Final assignment

Stefania Cucca, matr. 50050090

2022-05-24

Number of nights spent by non-residents by month and region

Gothenburg, Sweden - Jan 2016/Dec 2021

Sweden has been ranked as the most sustainable country for travel in a new index. Following the Scandinavian nation in Euromonitor's Top Countries for Sustainable Travel are Finland (second) and Austria (third). Rounding out the top five is Estonia and Norway.

Gothenburg, abbreviated **Gbg**; is the second-largest city in Sweden, fifth-largest in the Nordic countries, and capital of the Västra Götaland County. It is situated by Kattegat, on the west coast of Sweden, and has a population of approximately 590,000 in the city proper and about 1.1 million inhabitants in the metropolitan area.

According to the Global Destination Sustainability Index, the city of Gothenburg has been the world's most sustainable city four years running. Over half of its public transport energy comes from renewable sources. Furthermore, all meat served within Gothenburg must be organically raised.

Ever since 2016, Gothenburg has held the number 1 ranking of the Global Destination Sustainability Index.

Being an important destination in the framework of sustainable tourism, I wanted to investigate the situation regarding the Hospitality industry, referring in particular to the number of nights spent of all hotels, holiday villages, youth hostels, camping sites and commercially arranged private cottages and apartments by non-residents.

How many non-residents spend a night in the city of Gothenburg? How many choose it as a sustainable destination?

The website of Official Statistics of Sweden, SCB, provides such type of data.

With a simple research in their statistical database, in the Business activities and Accommodation statistics sections, I chose monthly data, from January 2016 to December 2021, regarding the number of nights spent in the area of Greater Gothenburg by non-residents.

https://www.statistikdatabasen.scb.se/pxweb/en/ssd/START___NV___NV1701___NV1701B/NV1701T4M/table/tableViewLayout1/

Representation of the data

I downloaded the table and named it "Downloadnights" as an xlsx file. Then I uploaded it on R and renamed it as "Dati".

```
getwd()
```

```
## [1] "/cloud/project/Forecasting - Stefania"
```

```
Dati <- read.xlsx("Downloadnights.xlsx")
Dati
```

```
##         X1 #Nights
## 1  2016M01   77492
## 2  2016M02   73194
## 3  2016M03   90535
## 4  2016M04   98287
## 5  2016M05  132276
## 6  2016M06  162644
## 7  2016M07  287123
## 8  2016M08  202326
## 9  2016M09  120691
## 10 2016M10  110612
## 11 2016M11   94451
## 12 2016M12  103969
## 13 2017M01   71979
## 14 2017M02   76252
## 15 2017M03   87388
## 16 2017M04  101625
## 17 2017M05  120896
## 18 2017M06  167531
## 19 2017M07  270583
## 20 2017M08  193326
## 21 2017M09  132083
## 22 2017M10  105310
## 23 2017M11   87633
## 24 2017M12   84091
## 25 2018M01   68781
## 26 2018M02   76200
## 27 2018M03   97168
## 28 2018M04   91018
## 29 2018M05  133200
## 30 2018M06  169352
## 31 2018M07  266238
## 32 2018M08  200726
## 33 2018M09  119146
## 34 2018M10  114910
## 35 2018M11   96239
## 36 2018M12   88755
## 37 2019M01   75274
## 38 2019M02   77118
## 39 2019M03   82621
## 40 2019M04  106166
## 41 2019M05  116984
## 42 2019M06  164971
## 43 2019M07  293129
## 44 2019M08  212396
## 45 2019M09  120862
## 46 2019M10  105174
## 47 2019M11   93076
## 48 2019M12   84066
## 49 2020M01   79794
## 50 2020M02   77194
## 51 2020M03   27638
## 52 2020M04    7373
## 53 2020M05   10890
```

```
## 54 2020M06   14810
## 55 2020M07   33675
## 56 2020M08   34222
## 57 2020M09   28906
## 58 2020M10   39340
## 59 2020M11   19242
## 60 2020M12   14427
## 61 2021M01   11747
## 62 2021M02   12671
## 63 2021M03   14933
## 64 2021M04   19493
## 65 2021M05   23586
## 66 2021M06   37836
## 67 2021M07  114818
## 68 2021M08  104324
## 69 2021M09   52533
## 70 2021M10   73513
## 71 2021M11   71768
## 72 2021M12   51986
```

The two columns represent:

- **The Year and the Month**: 2016-2021 = 5 years 1-12 (Jan–Dec)
- **#Nights**: the number of nights spent in the area of Greater Gothenburg by non-residents each month.

The table Dati was recognized by R as a data.frame object.

```
class(Dati)
```

```
## [1] "data.frame"
```

The time series object

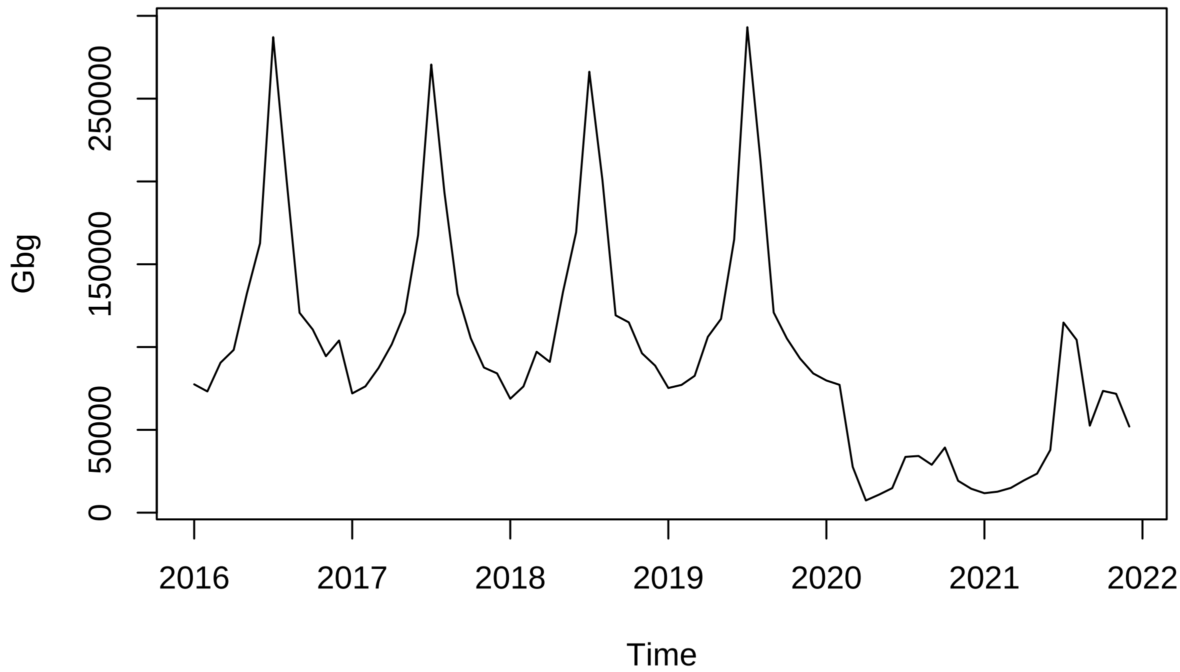
At this point I created a time series object, Gbg, from the data.frame object Dati, so that I could start working on it. The series had to start from January 2016 with a monthly frequency.

```
Gbg <- ts(Dati$`#Nights`, start=c(2016,1), frequency = 12)
Gbg
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2016  77492  73194  90535  98287 132276 162644 287123 202326 120691 110612
## 2017  71979  76252  87388 101625 120896 167531 270583 193326 132083 105310
## 2018  68781  76200  97168  91018 133200 169352 266238 200726 119146 114910
## 2019  75274  77118  82621 106166 116984 164971 293129 212396 120862 105174
## 2020  79794  77194  27638   7373  10890  14810  33675  34222  28906  39340
## 2021  11747  12671  14933  19493  23586  37836 114818 104324  52533  73513
##         Nov    Dec
## 2016  94451 103969
## 2017  87633  84091
## 2018  96239  88755
## 2019  93076  84066
## 2020  19242  14427
## 2021  71768  51986
```

This was the visualization of the time series object Gbg.

```
plot(Gbg)
```



Confirmation I created a time series object.

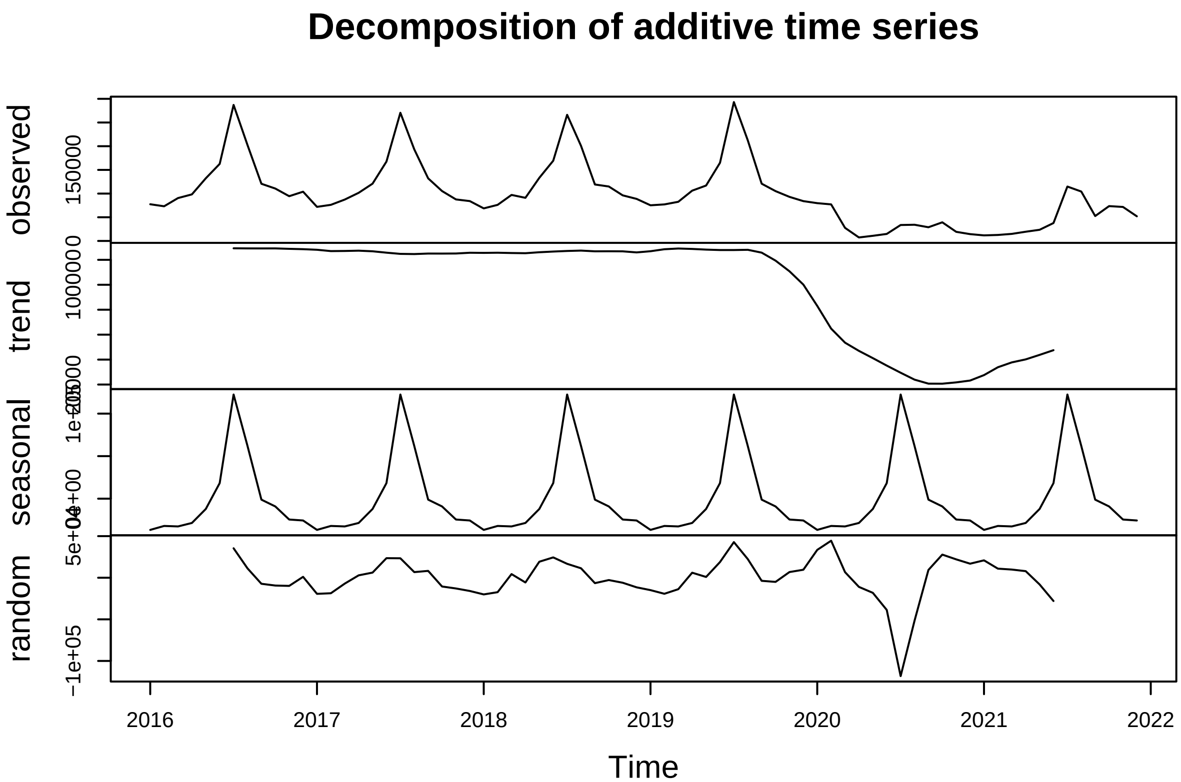
```
class(Gbg)
```

```
## [1] "ts"
```

Decompositon of the time series

The time series object GBG allowed me to proceed with the decomposition of the time series, which I called DecGbg:

```
DecGbg <- decompose(Gbg)
plot(DecGbg)
```



The following could be observed:

- The trend decreased quite rapidly from the year 2020, when the Covid pandemic started, which was already visible when plotting the whole series. It began to slightly increase in the year 2021, when most countries loosened the restrictions against the spread of the virus and most of the people could travel again, and thus spend nights as non-residents.
- There was a strong seasonal component, indicating the repetition of the same behaviour every year: the number of nights spent by non-residents in the area of Gothenburg reached its peak during the middle of the year, in the summer months, to decrease again in the winter months.
- For the first four years, the seasonal component did not follow the trend, and it remained constant.

Holt-Winters

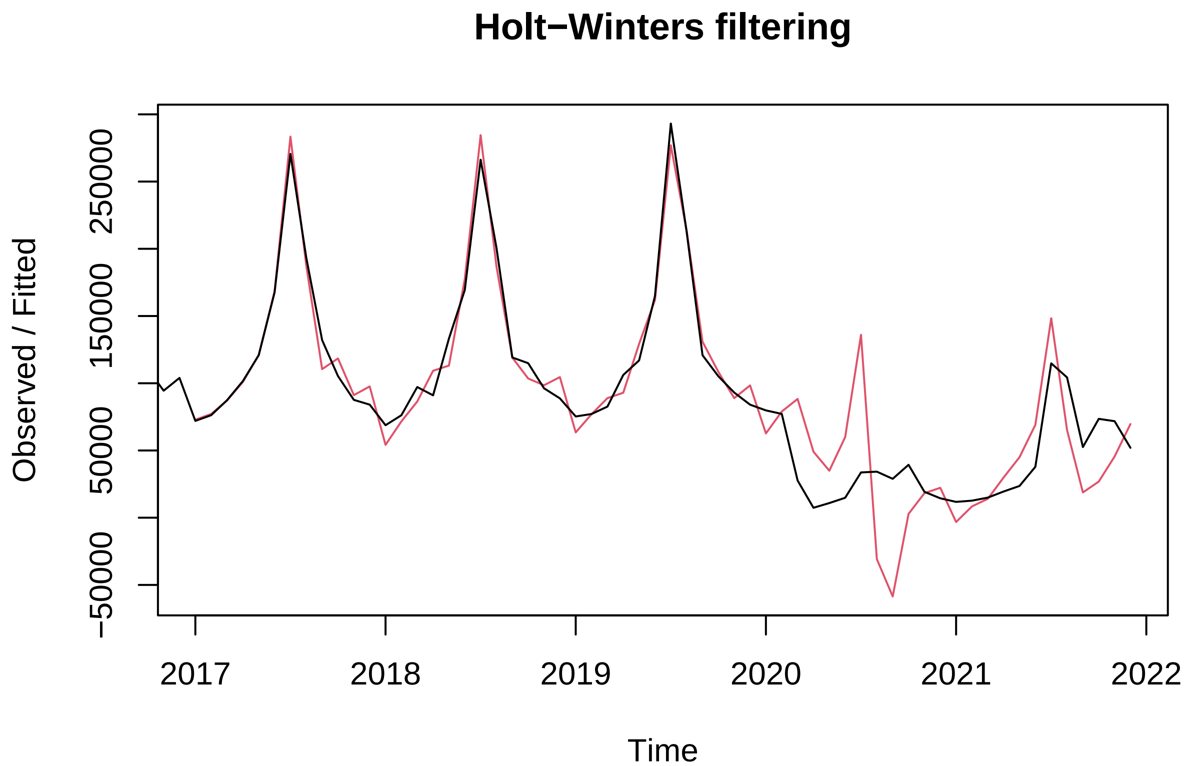
Additive I used the additive model, set as default, to make predictions 3 years ahead in the future.

```r
HWAddGbg <- HoltWinters(Gbg, alpha = NULL, beta = NULL,
                  gamma = NULL)
HWAddGbg
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = Gbg, alpha = NULL, beta = NULL, gamma = NULL)
##
## Smoothing parameters:
##  alpha: 0.849462
##  beta : 0
##  gamma: 1
##
## Coefficients:
```

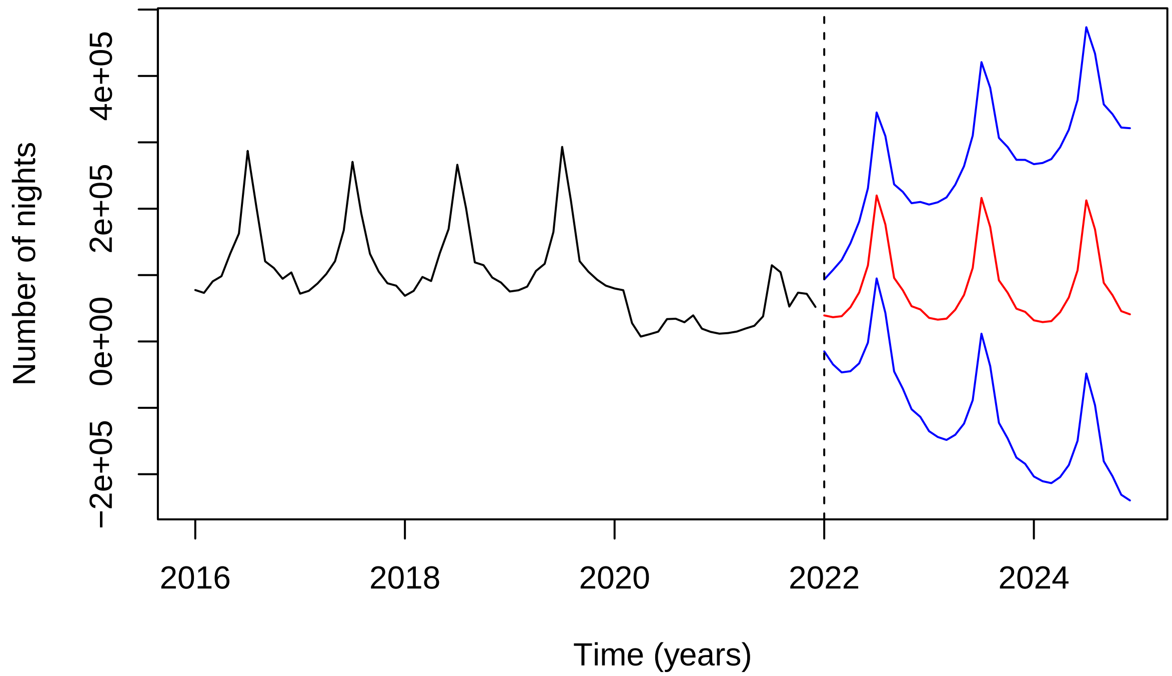
```
##              [,1]
## a     87695.9535
## b      -308.3795
## s1   -48132.1970
## s2   -50523.7024
## s3   -48723.4970
## s4   -34967.6989
## s5   -12367.8453
## s6    28695.1305
## s7   134448.2227
## s8    90975.4124
## s9    10824.4183
## s10   -7503.9611
## s11  -31232.8113
## s12  -35709.9535
```

```r
plot(HWAddGbg)
```




```r
Addpred <- predict(HWAddGbg, n.ahead = 12*3, prediction.interval = T)

ts.plot(cbind(Gbg, Addpred), xlab = "Time (years)",
        ylab="Number of nights",
        col=c("black","red", "blue", "blue"))
abline(v = 2022, lty = 2)
```



The prediction made with the Holt Winters function resembled what happened in the past (the shape of the curve was more similar to the ones in the years 2016-2019, rather than to what it is shown in the years 2020-2021), influenced by what happened in the last two years (meaning that the spikes were not as high as in the first years).

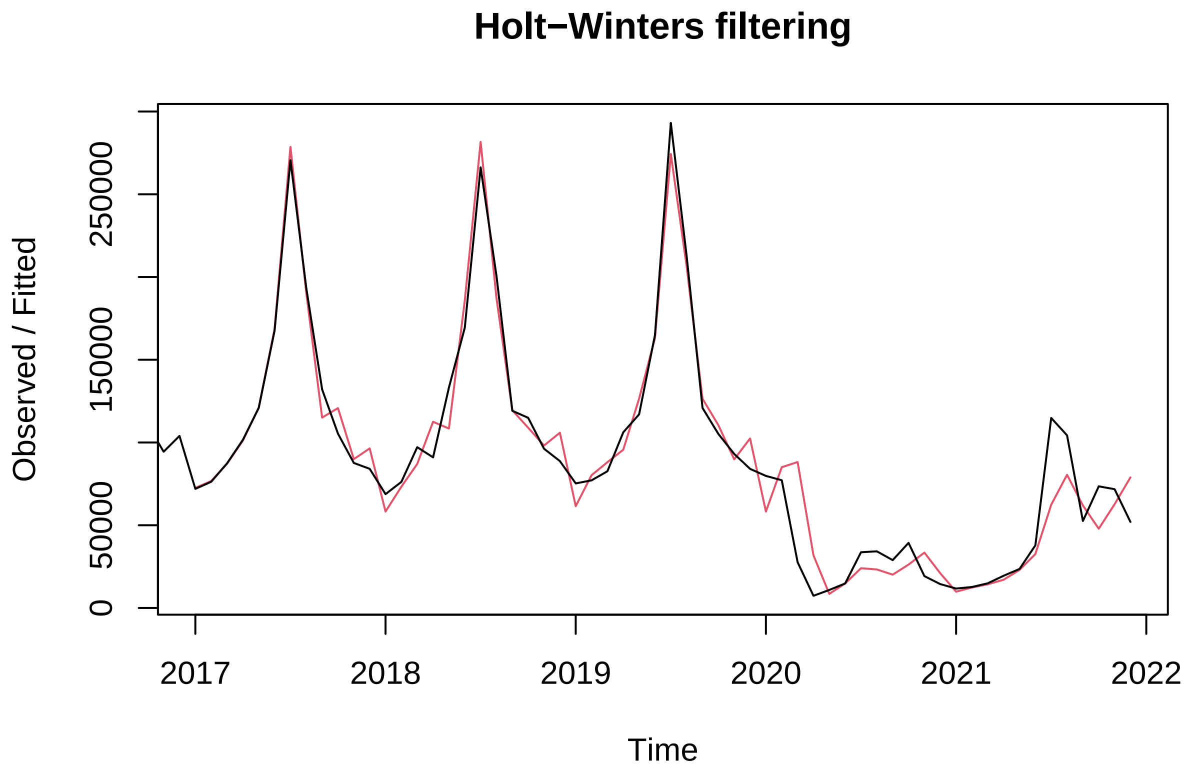
Multiplicative I did the same prediction, but using the multiplicative model.

```
HWMultGbg <- HoltWinters(Gbg, alpha = NULL, beta = NULL,
                  gamma = NULL, seasonal="multiplicative")
HWMultGbg
```

```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
##
## Call:
## HoltWinters(x = Gbg, alpha = NULL, beta = NULL, gamma = NULL,     seasonal = "multiplicative")
##
## Smoothing parameters:
##  alpha: 1
##  beta : 0
##  gamma: 0.2591274
##
## Coefficients:
##              [,1]
## a   64568.0349338
## b    -308.3795163
## s1      0.5595185
## s2      0.5977014
## s3      0.6844537
## s4      0.7943768
## s5      0.9487760
## s6      1.3263940
## s7      2.2113722
## s8      1.5595146
```

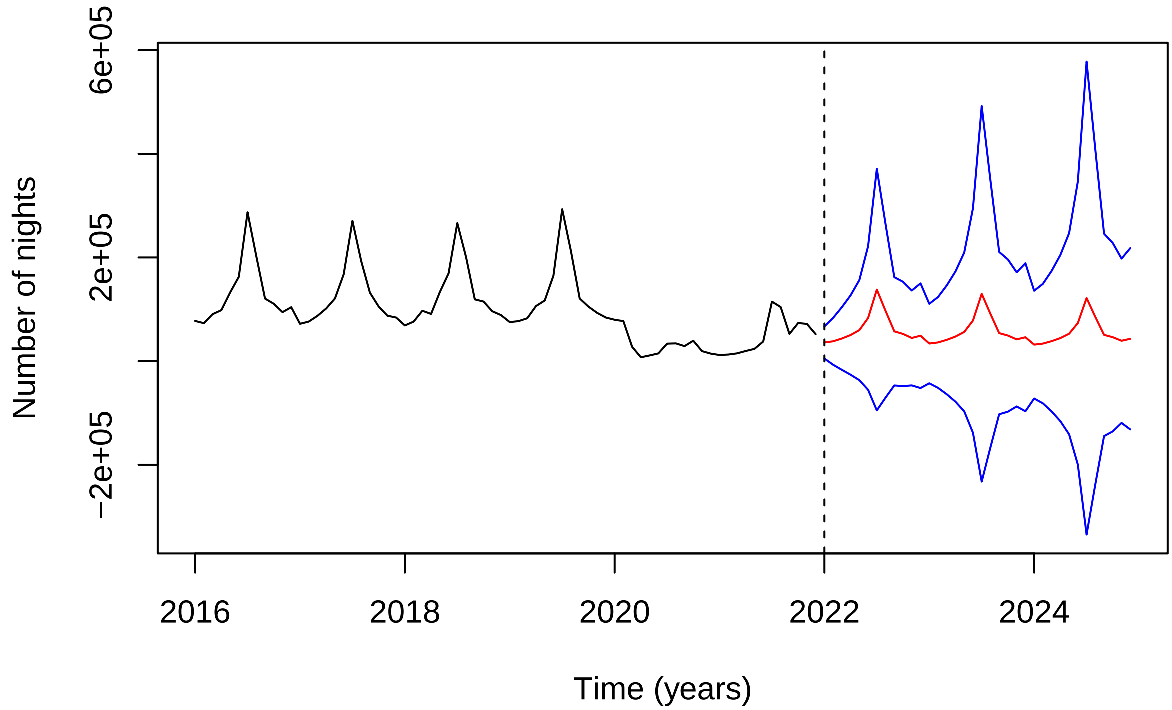
```
## s9        0.9303045
## s10       0.8525615
## s11       0.7298916
## s12       0.8051352
```

```r
plot(HWMultGbg)
```




```r
Multpred <- predict(HWMultGbg, n.ahead = 12*3, prediction.interval = T)

ts.plot(cbind(Gbg, Multpred), xlab = "Time (years)",
        ylab="Number of nights",
        col=c("black","red", "blue", "blue"))
abline(v = 2022, lty = 2)
```

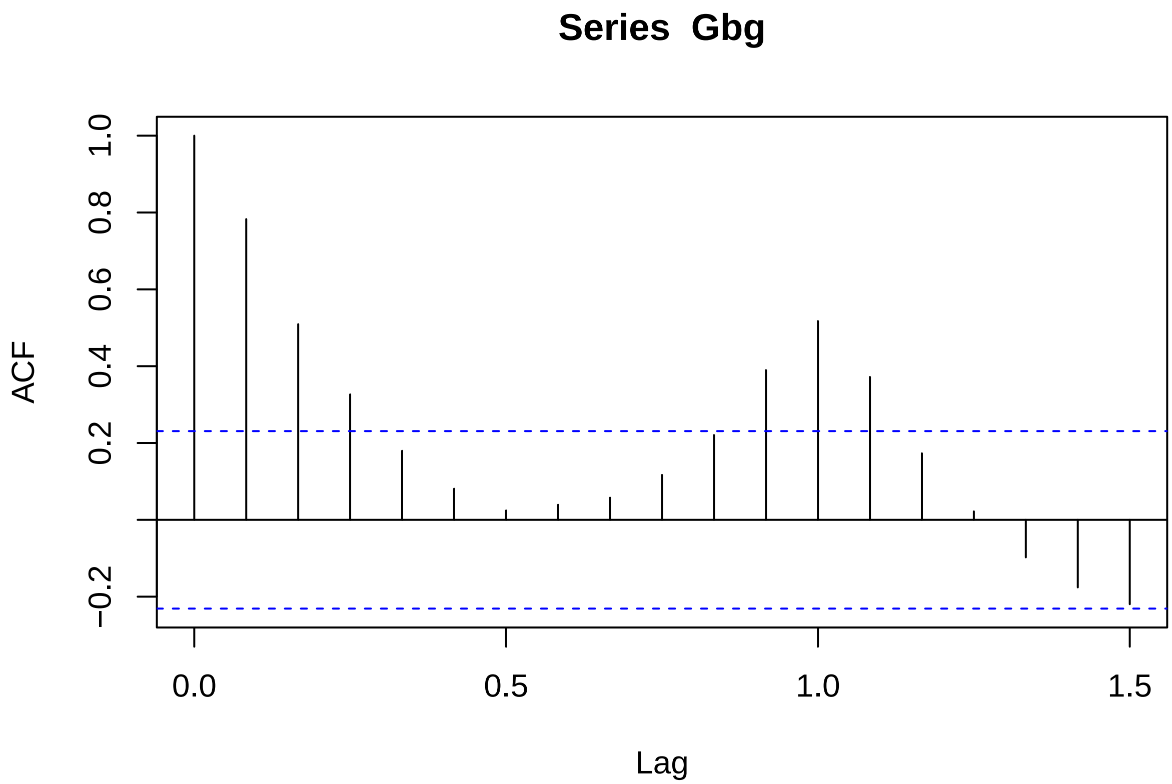


The prediction was strongly influenced by what had happened since 2020: the spikes indicating seasonality were much lower and flattened because of the declining trend of the last years.

Behaviour of acf and pacf

To assess the strength of the dependence between observations over time, I observed the acf:

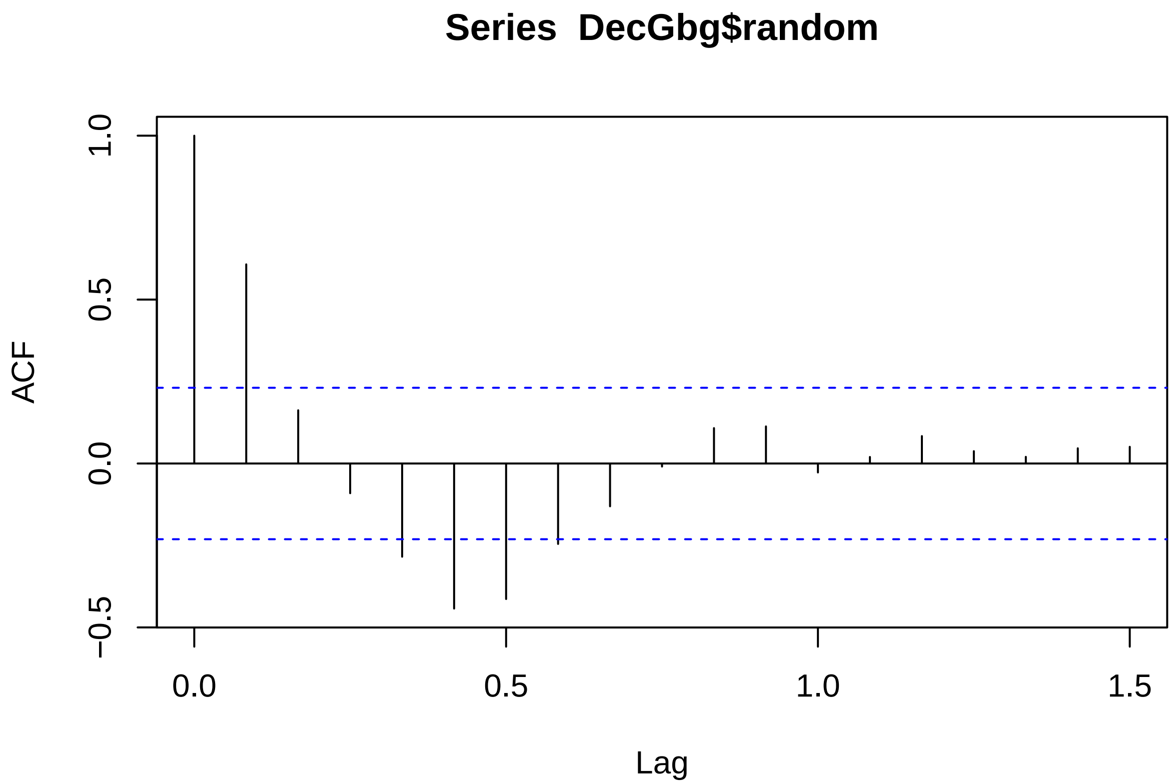
```
acf(Gbg)
```



It showed quite a strong dependence over time. The acf slowly decayed after lag 3, but it increased again at around lag 10. The increase at around lag 10 was due to the seasonality.

I wanted to see what the acf was only for the random component, purifying the series from the trend and the seasonality.

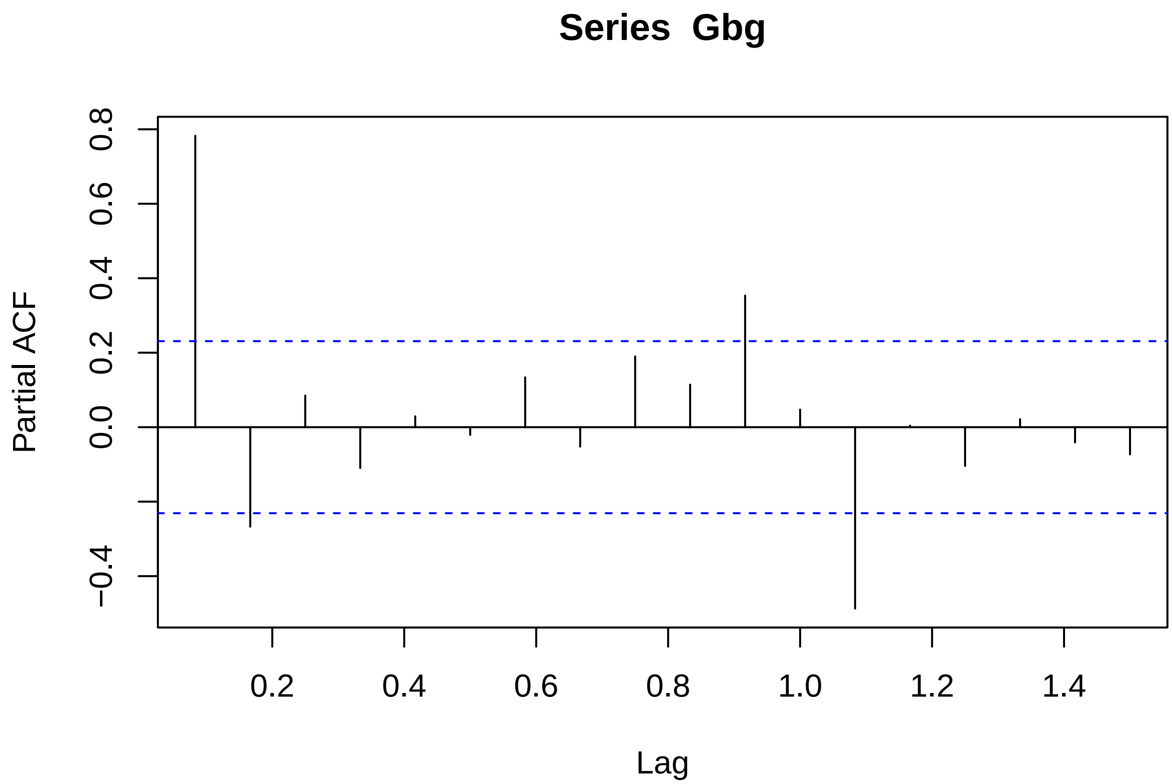
```
acf(DecGbg$random, na.action = na.pass)
```



There was still some dependence between observations over time even though the series was purified from the trend and the seasonality.

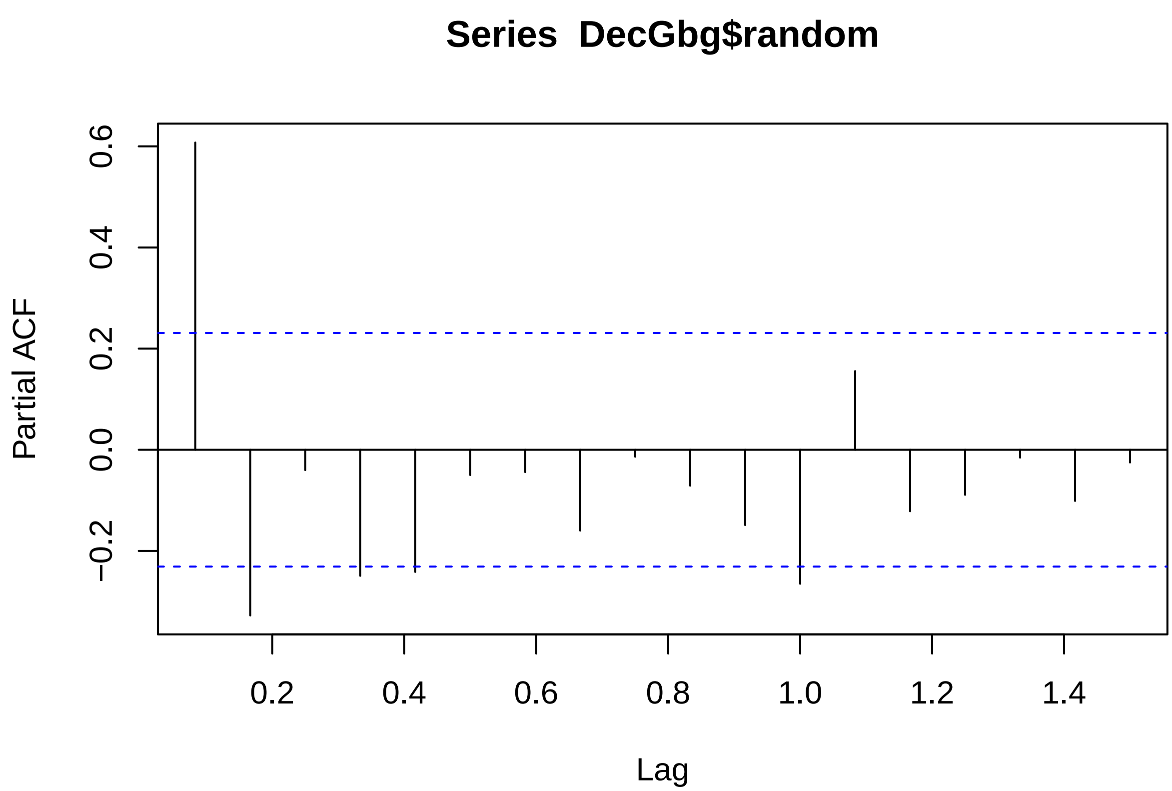
Then I observed the pacf for both the series and the series purified from the trend and the seasonality:

```
pacf(Gbg)
```



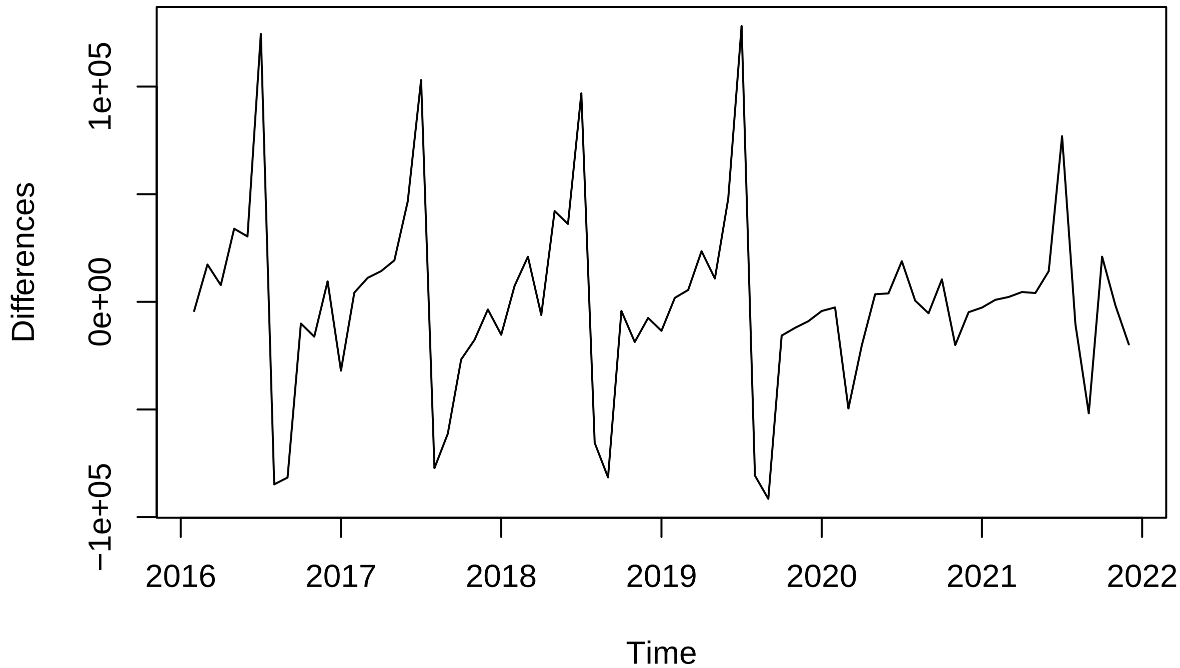
After the spike lag 0, the pacf seemed to have a cut-off, however, some spikes were still out of the blue lines.

```
pacf(DecGbg$random,na.action = na.pass)
```



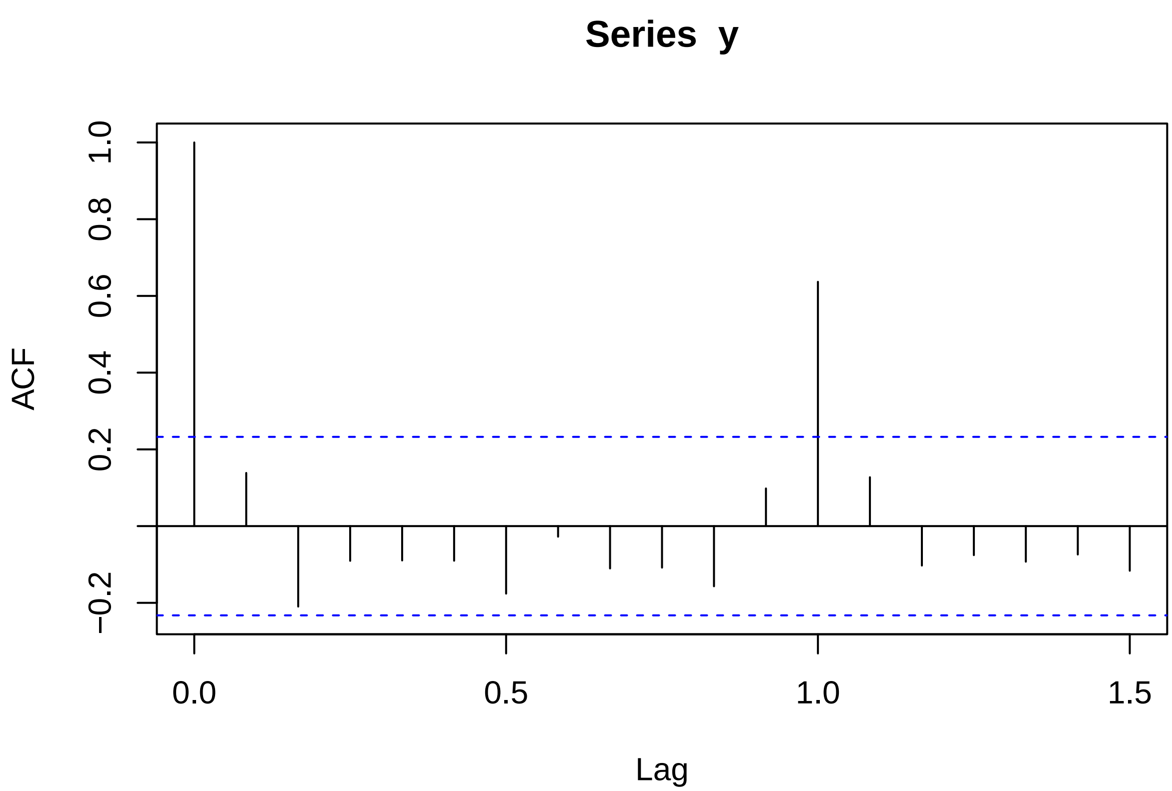
The same occurred when the series was purified from trend and seasonality. Having these results, I applied the first differences on the series, creating a new series called y.

```
y <- diff(Gbg)
plot(y, ylab="Differences", type="l")
```



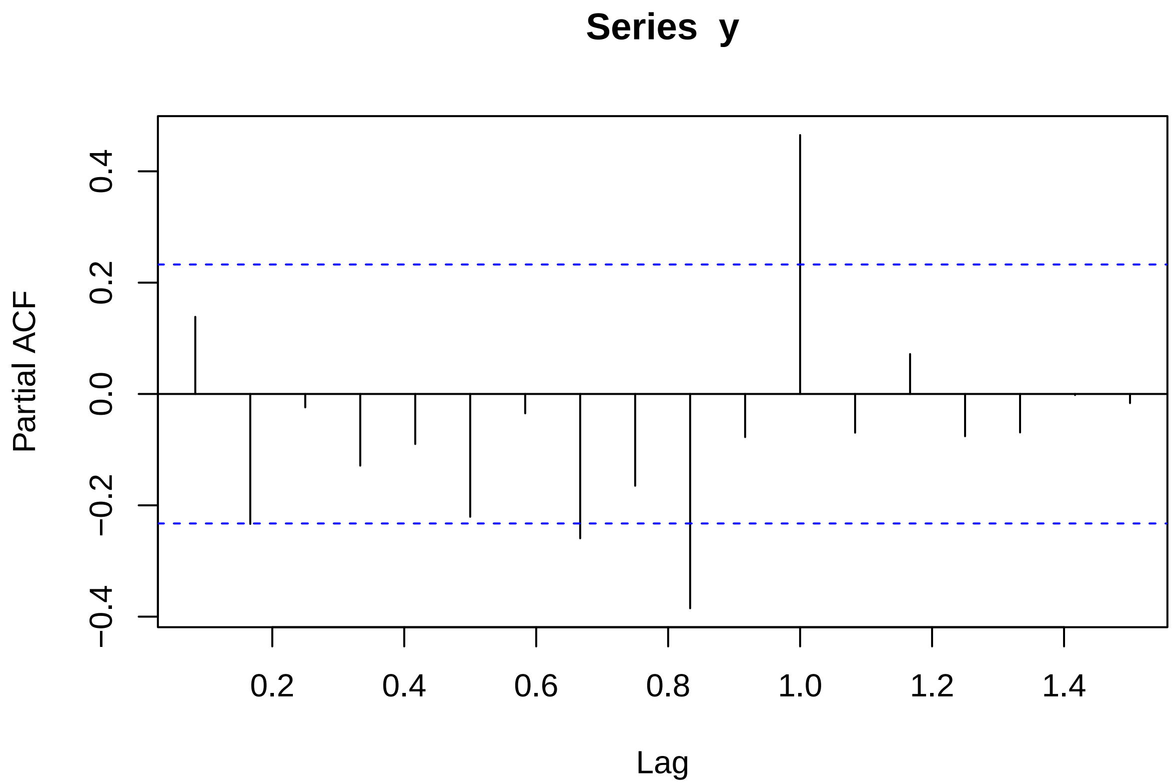
Then I observed again acf and pacf on the differentiated series.

```
acf(y)
```



Acf was equal to 1 at lag 0, then decreased to zero. The only exception was the spike outside the blue lines at lag 10.

```
pacf(y)
```



With regards to the pacf, it was almost always equal to zero, except for the spikes at around lag 1.

Test for stationarity

I used both the adf and the pp test to check the stationarity of the series.

```
adf.test(Gbg)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Gbg
## Dickey-Fuller = -3.9229, Lag order = 4, p-value = 0.01814
## alternative hypothesis: stationary
```

The p-value 0.01814 $<$ 0.05. So I failed to reject the null hypotesis and accepted the alternative hypothesis: stationarity.

```
pp.test(Gbg)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  Gbg
## Dickey-Fuller Z(alpha) = -23.263, Truncation lag parameter = 3, p-value
## = 0.02347
## alternative hypothesis: stationary
```

The p-value 0.02347 < 0.05, like with the other test. So I failed to reject the null hypotesis and accepted the alternative hypothesis: stationarity again.

I performed both tests also on the differentiated series y:

```
adf.test(y)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  y
## Dickey-Fuller = -4.5453, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
pp.test(y)
```

```
## Warning in pp.test(y): p-value smaller than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  y
## Dickey-Fuller Z(alpha) = -53.416, Truncation lag parameter = 3, p-value
## = 0.01
## alternative hypothesis: stationary
```

Consistently with the non differentiated series, the p-value was smaller than 0.05 in both test.

Choosing the model

To choose the model to fit the series, I considered the flow-chart based on Diggle (1990).

- Is the plot of the series stationary? Yes
- Acf decays to zero? Yes (after lag 3, but not totally, in the non differentiated series, then after lag 0 in y, after having applied the differences)
- Sharp cut off in acf? Yes in the differentiated series

These answers led me to choose a MA model. I considered also the fact that I applied the first differences on the series.

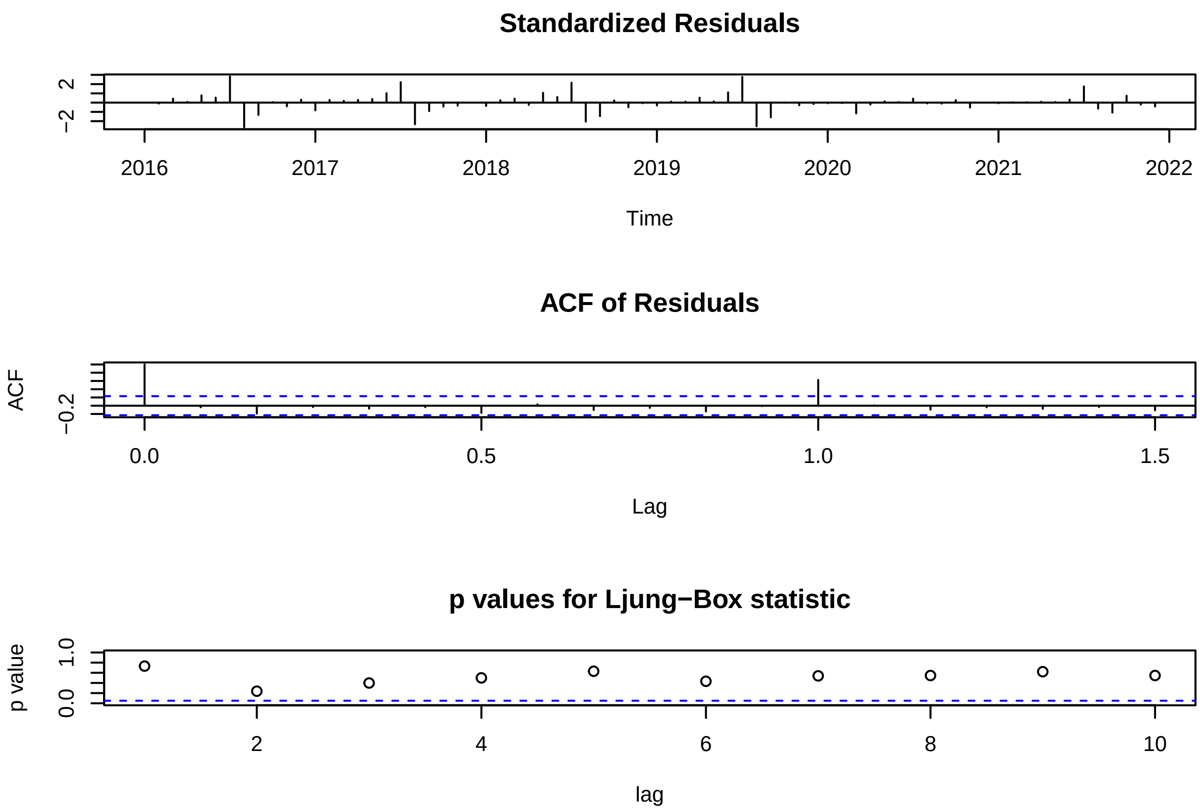
I tried to fit the series in some models.

For the first model, I took into consideration a MA component (as suggested by the flow-chart) and the first differences.

```
m <- arima(Gbg, order=c(0,1,1))
m
```

```
##
## Call:
## arima(x = Gbg, order = c(0, 1, 1))
##
## Coefficients:
##          ma1
##       0.2293
## s.e.  0.1428
##
## sigma^2 estimated as 1.781e+09:  log likelihood = -856.94,  aic = 1717.88
```

```r
tsdiag(m)
```



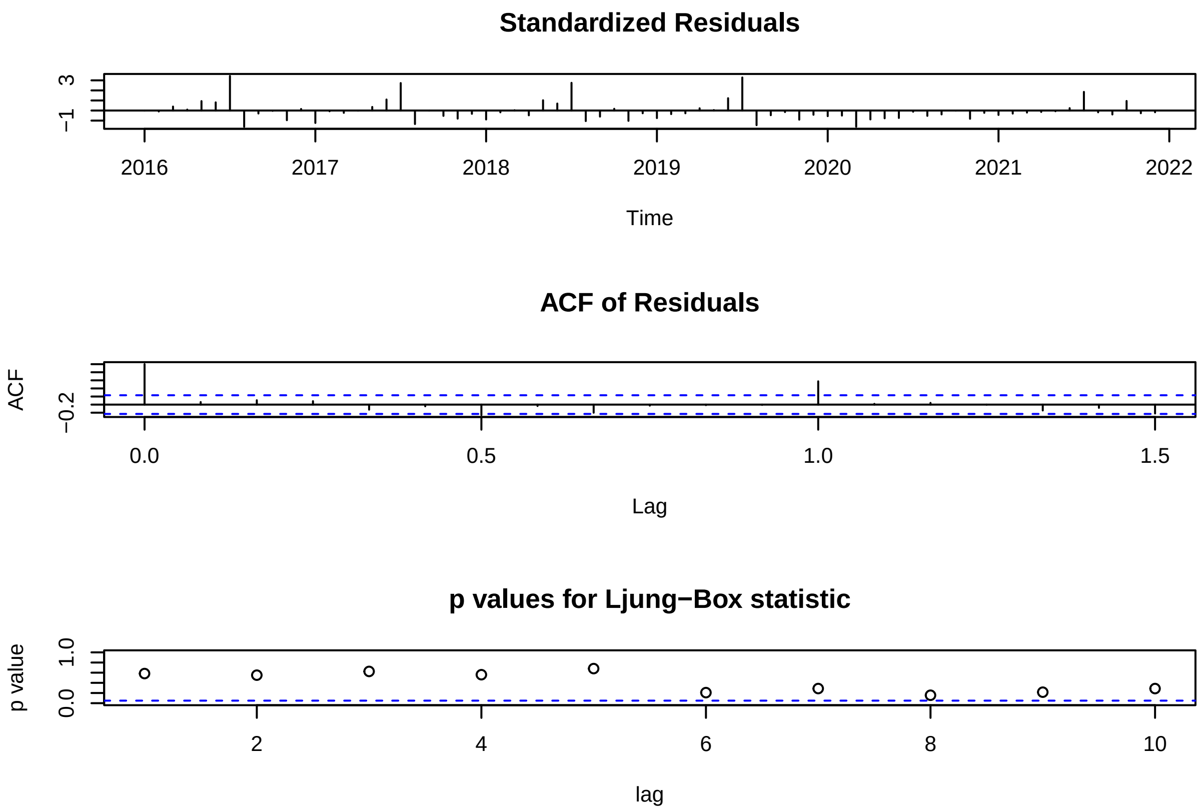
The test diagnostic showed very low p-values.

For the second model, I wanted to see what would happen if I increased the number of parameters to 3 (the lag after which the acf decayed).

```r
m1 <- arima(Gbg, order=c(0,1,3))
m1
```

```
##
## Call:
## arima(x = Gbg, order = c(0, 1, 3))
##
## Coefficients:
##         ma1      ma2      ma3
##       0.001  -0.4520  -0.2726
## s.e.  0.111   0.1495   0.1406
##
## sigma^2 estimated as 1.598e+09:  log likelihood = -853.47,  aic = 1714.95
```

```r
tsdiag(m1)
```



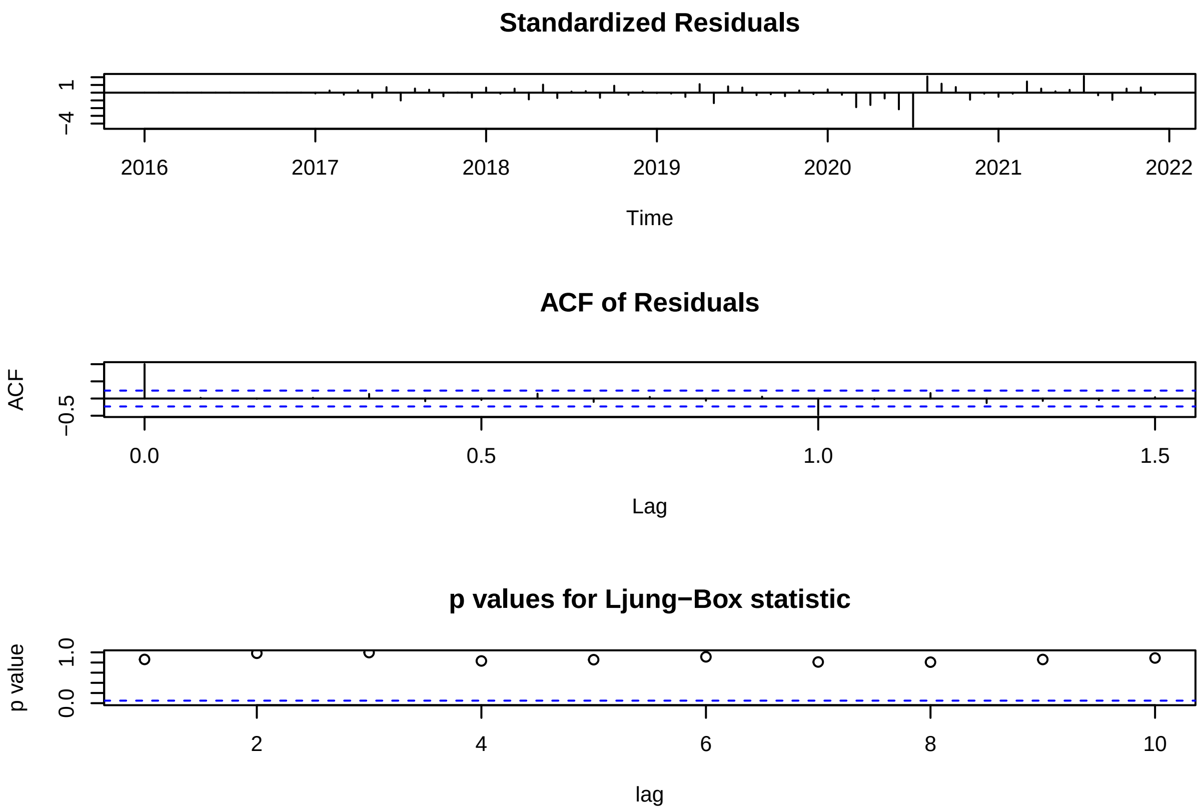
The AIC was slightly lower, but the p-values were still too low.

Dealing with a time series in which there was a strong seasonal component, I changed and used the sarima models, adding the seasonal part.

```
m2 <- arima(Gbg, order=c(1,0,3), seasonal=list(order=c(0,1,0)))
m2
```

```
##
## Call:
## arima(x = Gbg, order = c(1, 0, 3), seasonal = list(order = c(0, 1, 0)))
##
## Coefficients:
##          ar1     ma1      ma2     ma3
##       0.8237  0.2264  -0.1230  0.0839
## s.e.  0.1045  0.1600   0.1542  0.1861
##
## sigma^2 estimated as 752935398:  log likelihood = -699.1,  aic = 1408.2
```

```
tsdiag(m2)
```



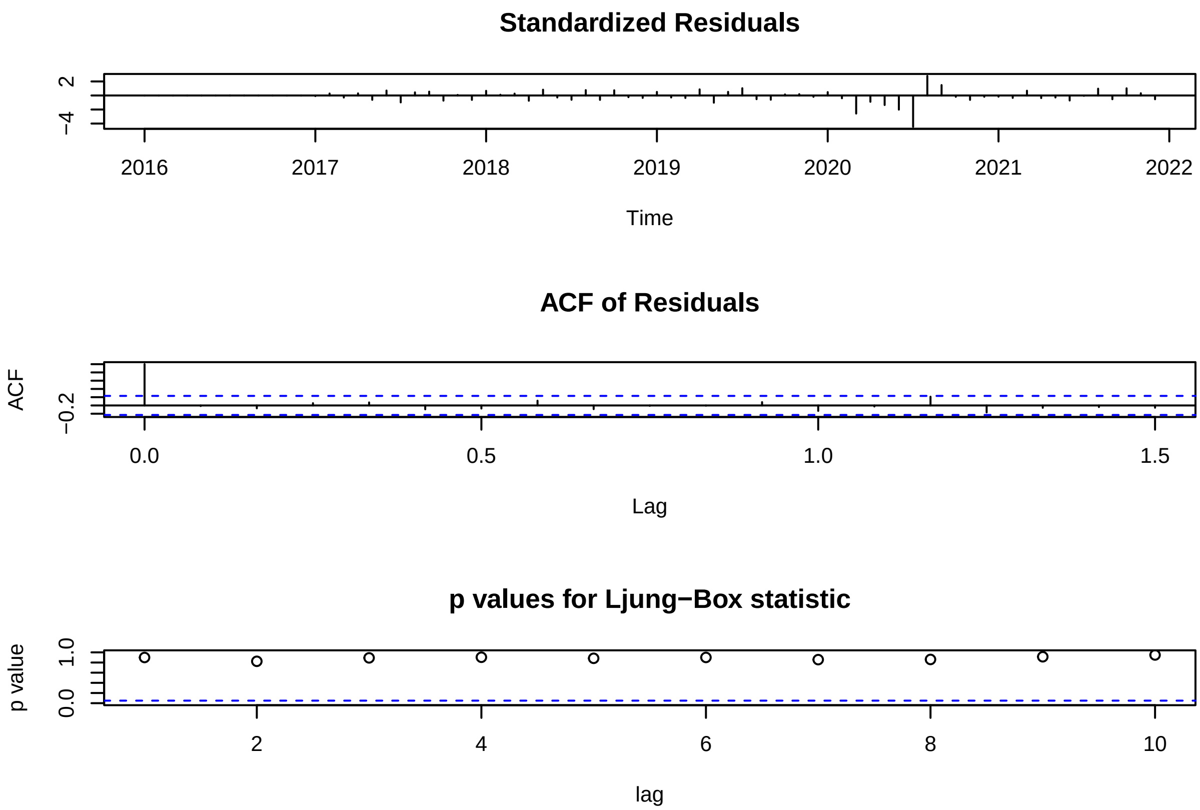
Increasing the number of parameters, the AIC was much lower than before and the p-values were high in the test diagnostic.

I tried to add one parameter in the autoregressive part of the non-seasonal parenthesis, since what happened each year depended on what happened the previous year.

```
m3 <- arima(Gbg, order=c(1,0,1), seasonal=list(order=c(1,1,0)))
m3
```

```
##
## Call:
## arima(x = Gbg, order = c(1, 0, 1), seasonal = list(order = c(1, 1, 0)))
##
## Coefficients:
##          ar1     ma1     sar1
##       0.8394  0.2941  -0.5097
## s.e.  0.0754  0.1469   0.1062
##
## sigma^2 estimated as 542563534:  log likelihood = -691.11,  aic = 1390.22
```

```
tsdiag(m3)
```

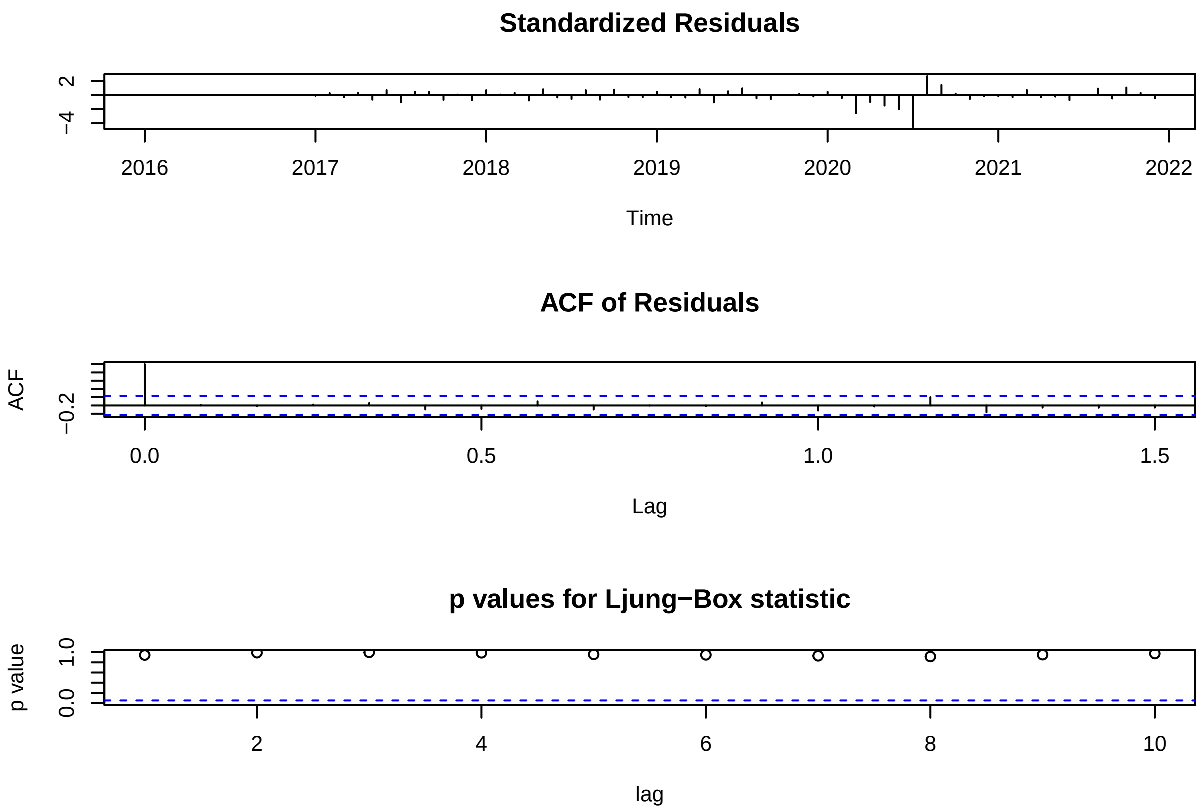


And eventually I tried to fit the series with one last model.

```
m4 <- arima(Gbg, order=c(1,0,3), seasonal=list(order=c(1,1,0)))
m4
```

```
##
## Call:
## arima(x = Gbg, order = c(1, 0, 3), seasonal = list(order = c(1, 1, 0)))
##
## Coefficients:
##          ar1     ma1      ma2      ma3     sar1
##       0.8661  0.2466  -0.0833  -0.0012  -0.5023
## s.e.  0.0903  0.1529   0.1590   0.1822   0.1079
##
## sigma^2 estimated as 5.41e+08:  log likelihood = -690.94,  aic = 1393.88
```

```
tsdiag(m4)
```



I then compared the AIC values.

```
AIC(m)
```

```
## [1] 1717.879
```

```
AIC(m1)
```

```
## [1] 1714.948
```

```
AIC(m2)
```

```
## [1] 1408.199
```

```
AIC(m3)
```

```
## [1] 1390.222
```

```
AIC(m4)
```

```
## [1] 1393.881
```

The lowest one was the one related to model 3. So I chose it to make some predictions.

Mathematical form of the model

Seasonal ARIMA (p,d,q)(P,D,Q)s:

$$\Theta_P(B^s)\theta_p(B)(1-B^s)^D(1-B)^d X_t = \Phi_Q(B^s)\phi_q(B)\epsilon_t$$

m3: (1,0,1)(1,1,0)12

$$\Theta_P(B^s)\theta_p(B)(1-B^s)^D X_t = \phi_q(B)\epsilon_t$$

$$(1-\Theta B^{12})(1-\theta B)(1-B^{12})X_t = (1+\phi B)\epsilon_t$$

$$(1-A_1 B^{12})(1-\alpha_1 B)(1-B^{12})X_t = (1+\beta_1 B)\epsilon_t$$

$$(X_{t-12}-AX_{t-13})(X_t-\alpha X_{t-1})(X_t-X_{t-12}) = \epsilon_t + \beta(\epsilon_{t-1})$$

$$\alpha = 0.8394$$

$$\beta = 0.2941$$

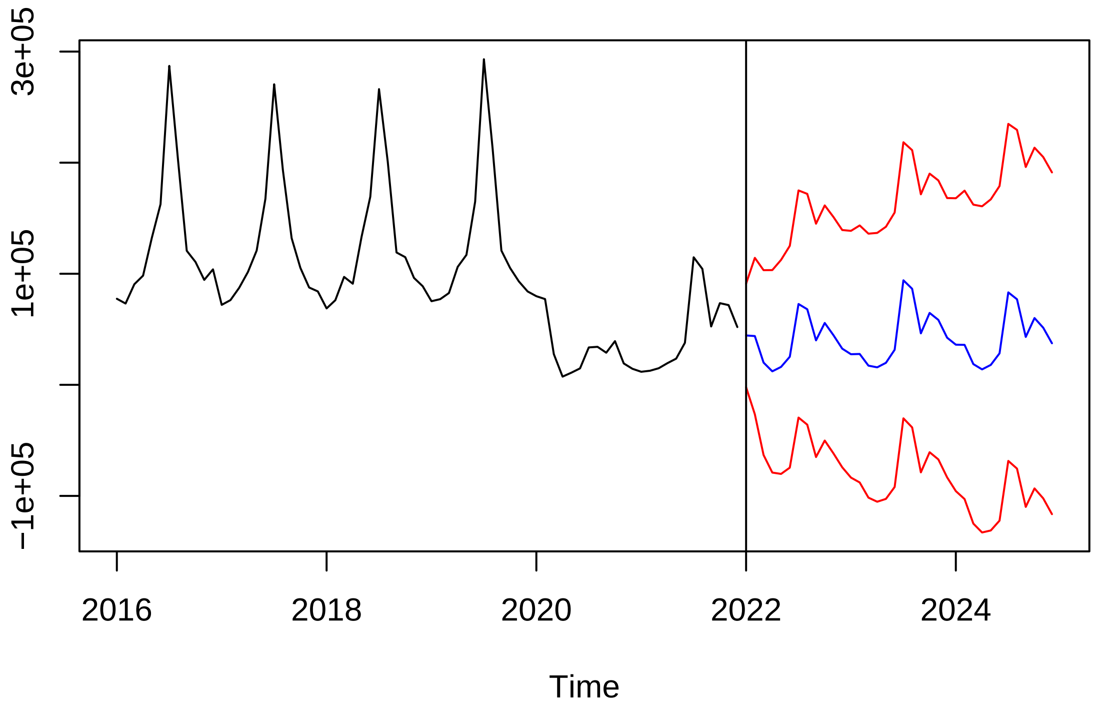
$$A = -0.5097$$

Prediction

I decided to make a prediction of 3 years ahead in the future for the years 2022, 2023 and 2024.

```r
predm3 <- predict(m3, n.ahead =12*3, se.fit = T)

ts.plot(cbind(Gbg, predm3$pred, predm3$pred-2*predm3$se,
               predm3$pred+2*predm3$se),
        col = c("black","blue", "red", "red"))
abline(v=2022)
```



Comment The prediction was affected by the declining trend the series had had in the year 2020, the year in which the Covid pandemic started. It seemed to replicate what happened during the last twelve observations, those belonging to the year 2021, as if the prediction of the following years strongly depended on the last observations. The seasonality was preserved and the values fluctuated inside the same range in all the three years. Also the trend was preserved, since the blue line slightly declined. It seemed quite a good prediction as the lines were not flattened after a certain point (the coefficients were not the same after a certain number of future observations).

Cross validation

To check the consistency of the prediction and the goodness of the model, I decided to perform a cross validation by using a window of values from January 2016 to December 2020 and use the same model to predict the observations of the next 12 months (the year 2021).

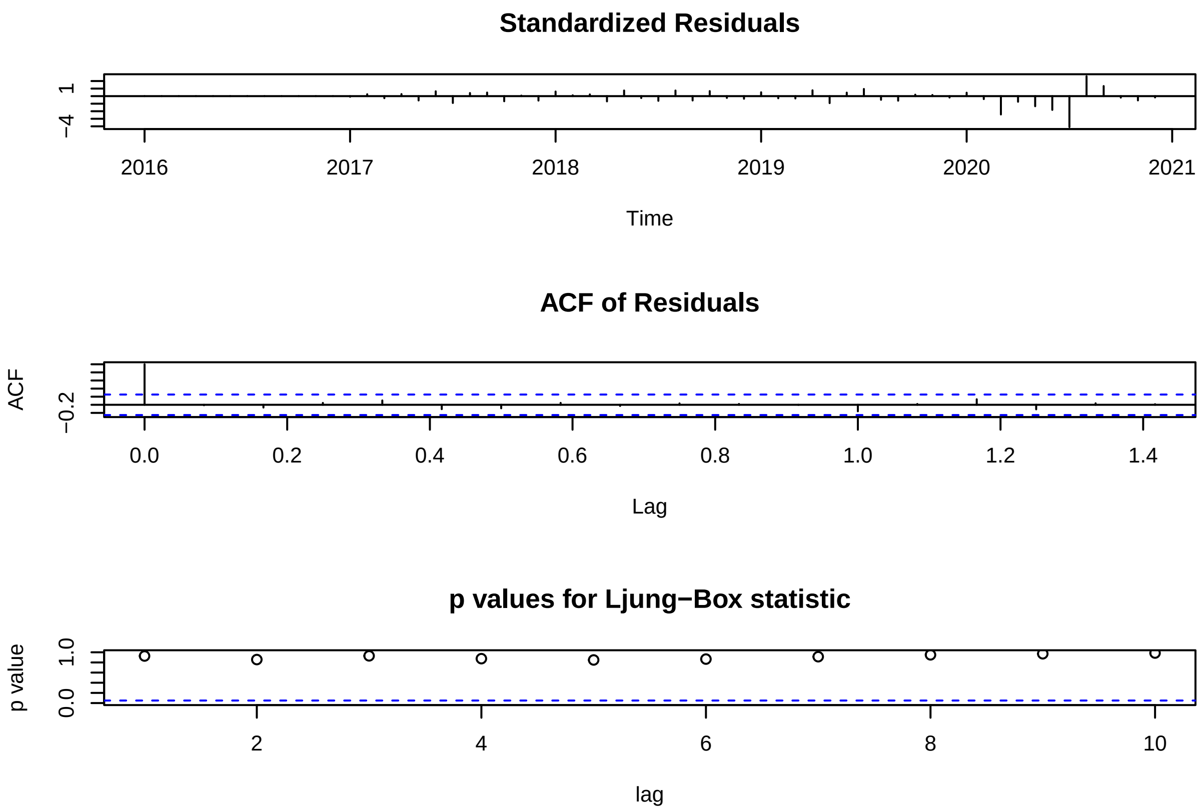
```
w <- window(Gbg, start=c(2016,1), end=c(2020,12))
w
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 2016  77492  73194  90535  98287 132276 162644 287123 202326 120691 110612
## 2017  71979  76252  87388 101625 120896 167531 270583 193326 132083 105310
## 2018  68781  76200  97168  91018 133200 169352 266238 200726 119146 114910
## 2019  75274  77118  82621 106166 116984 164971 293129 212396 120862 105174
## 2020  79794  77194  27638   7373  10890  14810  33675  34222  28906  39340
##         Nov    Dec
## 2016  94451 103969
## 2017  87633  84091
## 2018  96239  88755
## 2019  93076  84066
## 2020  19242  14427
```

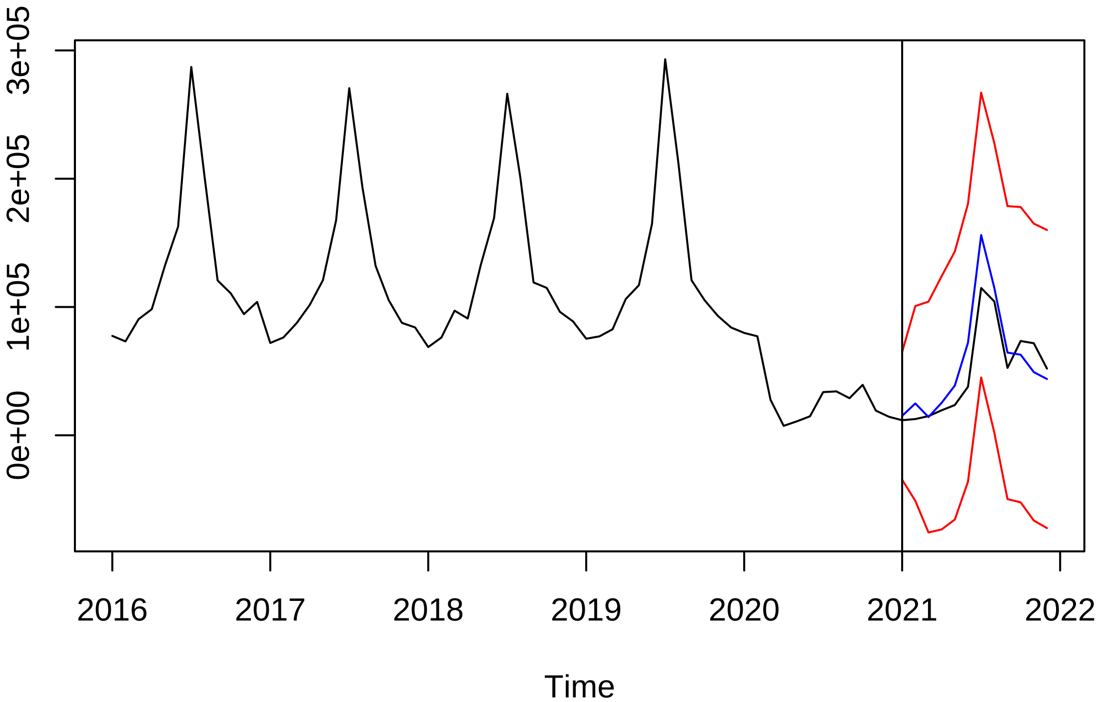
```
mw <- arima(w, order=c(1,0,1), seasonal=list(order=c(1,1,0)))
mw
```

```
##
## Call:
## arima(x = w, order = c(1, 0, 1), seasonal = list(order = c(1, 1, 0)))
##
## Coefficients:
##          ar1     ma1     sar1
##       0.8418  0.3087  -0.5570
## s.e.  0.0864  0.1590   0.1575
##
## sigma^2 estimated as 620127819:  log likelihood = -557.06,  aic = 1122.12
```

```
tsdiag(mw)
```




```
predw <- predict(mw, n.ahead = 12, se.fit=T)
ts.plot(cbind(Gbg, predw$pred, predw$pred-2*predw$se,
              predw$pred+2*predw$se),
        col = c("black","blue", "red", "red"))
abline(v=2021)
```



The model I chose was quite good as it was consistent in predicting (blue line) what happened in 2021 (black line), though being not completely precise.