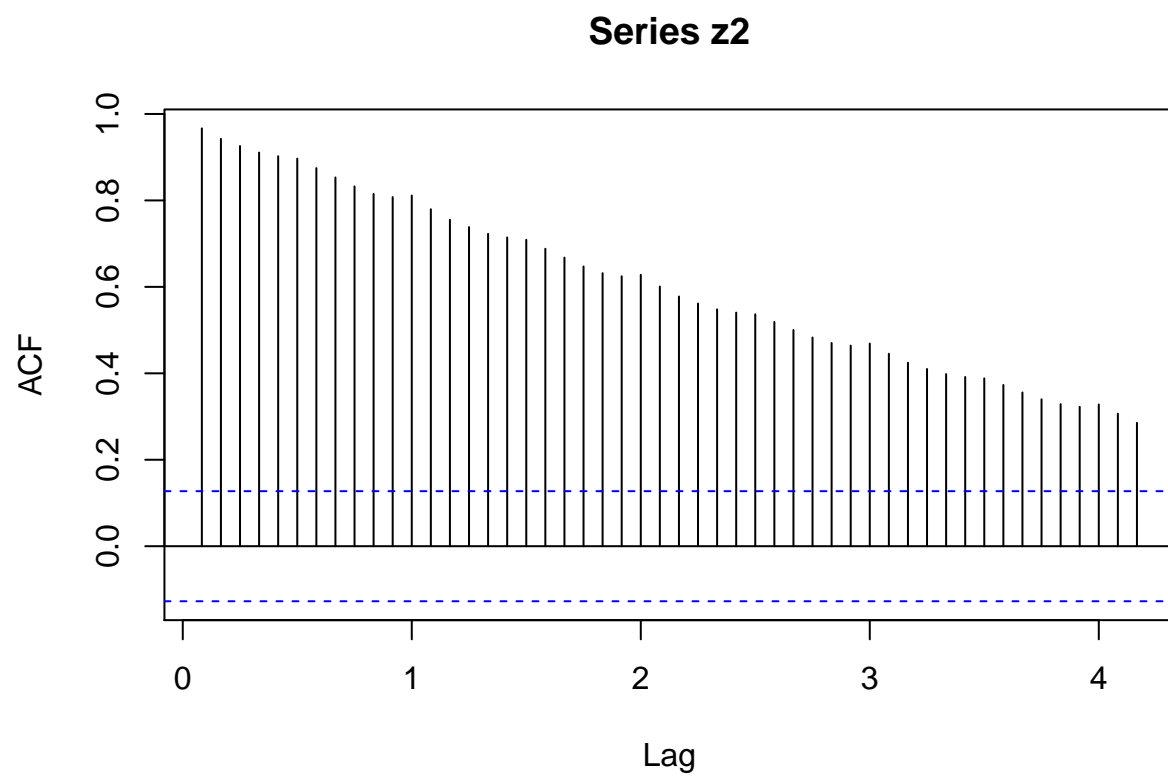
z2 <- ts(series2$Tax_Revenue, frequency=12, start=c(series2$Year[1], series2$Month[1]))

plot(z2)

```

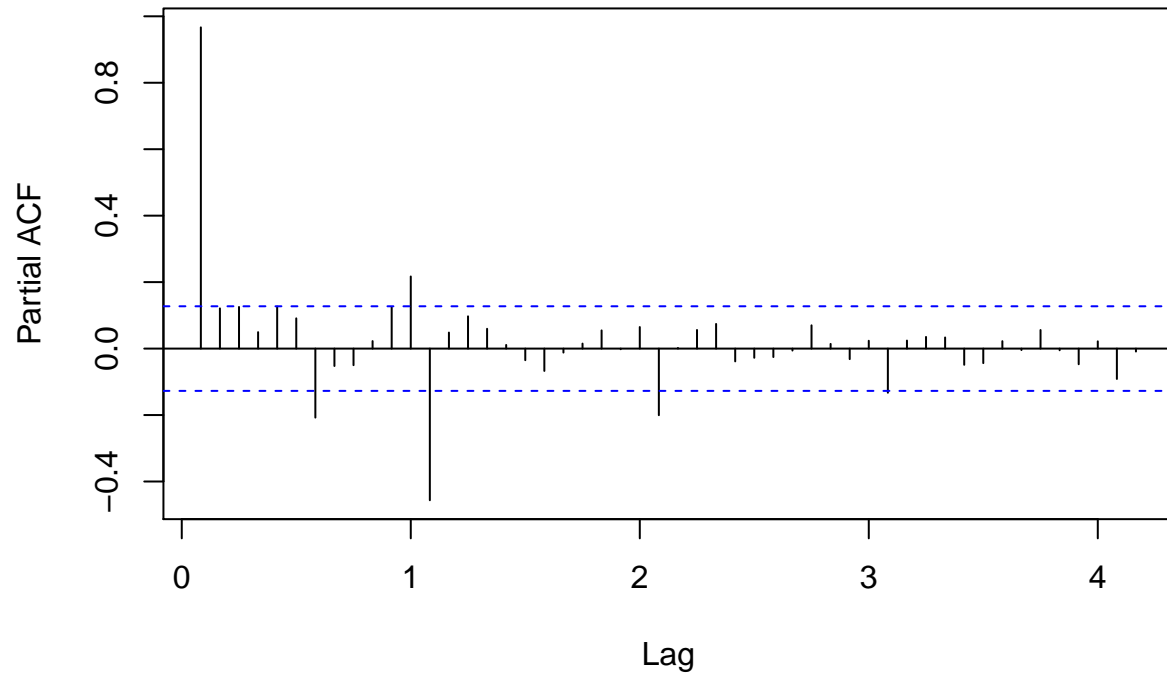


```
acf(z2, 50)
```

```
pacf(z2, 50)
```

Series z2



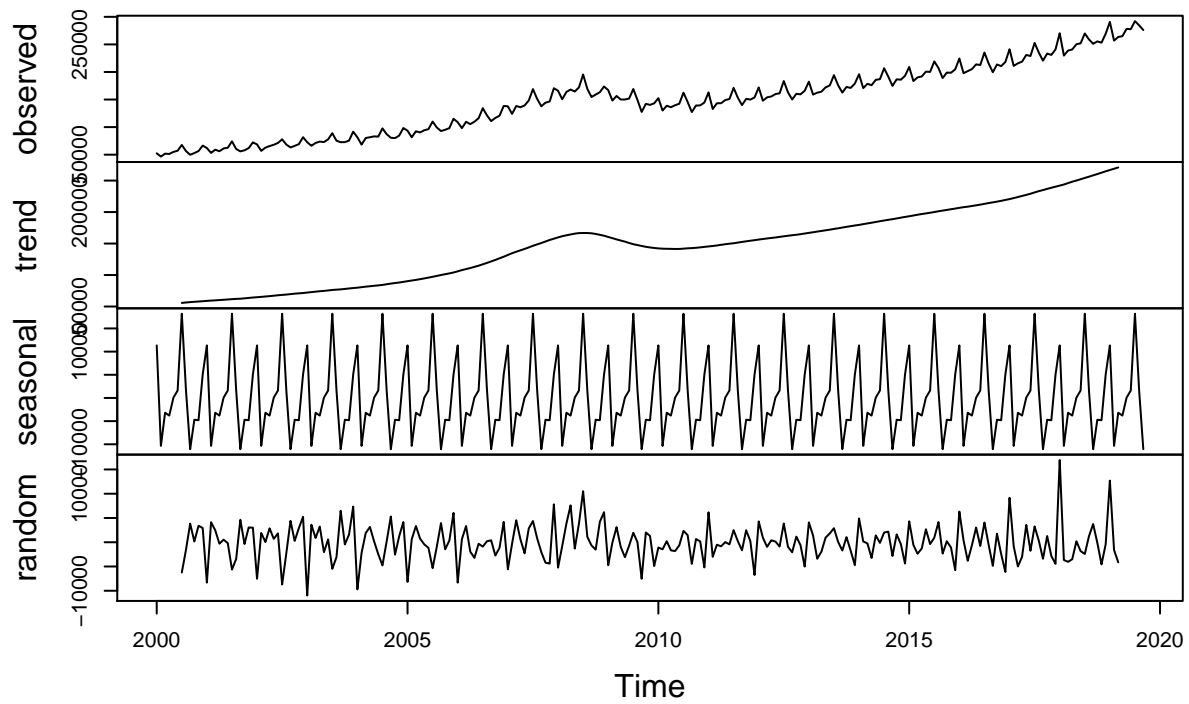
Three things jump out at me from the time series plot and the ACF and PACF plots:

1. The original series is not stationary.
2. There is obvious seasonality in the data.
3. There is obvious trend in the data.

Let's confirm these three observations using the `decompose` function:

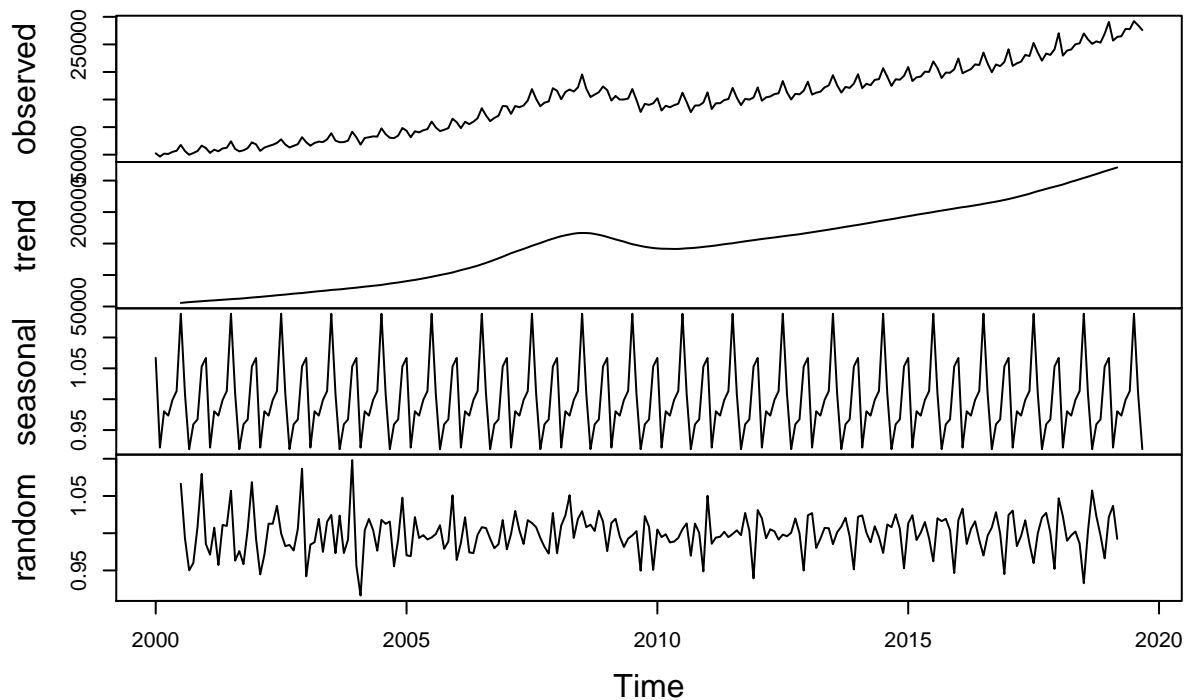
```
plot(decompose(z2, type = "additive", filter = NULL))
```

Decomposition of additive time series



```
plot(decompose(z2, type = "multiplicative", filter = NULL))
```

Decomposition of multiplicative time series

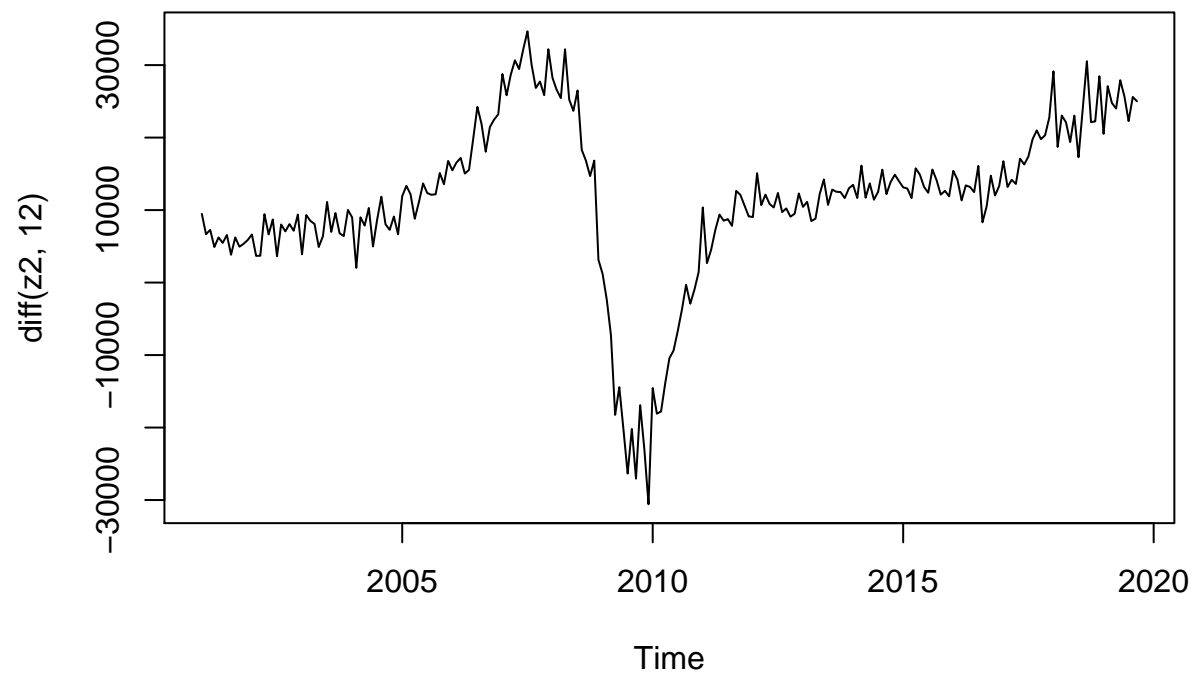


These two plots confirm that there is trend in the data as well as yearly seasonality. It makes sense that there is trend in tax revenue data (due to cost of living increases and inflation) as well as a seasonal aspect, since many types of taxes are collected at discrete points in the calendar year rather than continuously (in the case of sales tax for example).

My first objective is to transform the data so they are stationary.

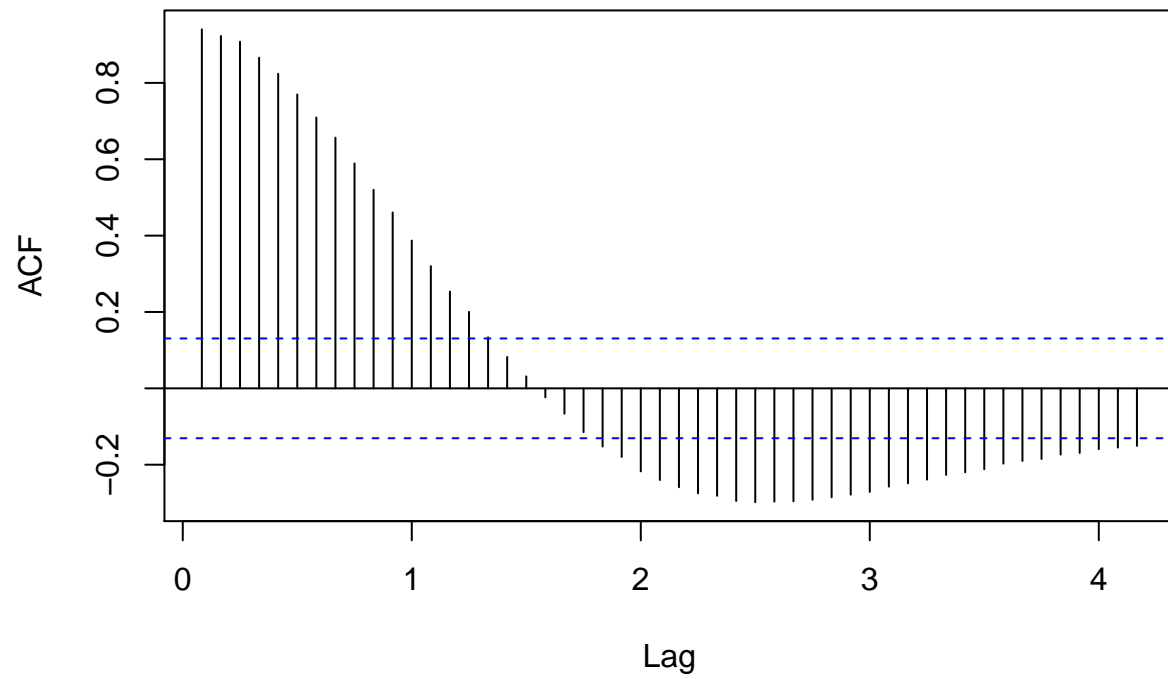
Let's start by taking a seasonal difference ($s = 12$).

```
plot(diff(z2, 12))
```



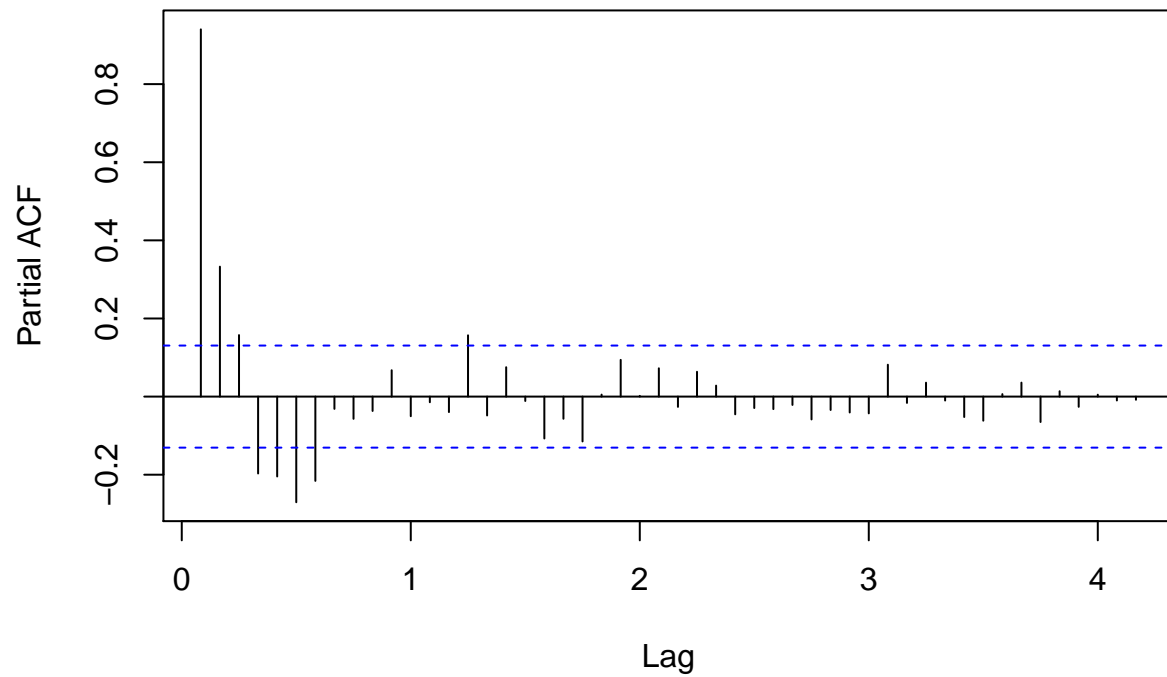
```
acf(diff(z2, 12), 50)
```

Series diff(z2, 12)



```
pacf(diff(z2, 12), 50)
```

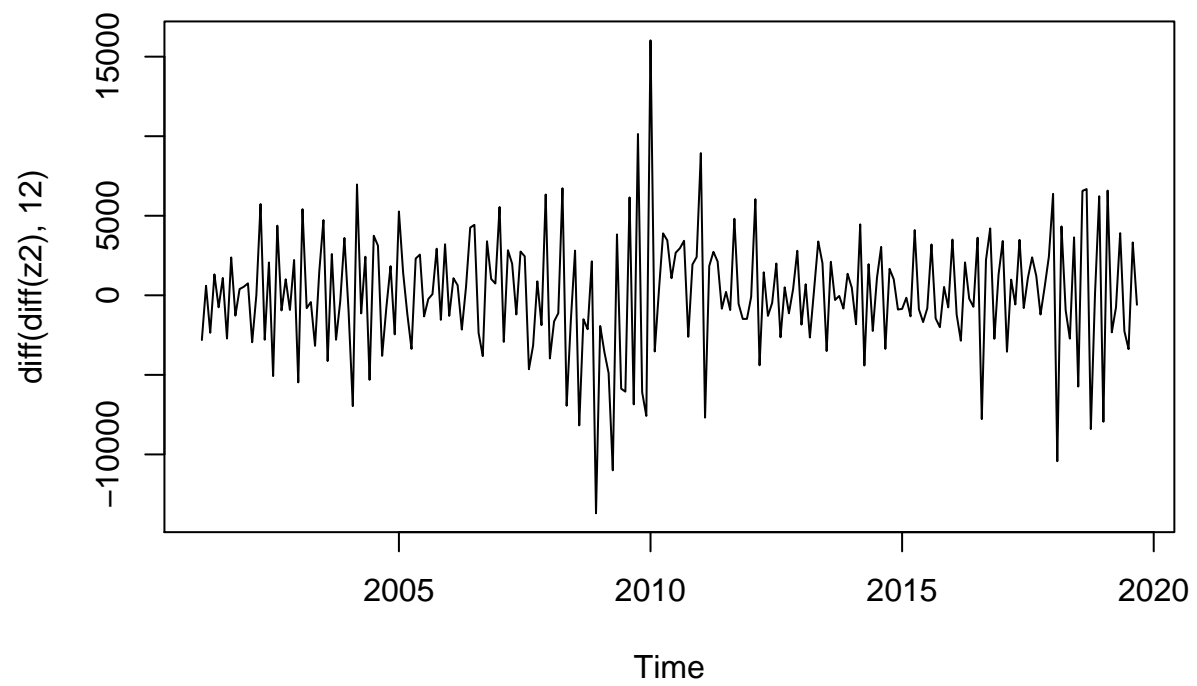
Series `diff(z2, 12)`



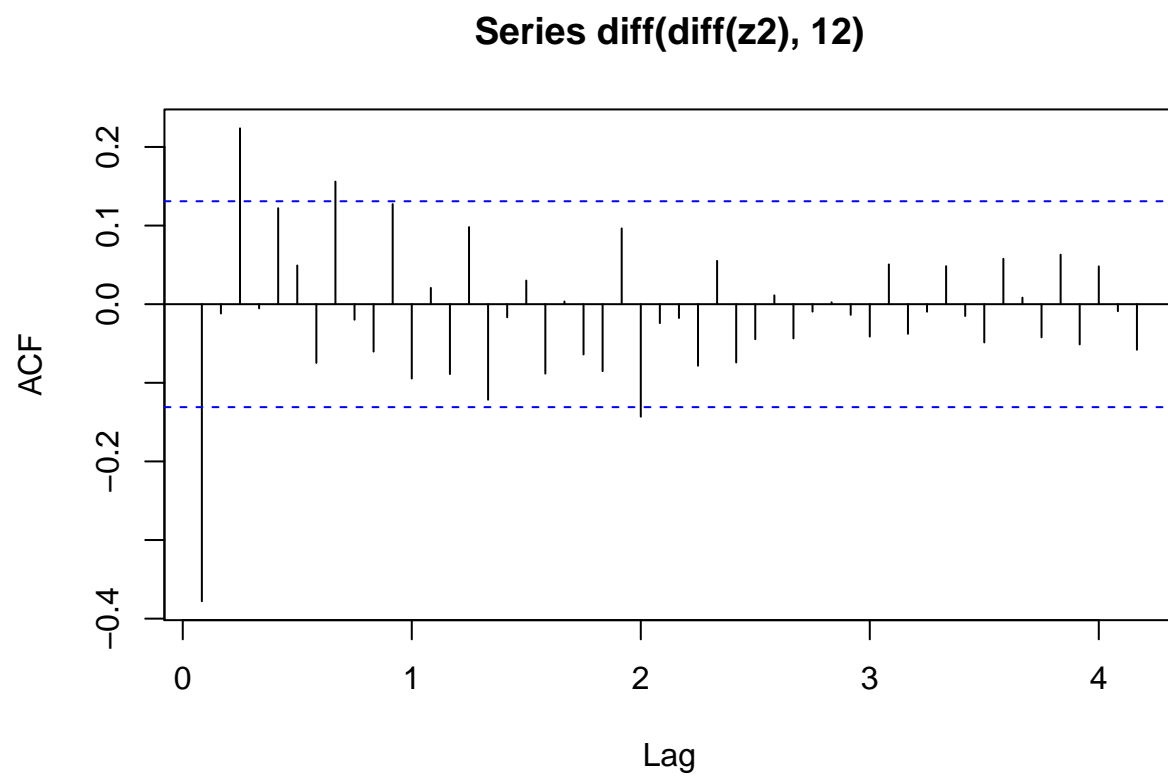
The seasonal differenced series is still not stationary, has no fixed mean and has noticeable trend, aside from the massive dip thanks to the Great Financial Crisis (thank you, Bear Stearns and AIG).

Let's follow the procedure sketched out in pp. 2-5 of the Week 9 lecture notes and take an ordinary difference to detrend the seasonal differenced series.

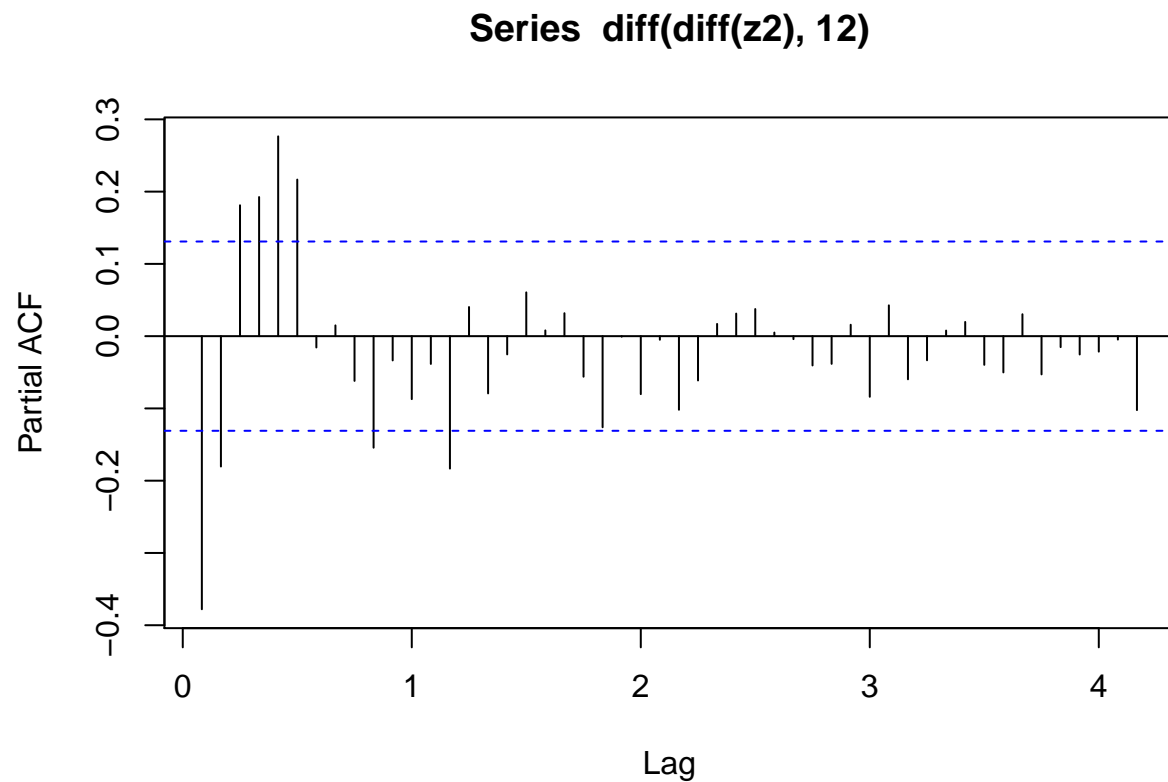
```
plot(diff(diff(z2), 12))
```



```
acf(diff(diff(z2), 12), 50)
```

```
pacf(diff(diff(z2), 12), 50)
```



Now the series looks stationary.

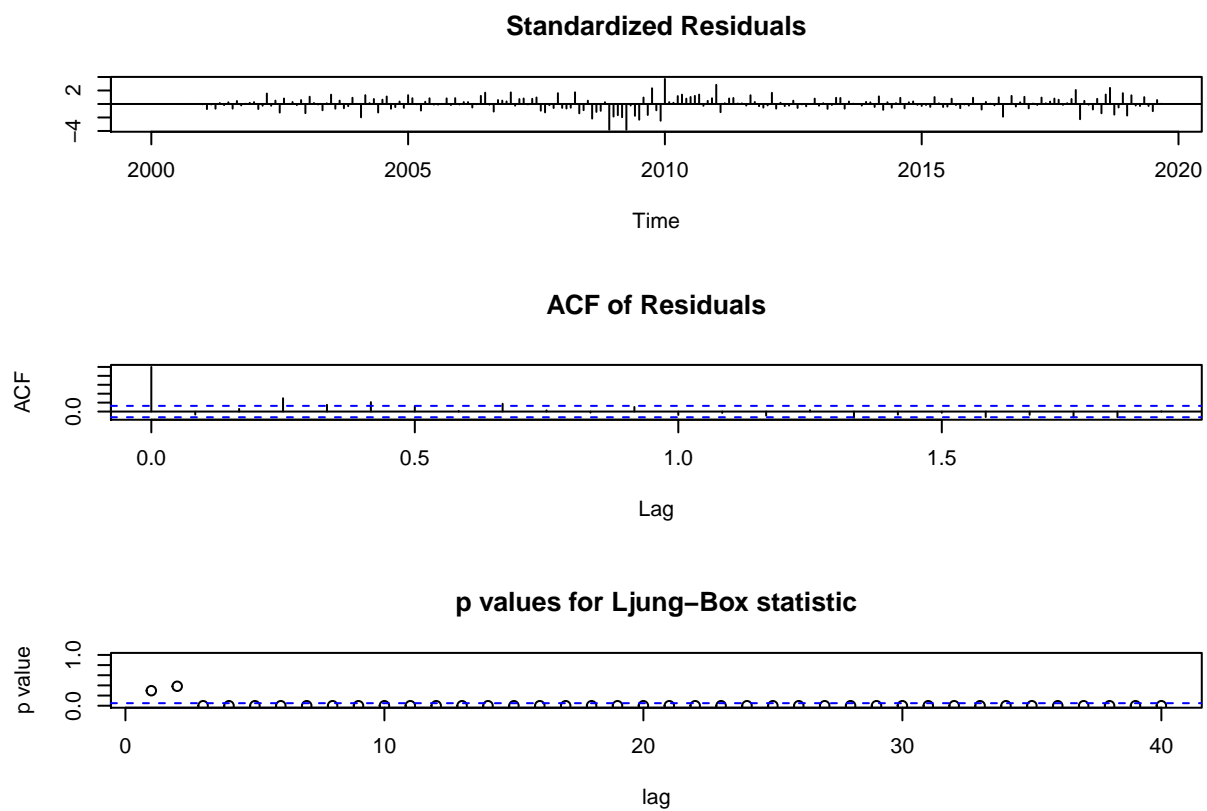
Judging from the ACF plot of the seasonally differenced and ordinary differenced series, $MA(1)$ and $MA(2)$ are candidate models for the low order terms.

Since there are no lags outside of the 95% confidence interval for multiples of 12, it may be the case that an $ARIMA(p, 1, q)x(0, 1, 0)_{12}$ model is suitable.

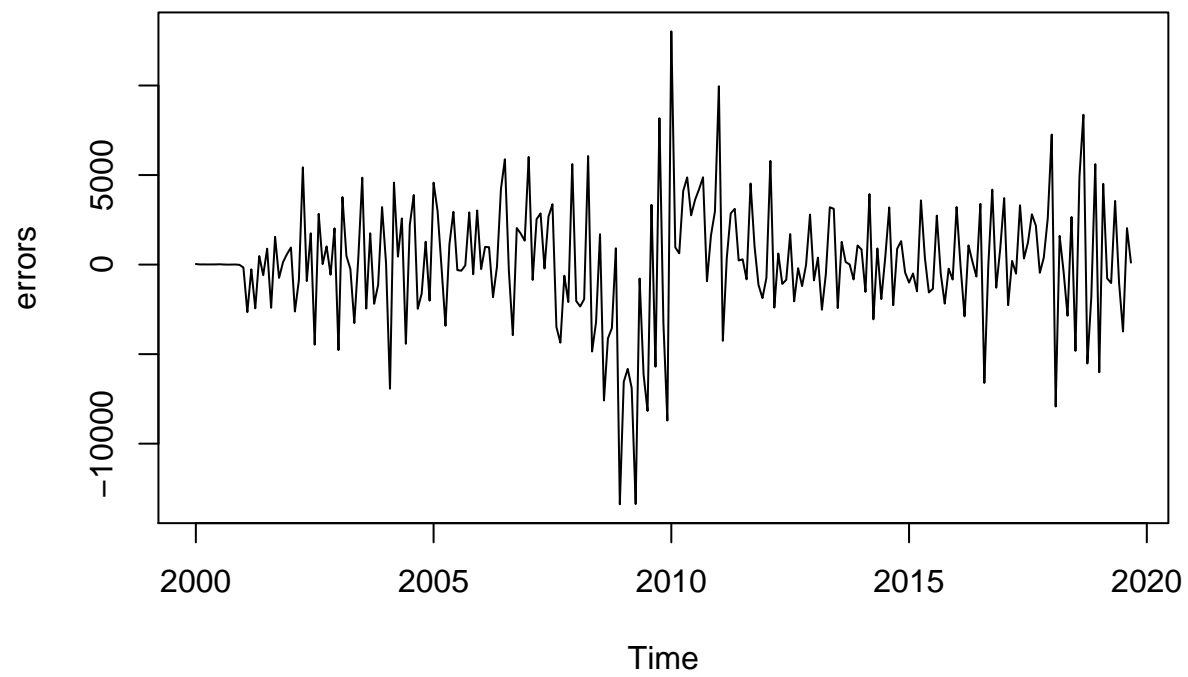
Let's start with an $ARIMA(0, 1, 1)x(0, 1, 0)_{12}$ model.

```
z2ARIMA011x010_12model <- arima(z2, order=c(0,1,1), seasonal = list(order=c(0,1,0), period=12))

tsdiag(z2ARIMA011x010_12model, 40)
```

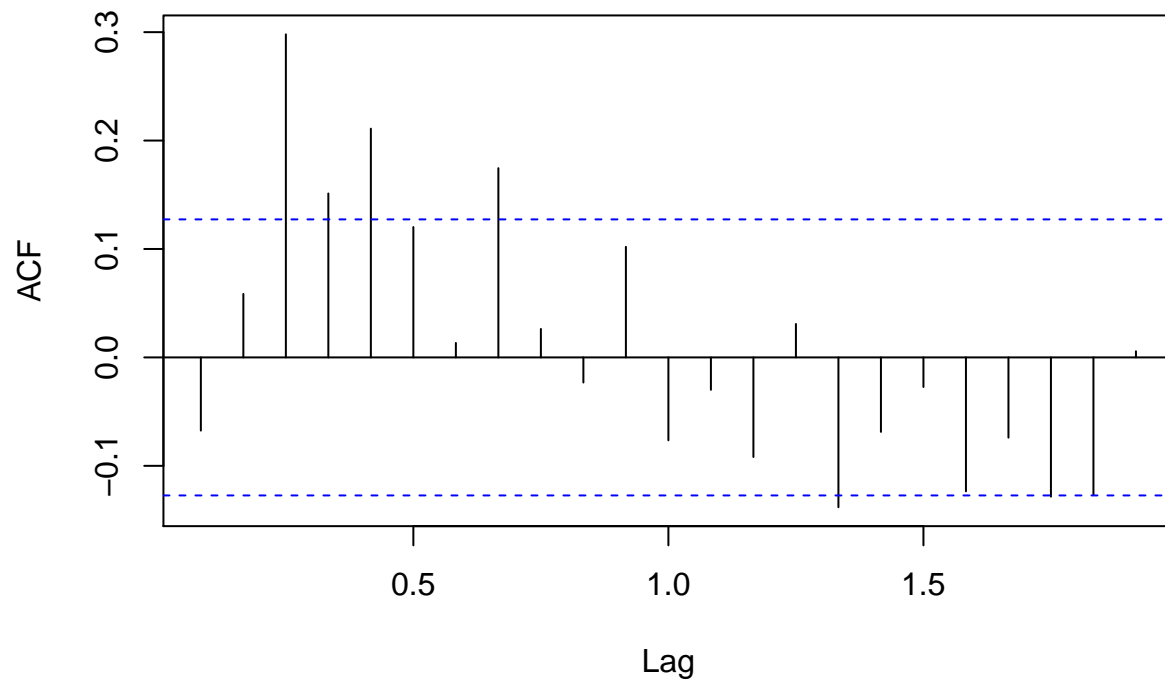


```
errors <- residuals(z2ARIMA011x010_12model)
plot(errors)
```



```
acf(errors)
```

Series errors



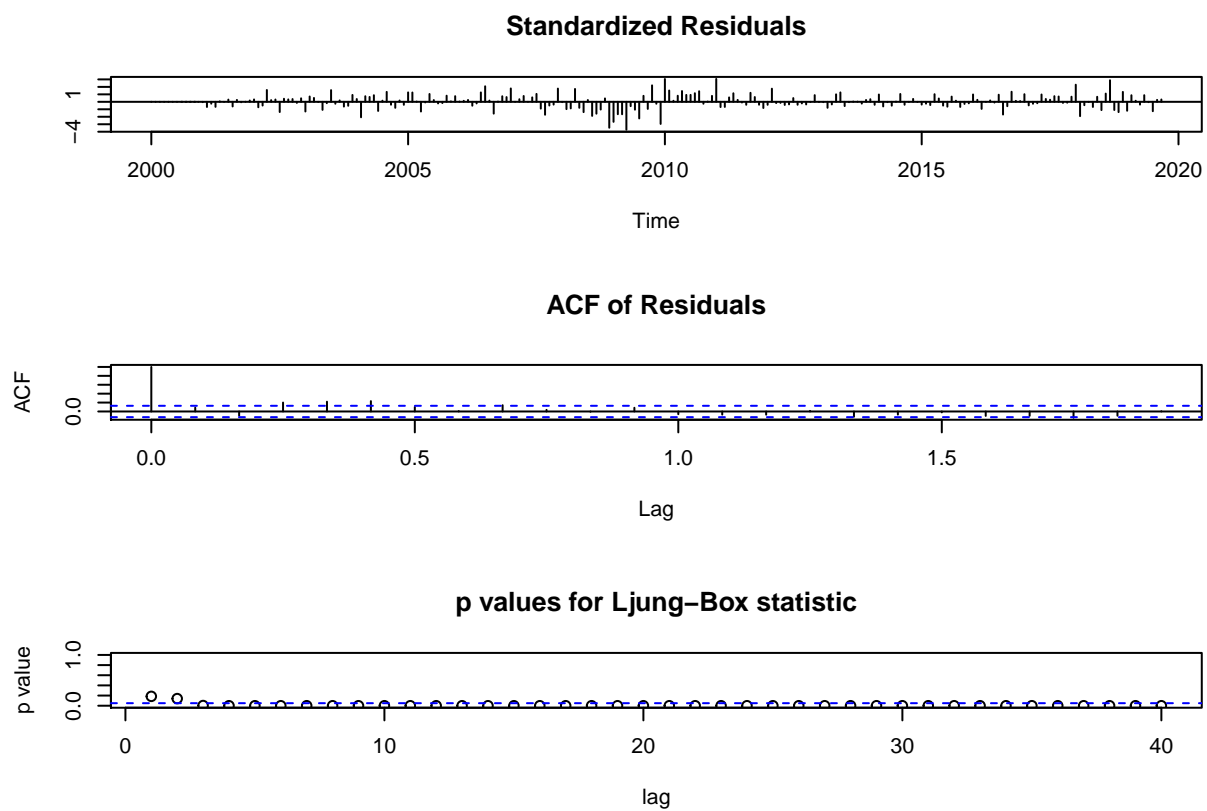
```
Box.test(errors, 60, type = "Ljung-Box", fitdf=1)
```

```
##  
## Box-Ljung test  
##  
## data: errors  
## X-squared = 120.07, df = 59, p-value = 4.643e-06
```

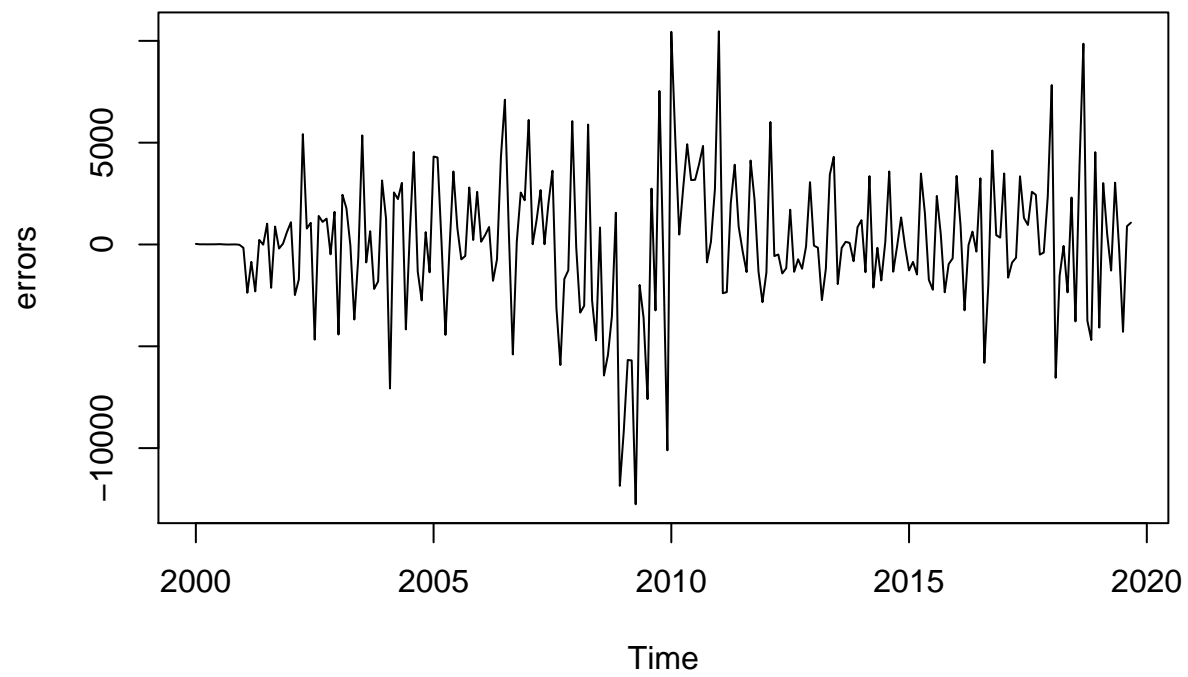
This is not a suitable model as Ljung-Box values are all below the threshold for group sizes 3 and above, as well as the fact that autocorrelations of residuals are outside of the 95% confidence interval for multiple lags.

Instead, let's try an $ARIMA(0, 1, 2)x(0, 1, 0)_{12}$ model.

```
z2ARIMA012x010_12model <- arima(z2, order=c(0,1,2), seasonal = list(order=c(0,1,0), period=12))  
  
tsdiag(z2ARIMA012x010_12model, 40)
```

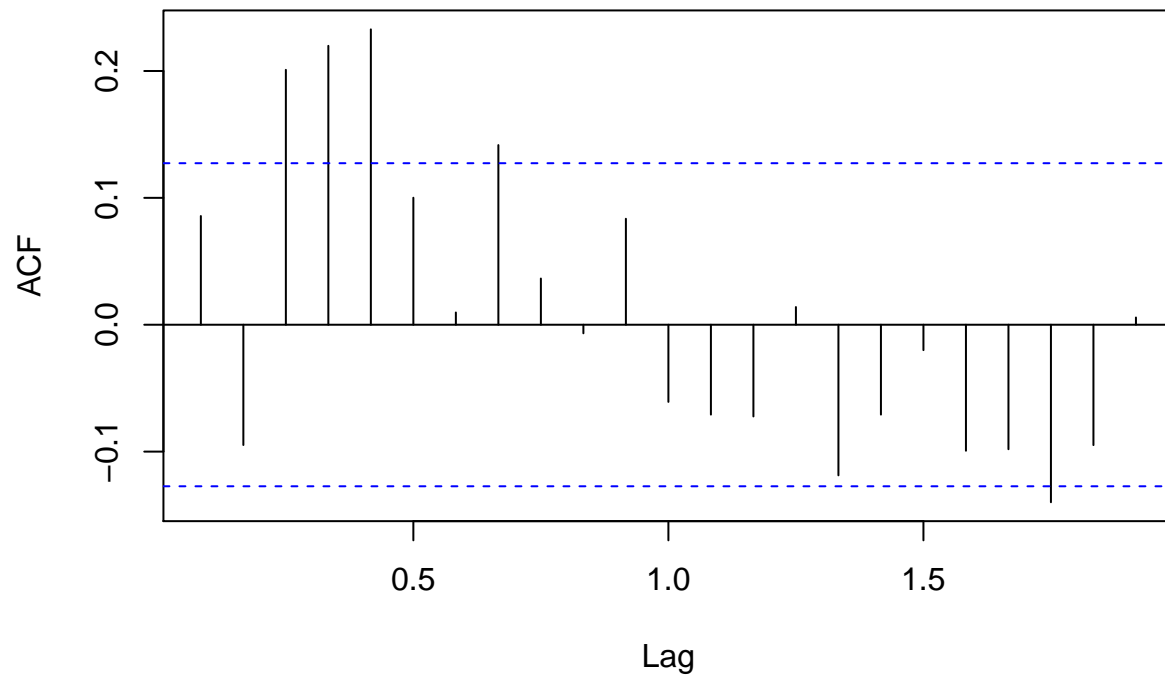


```
errors <- residuals(z2ARIMA012x010_12model)
plot(errors)
```



```
acf(errors)
```

Series errors



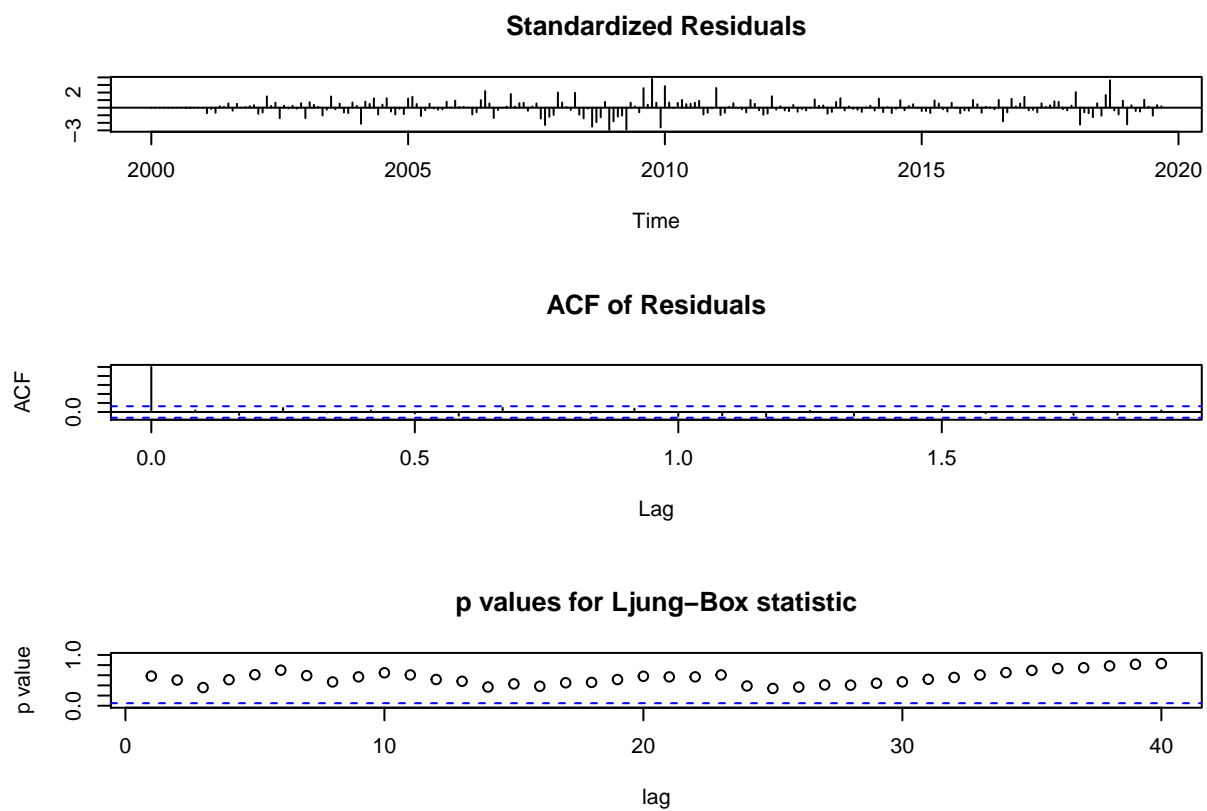
```
Box.test(errors, 60, type = "Ljung-Box", fitdf=2)
```

```
##  
## Box-Ljung test  
##  
## data: errors  
## X-squared = 107.85, df = 58, p-value = 7.756e-05
```

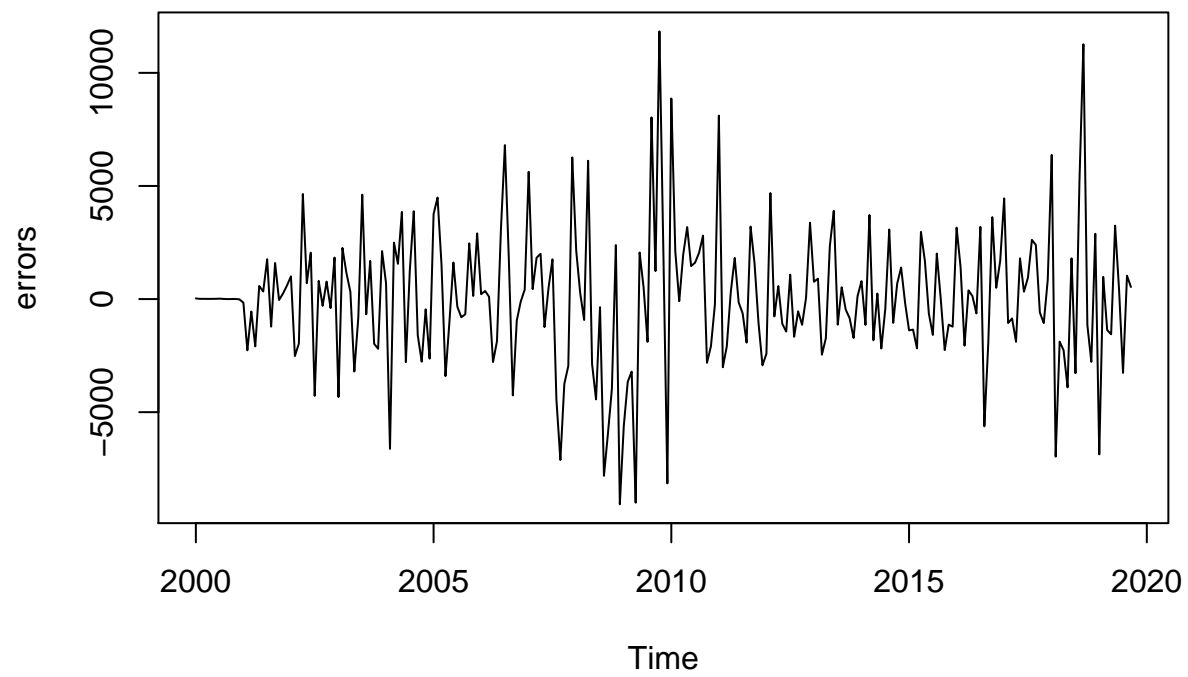
Also not a suitable model as Ljung-Box values are all below the threshold for group sizes 3 and above, as well as the fact that autocorrelations of residuals are outside of the 95% confidence interval for multiple lags.

Let's try an $ARIMA(1, 1, 2)x(0, 1, 0)_{12}$ model.

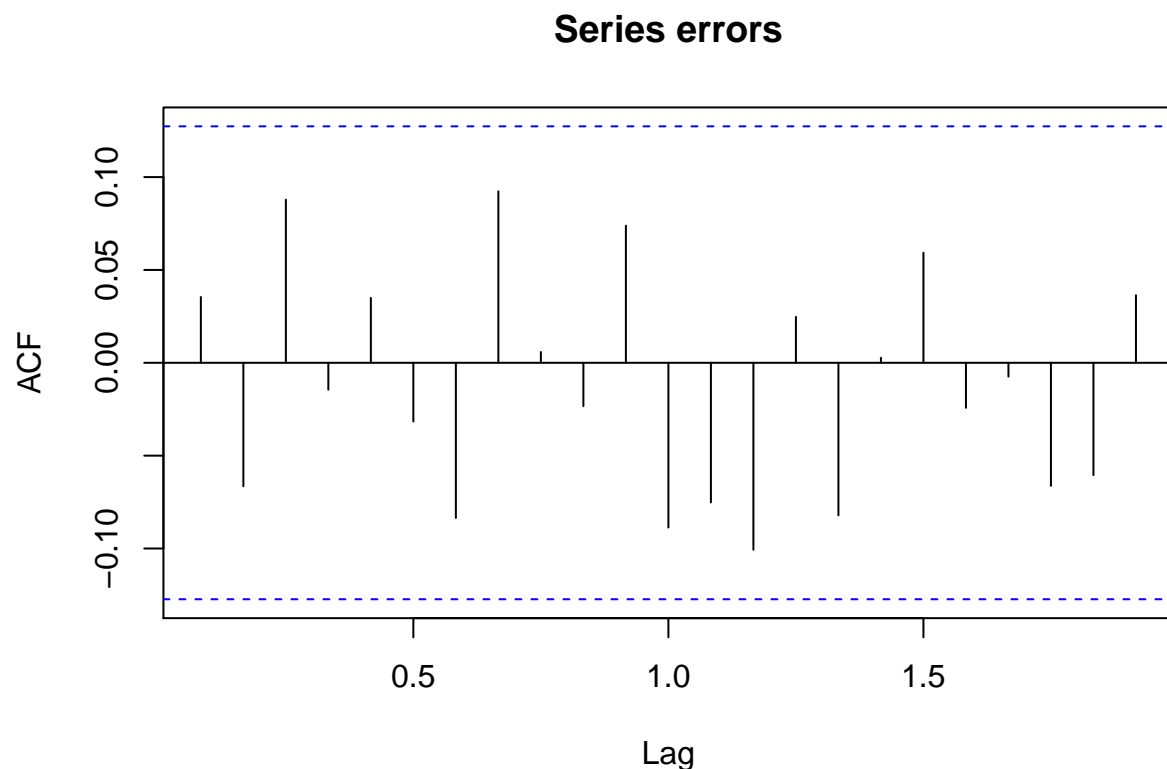
```
z2ARIMA112x010_12model <- arima(z2, order=c(1,1,2), seasonal = list(order=c(0,1,0), period=12))  
  
tsdiag(z2ARIMA112x010_12model, 40)
```

```
errors <- residuals(z2ARIMA112x010_12model)
plot(errors)
```



```
acf(errors)
```



```
Box.test(errors, 60, type = "Ljung-Box", fitdf=3)
```

```
##
## Box-Ljung test
##
## data: errors
## X-squared = 46.703, df = 57, p-value = 0.833
```

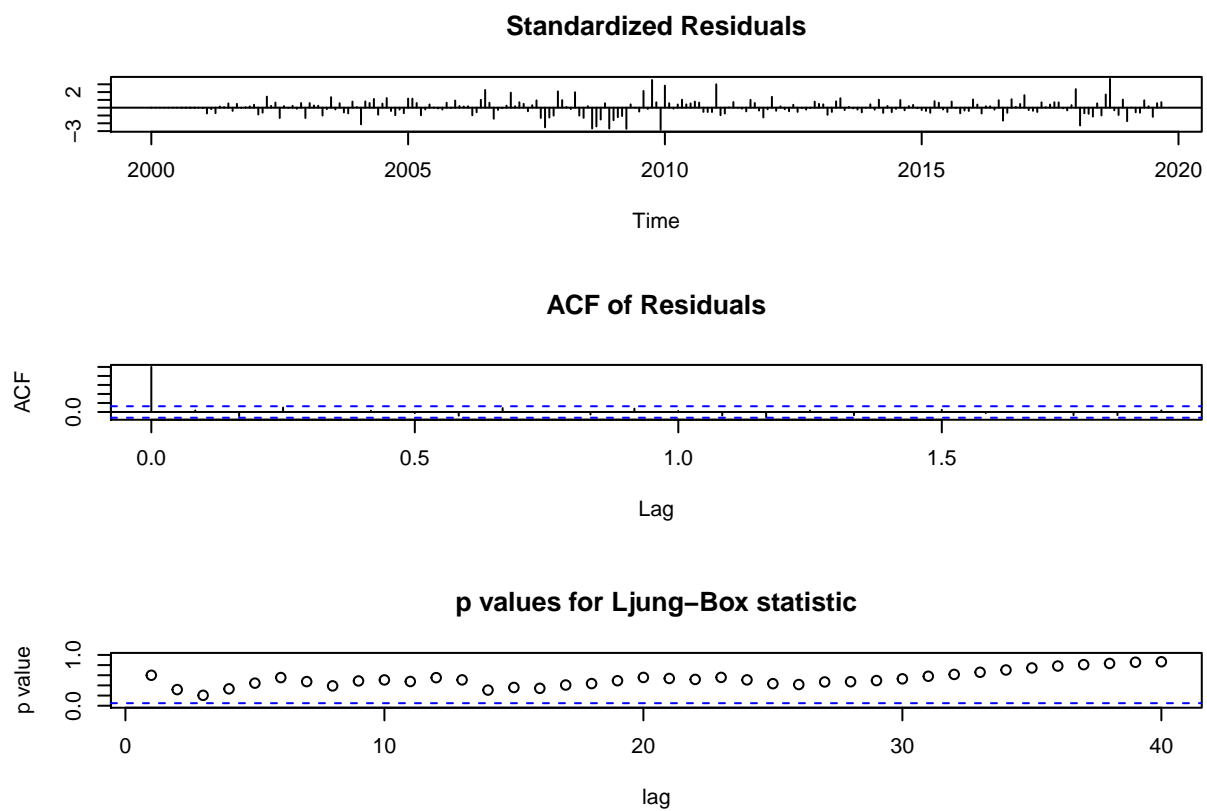
This looks to be a strong model based on the residuals resembling white noise. In addition, Ljung-Box values are above the significance threshold.

This is confirmed by examining the Ljung-Box test p-value, which equals 0.833 for group size 60. Thus we retain the null hypothesis that the time series is iid, i.e., white noise, since the p-value > 0.05. Furthermore, all autocorrelations of residuals are within the 95% confidence interval.

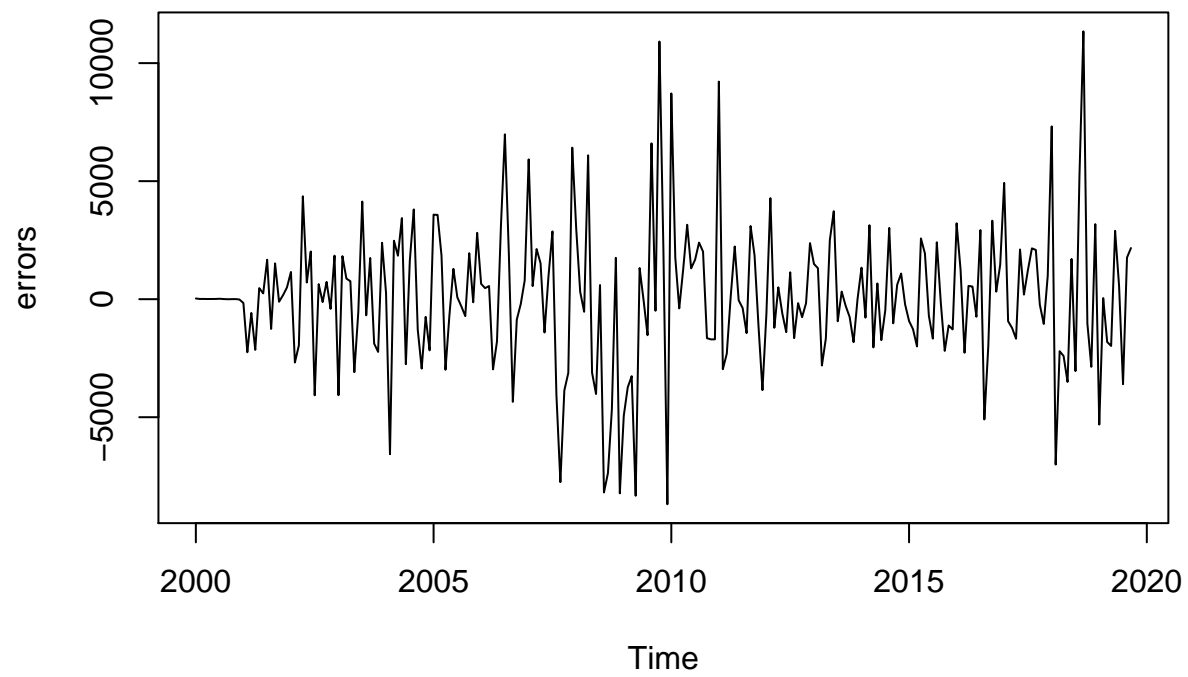
Just to be thorough, I also examined an $ARIMA(1, 1, 2)x(1, 1, 1)_{12}$ model.

```
z2ARIMA112x111_12model <- arima(z2, order=c(1,1,2), seasonal = list(order=c(1,1,1), period=12))

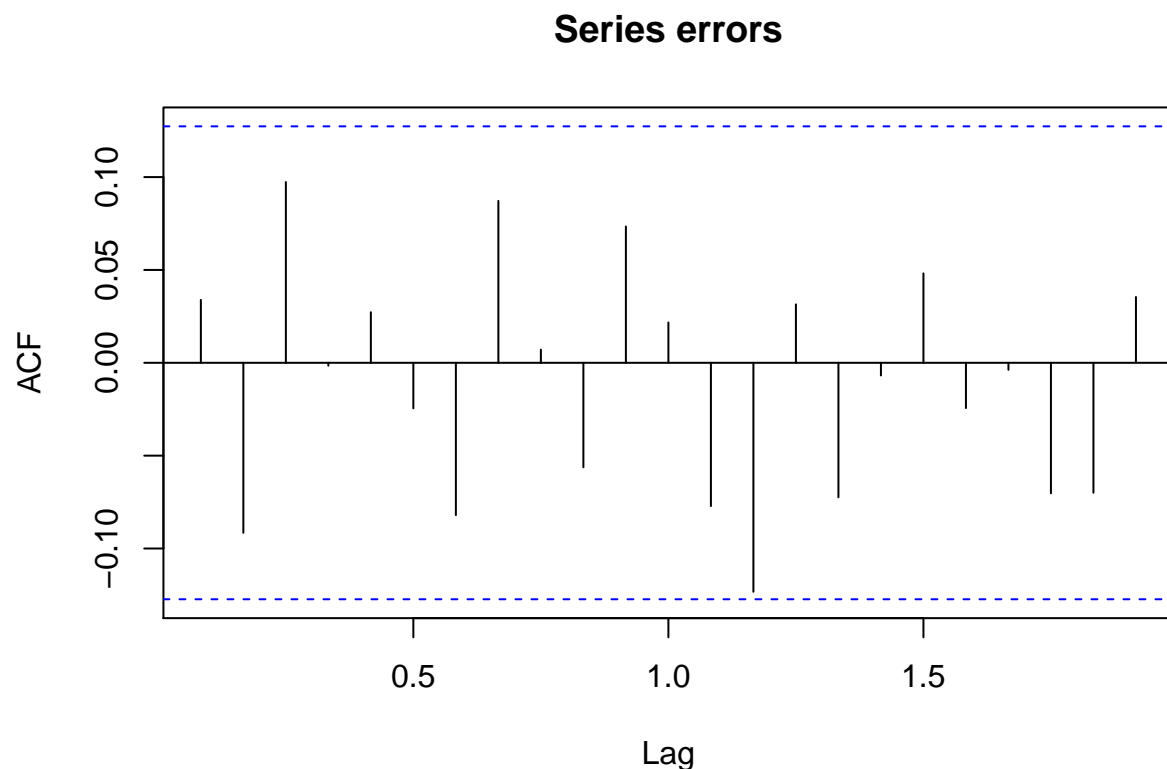
tsdiag(z2ARIMA112x111_12model, 40)
```



```
errors <- residuals(z2ARIMA112x111_12model)
plot(errors)
```



```
acf(errors)
```



```
Box.test(errors, 60, type = "Ljung-Box", fitdf=5)
```

```
##
## Box-Ljung test
##
## data: errors
## X-squared = 48.032, df = 55, p-value = 0.7358
```

This is also a good model based on the residuals resembling white noise. In addition, Ljung-Box values are above the significance threshold.

This is confirmed by examining the Ljung-Box test p-value, which equals 0.7358 for group size 60. Thus we retain the null hypothesis that the time series is iid, i.e., white noise, since the p-value > 0.05. Furthermore, all autocorrelations of residuals are within the 95% confidence interval.

I examined nearly a dozen other models before writing this report, but the two models immediately above were the only two models that were suitable for predictions. So let's compare the two:

```
AIC(z2ARIMA112x010_12model, z2ARIMA112x111_12model)
```

```
##                df      AIC
## z2ARIMA112x010_12model  4 4249.157
## z2ARIMA112x111_12model  6 4247.928
```

```
BIC(z2ARIMA112x010_12model, z2ARIMA112x111_12model)
```

```
##                df        BIC
## z2ARIMA112x010_12model    4 4262.804
## z2ARIMA112x111_12model    6 4268.398
```

I chose to use the $ARIMA(1,1,2)x(0,1,0)_{12}$ model because it has a lower BIC and fewer parameters than the $ARIMA(1,1,2)x(1,1,1)_{12}$ model.

```
z2ARIMA112x010_12model
```

```
##
## Call:
## arima(x = z2, order = c(1, 1, 2), seasonal = list(order = c(0, 1, 0), period = 12))
##
## Coefficients:
##          ar1          ma1          ma2
##         0.8335      -1.4746      0.7274
## s.e.   0.0498      0.0611      0.0511
##
## sigma^2 estimated as 9718086:  log likelihood = -2120.58,  aic = 4247.16
```

The equation of an $ARIMA(1,1,2)x(0,1,0)_{12}$ model is the following:

$$(1 - \phi_1 B)(1 - B)(1 - B^{12})Z_t = (1 + \theta_1 B + \theta_2 B^2)A_t$$

This is equivalent to

$$Z_t = (1 + \phi_1)Z_{t-1} - \phi_1 Z_{t-2} + Z_{t-12} - (1 + \phi_1)Z_{t-13} + \phi_1 Z_{t-14} + A_t + \theta_1 A_{t-1} + \theta_2 A_{t-2}$$

$\phi_1 = 0.8335$, $\theta_1 = -1.4746$ and $\theta_2 = 0.7274$.

Thus the equation of this model is:

$$Z_t = 1.8335Z_{t-1} - 0.8335Z_{t-2} + Z_{t-12} - 1.8335Z_{t-13} + 0.8335Z_{t-14} + A_t - 1.4746A_{t-1} + 0.7274A_{t-2}$$

Now, let's predict the next four values.

```
predict(z2ARIMA112x010_12model, 4)
```

```
## $pred
##           Jan Feb Mar Apr May Jun Jul Aug Sep           Oct           Nov           Dec
## 2019                                280007.0 277581.1 293772.8
## 2020 315146.2
##
## $se
##           Jan Feb Mar Apr May Jun Jul Aug Sep           Oct           Nov           Dec
## 2019                                3117.384 3312.091 3732.415
## 2020 4343.966
```

The following is a graph of the actual and fitted values for all observations in the time series:

```

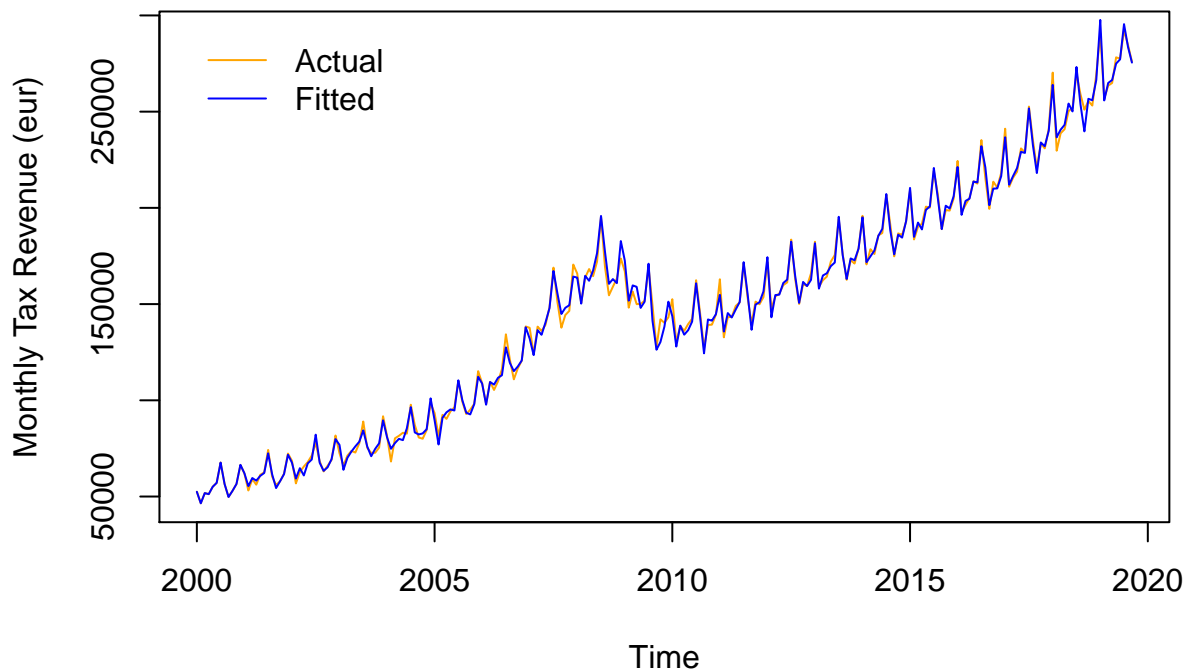
plot(z2, col="orange", main="GF027: State Budget Tax Revenues by Year and Month",
     ylab="Monthly Tax Revenue (eur)")

lines(fitted(z2ARIMA112x010_12model), col="blue")

legend("topleft", inset=.02, legend=c("Actual", "Fitted"),
      col=c("orange","blue"),lty=1, box.lty=0)

```

GF027: State Budget Tax Revenues by Year and Month



The 12 most recent actual and fitted values, and predicted values with 95% prediction intervals:

```

# last 12 actual values and next four predictions with 95% confidence intervals
pred2 <- plot(z2ARIMA112x010_12model, n.ahead=4,
             main="Last 12 Actual and Fitted Values and Next 4 Predictions",
             ylab='Monthly Tax Revenue (eur)', xlab='Month', n1=c(2018, 10), pch=19)

# add fitted values
points(fitted(z2ARIMA112x010_12model), col="blue")
lines(fitted(z2ARIMA112x010_12model), col="blue")

lines(pred2$lower, col="green")
lines(pred2$upper, col="green")

legend("topleft", inset=0.02,
      legend=c("Actual", "Fitted", "Predicted", "95% Confidence Interval"),
      col=c("black","blue","black", "green"), cex=.5, lty=c(1,1,2,1))

```


Last 12 Actual and Fitted Values and Next 4 Predictions

