Spatio-Temporal Data Analysis Project

2020-04-26

Patterns in foreign sims connected to OpenWiFi-Milan

Author: Bernardi Riccardo - 864018

Contents

1 Introduction & Motivation

The dataset that I've chosen is about the presence of foreign smartphone's sims to the OpenWifi of the Municipality of Milan. This data is open and available on the website data.gov.it. The reasons why I would like to go further with this project is that I strongly believe that are present seasonalities that can be interesting to be analysed but also can be more interesting to relate the outliers to some events that happened in the past with a certain mediatic relevance. In practice I would like to both analyse trend and seasonalities to know in which months there are more foreign people and if the trend is increasing in time and both search for outlier peaks to be related to important happenings in the Milan city. Finally I would like to forecast the possible presences in the new year in the city of Milan.

2 The Data

The dataset comes from the open data provided by all the municipalities of Milan. This repository is available at dati.gov.it. From this repository I selected the data going from January of 2018 to October of the 2019.

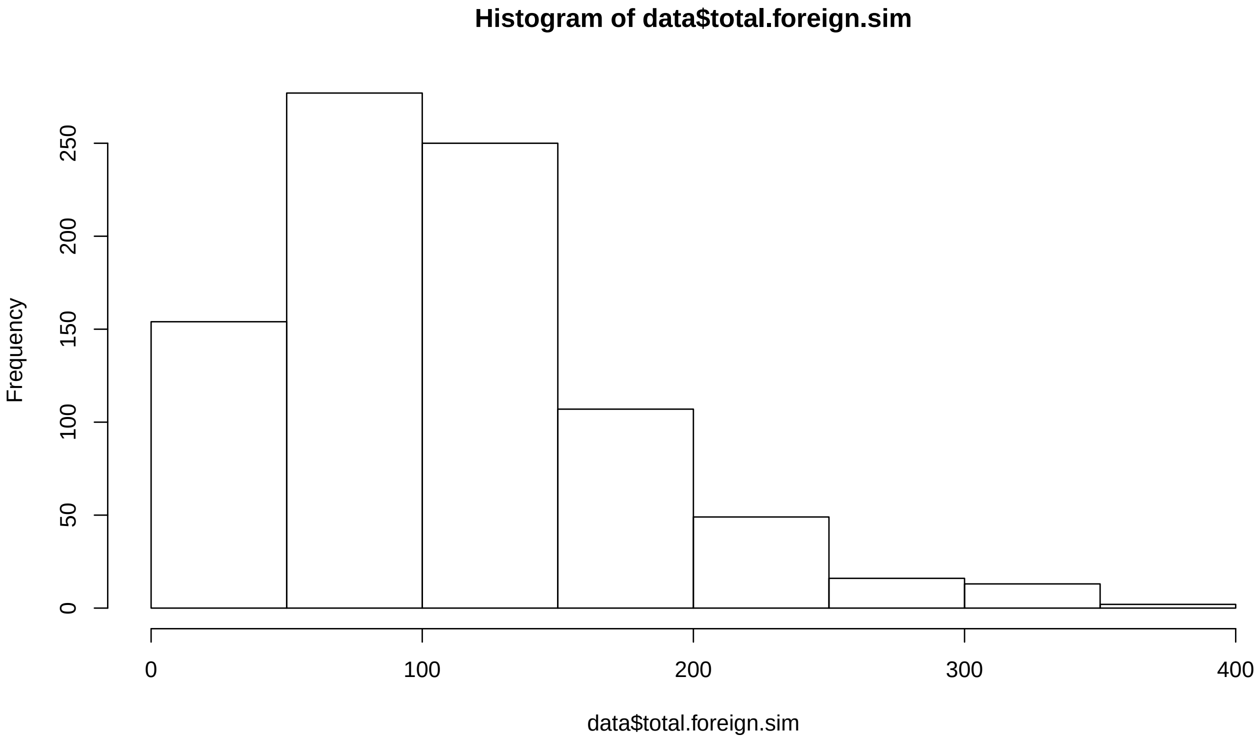
Characteristics of the DataSet:

- the dataset contains 2 columns "Date, Number_of_Foreign_Sims"
- has 658 rows
- Dates goes from from 01/01/18 to 30/10/19 (~2 years)
- the datasets have no NA
- no lacking days
- the "Number_of_Foreign_Sims" is a discrete variable about total number of foreign sims in a certain Date connected to the OpenWifi of Milan

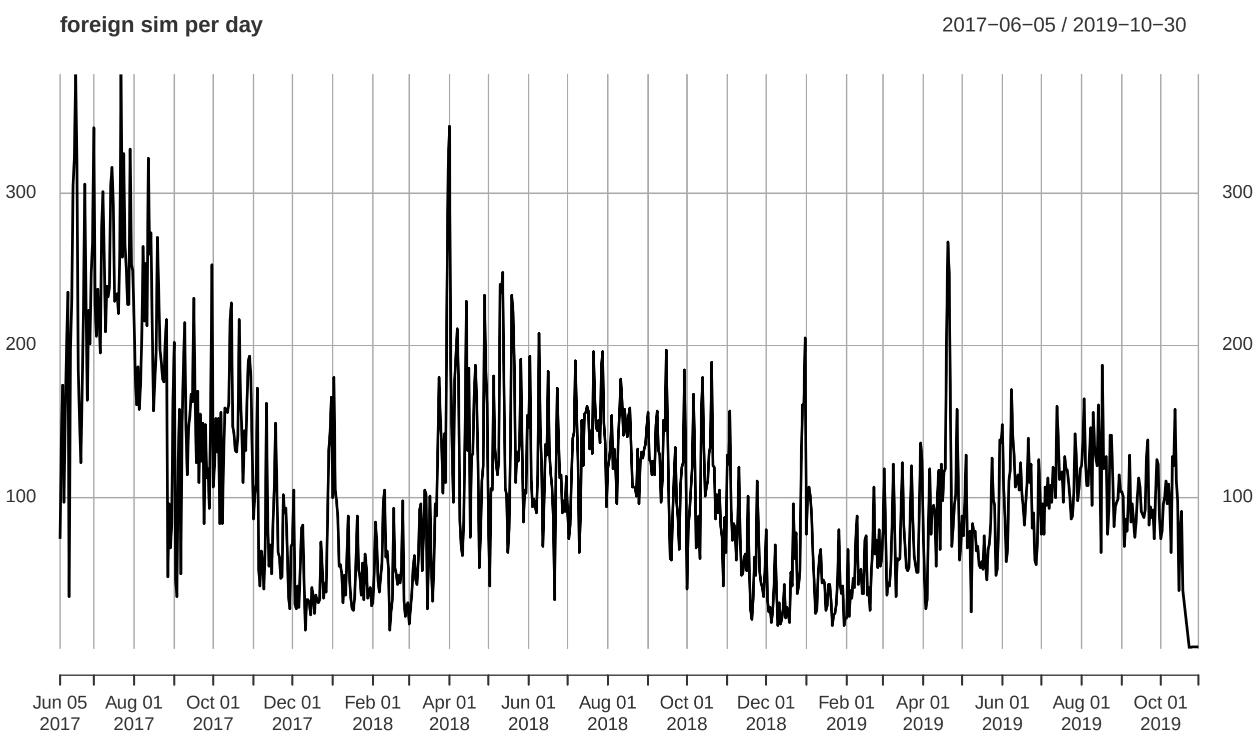
3 Exploration of the Data

```
## [1] "minimum, lower-hinge, median, upper-hinge, maximum)"
```

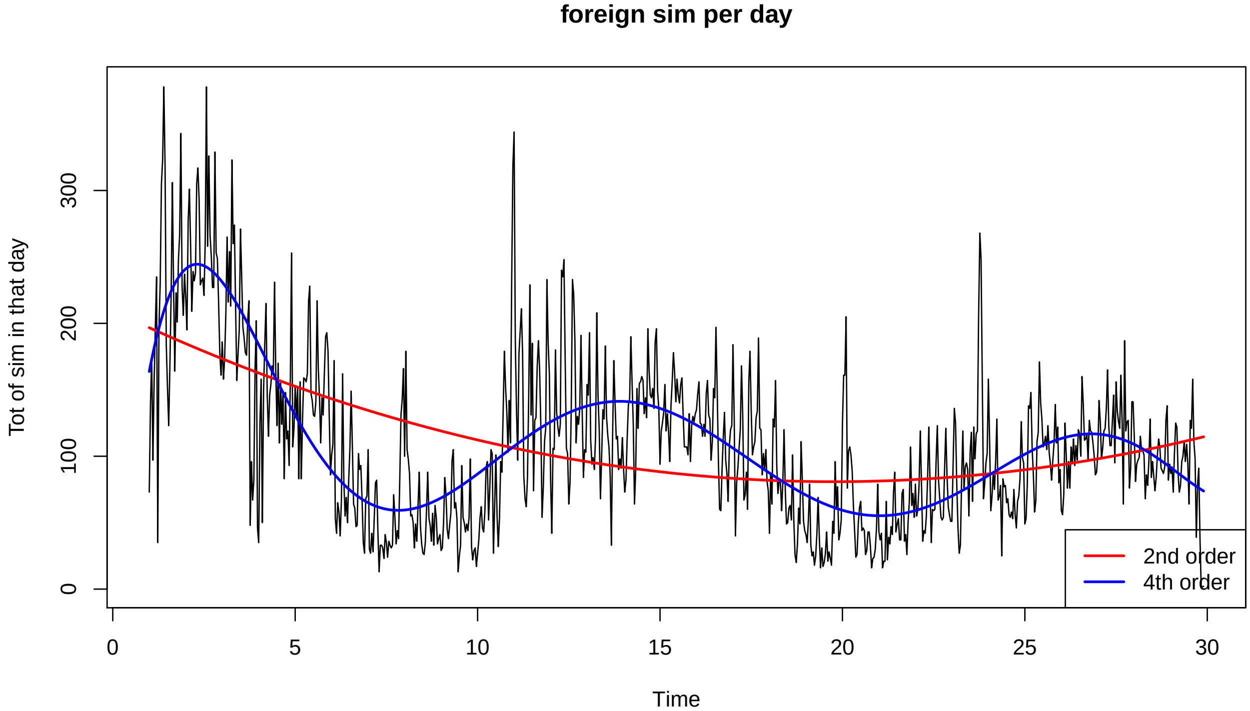
```
## [1]   1.0  61.5 101.0 141.0 378.0
```



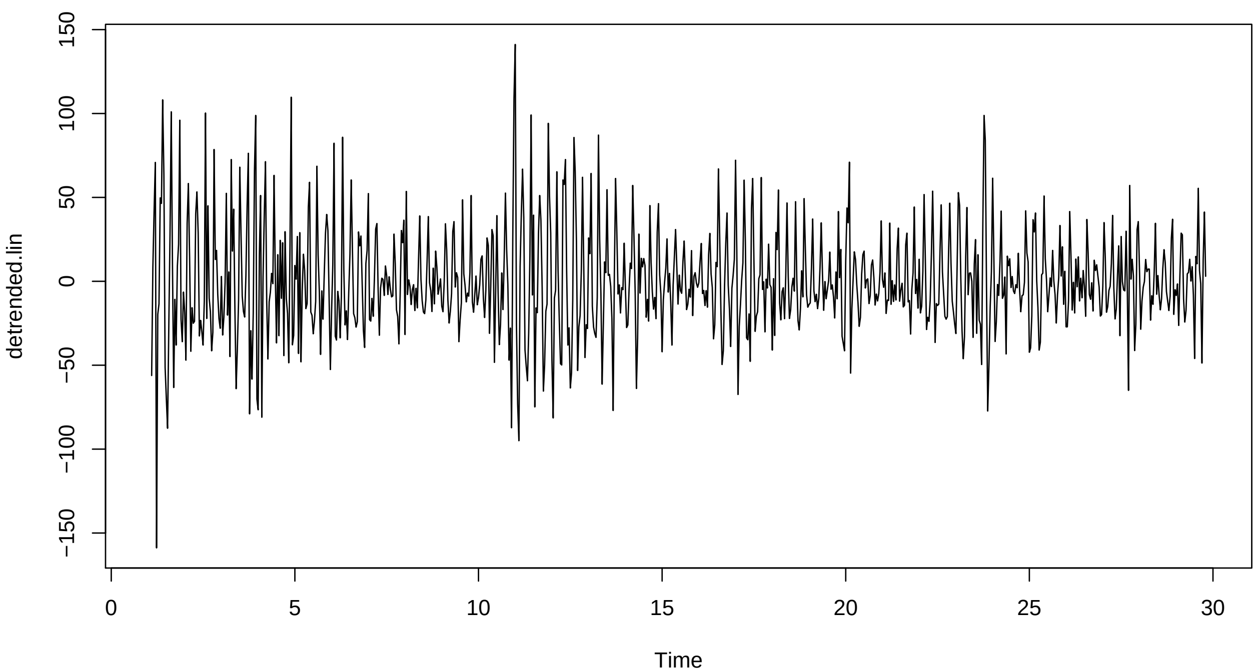
We loaded the dataset from the various datasets aggregating into only one dataset with 655 rows representing 2 years of data gathered. Starting from 01.01.2018 to 30.10.2019. Data is here:



4 Trend recognition

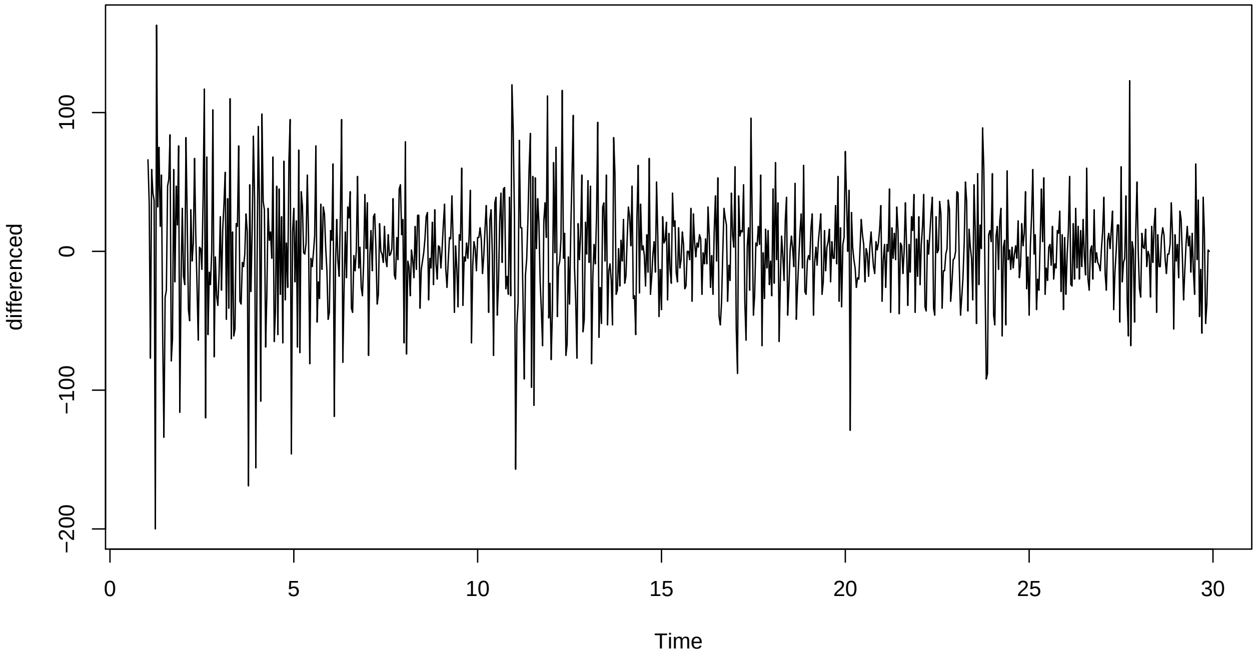


4.1 Detrending using LM



5 Removing seasonality

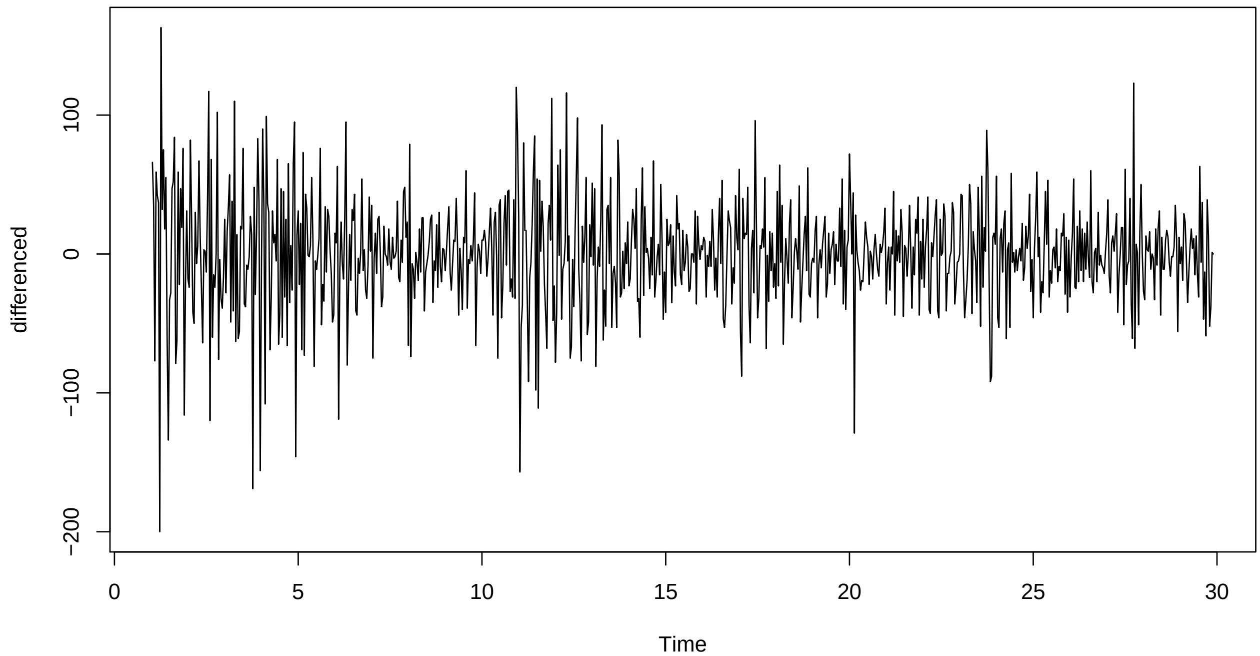
A good idea is to differenciate before decomposing. With the multiplicative model



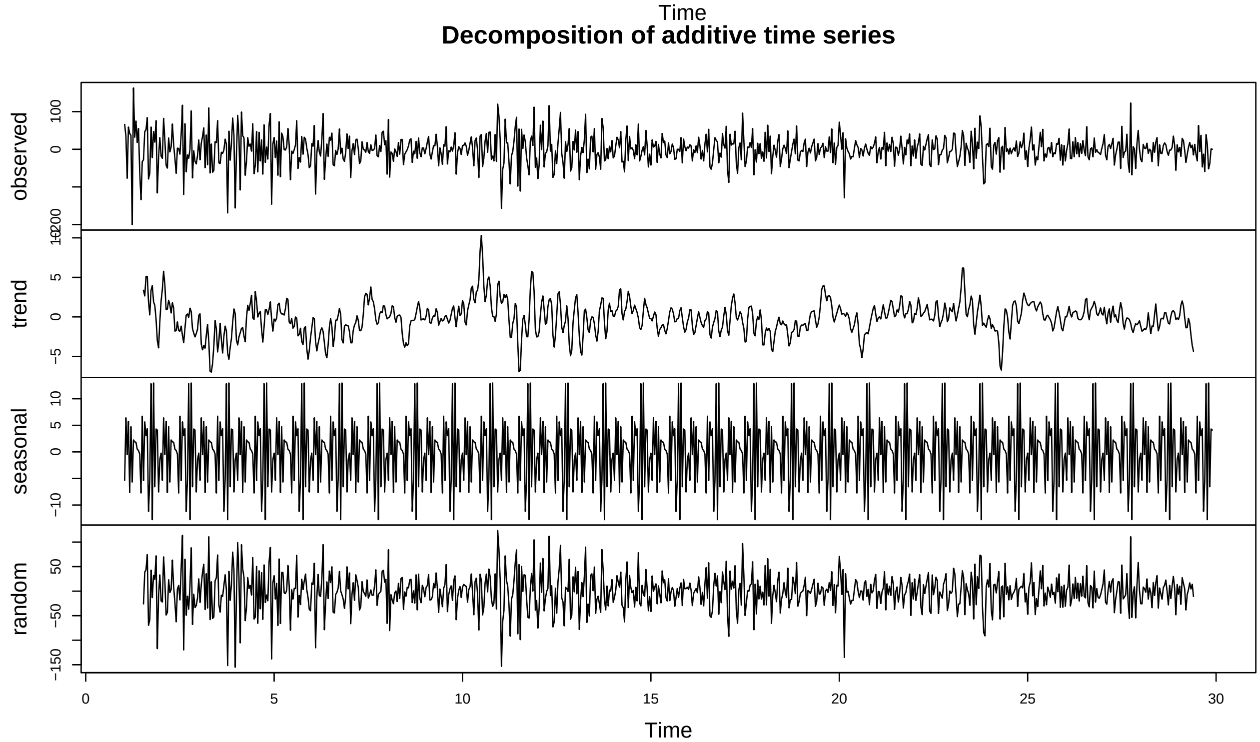

```
## [1] -200

## [1] 163

##
##  Box-Pierce test
##
## data:  decomposed$random
## X-squared = NA, df = 5, p-value = NA

##
##  Box-Ljung test
##
## data:  decomposed$random
## X-squared = NA, df = 5, p-value = NA
```

6 The additive model doesn't work for us

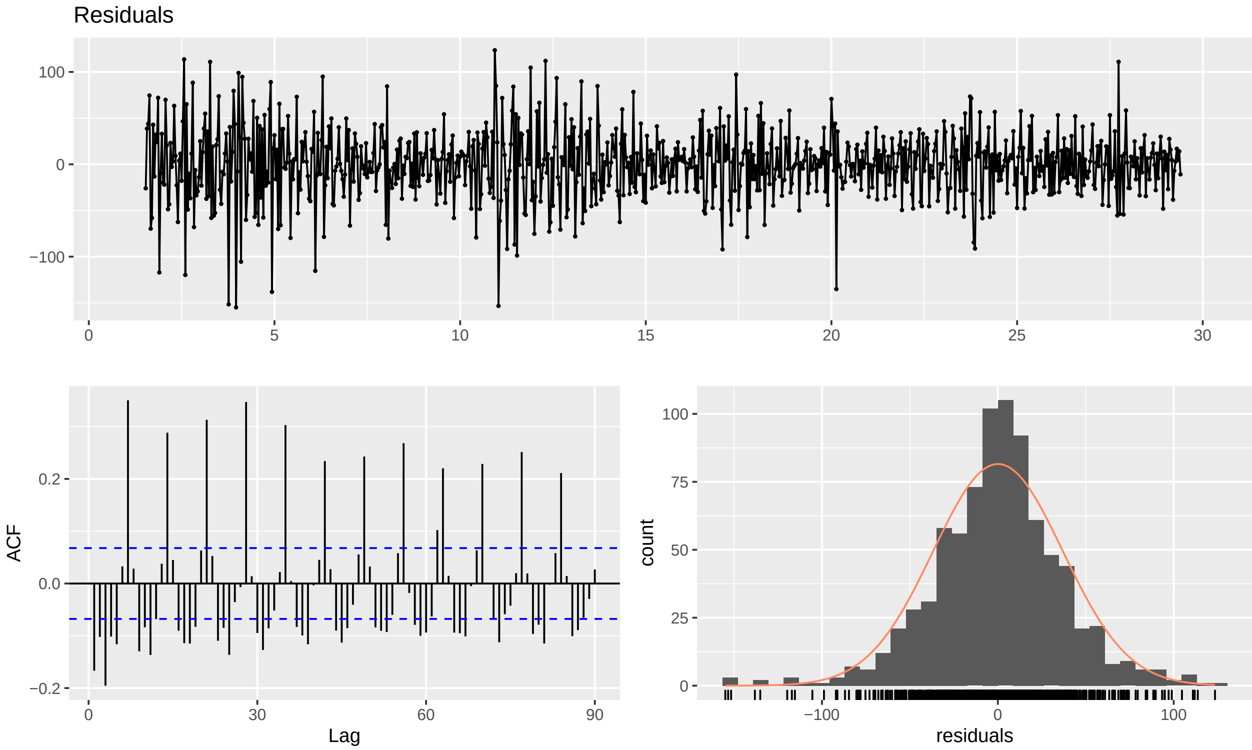
With the additive model This model doesn't work at all



differenced
100
0
−100
−200
0
5
10
15
20
25
30
Time



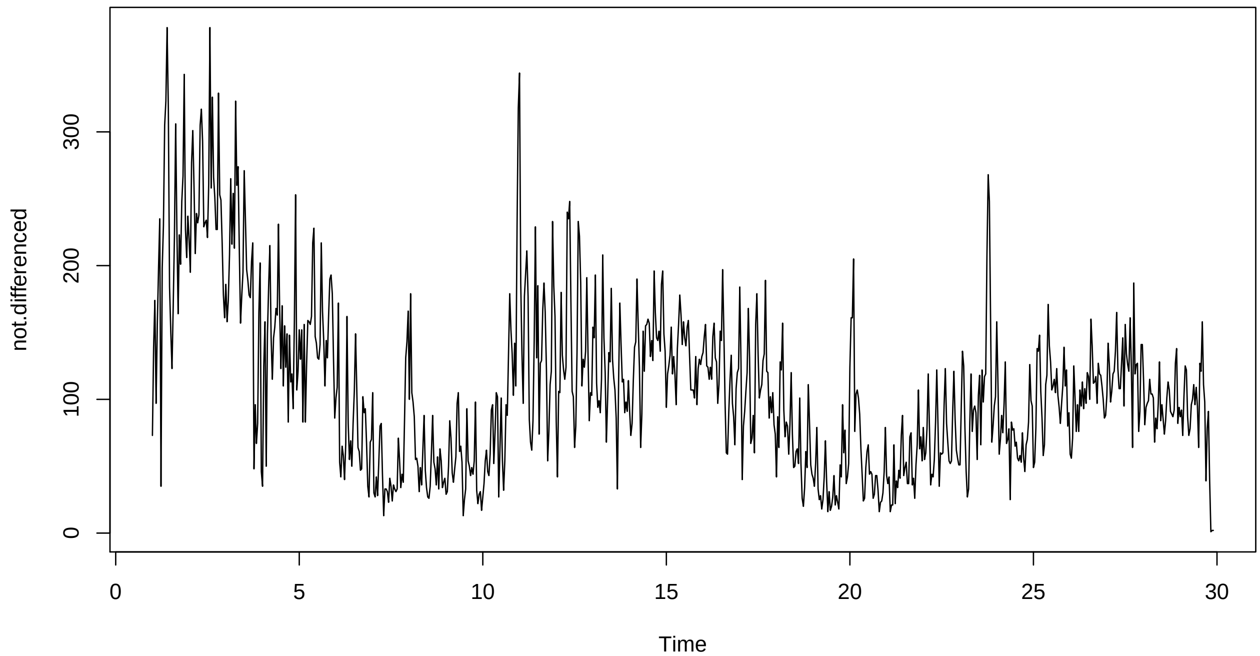
Decomposition of additive time series
observed
trend
seasonal
random
Time



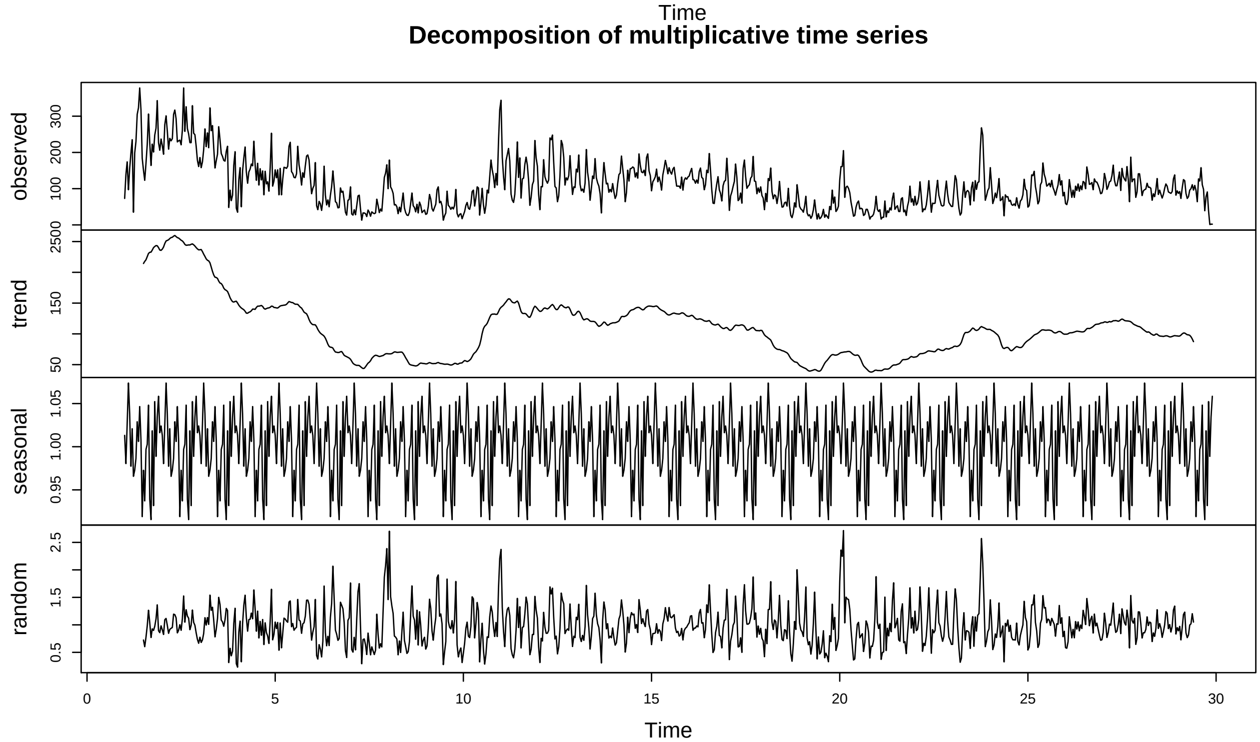

```
##
##  Box-Pierce test
##
## data:  decomposed$random
## X-squared = 83.893, df = 5, p-value < 2.2e-16
```

```
##
##  Box-Ljung test
##
## data:  decomposed$random
## X-squared = 84.367, df = 5, p-value < 2.2e-16
```

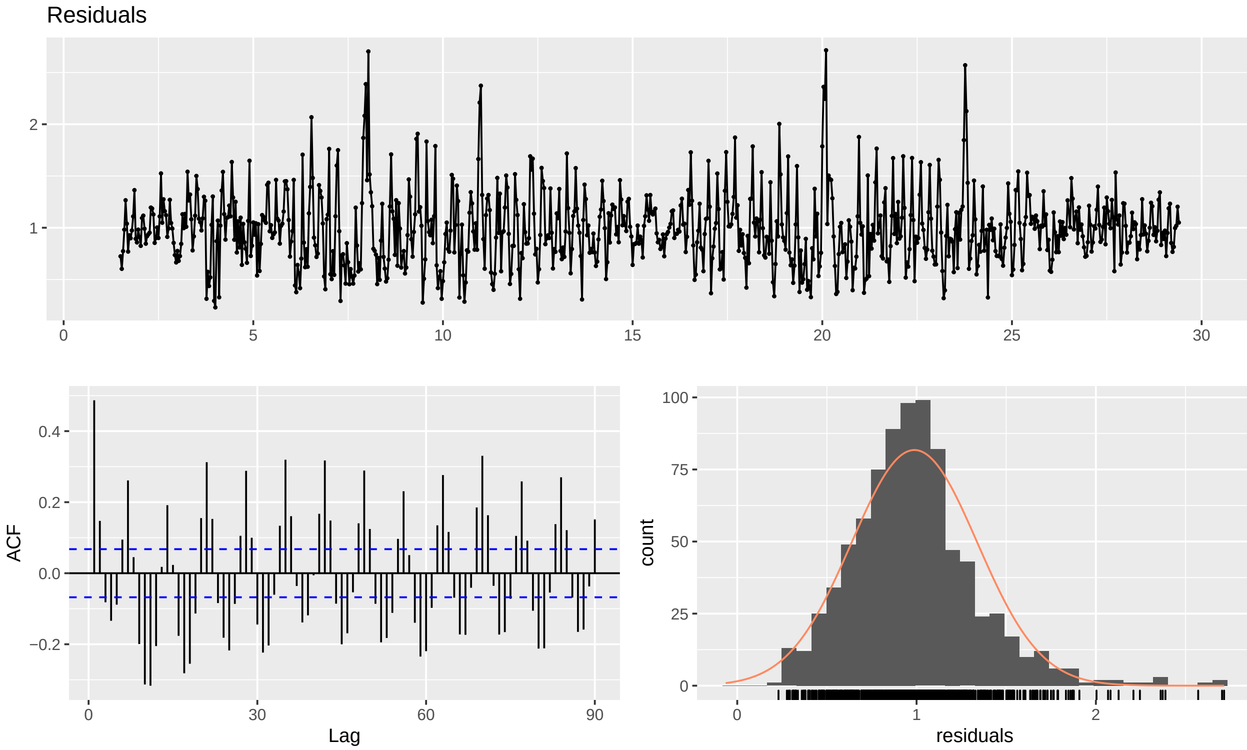
Without the first differentiation the result will have been much worse:



not.differenced
0
100
200
300
0
5
10
15
20
25
30
Time



Decomposition of multiplicative time series
observed
trend
seasonal
random
Time

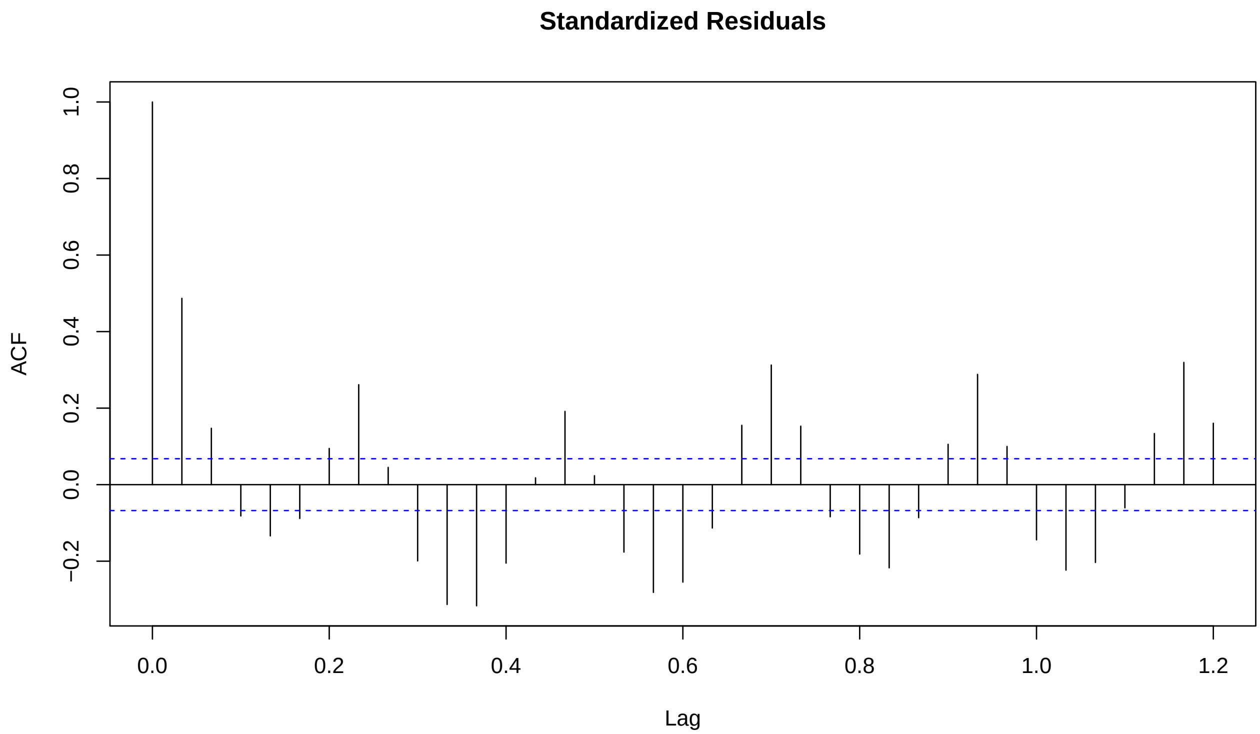


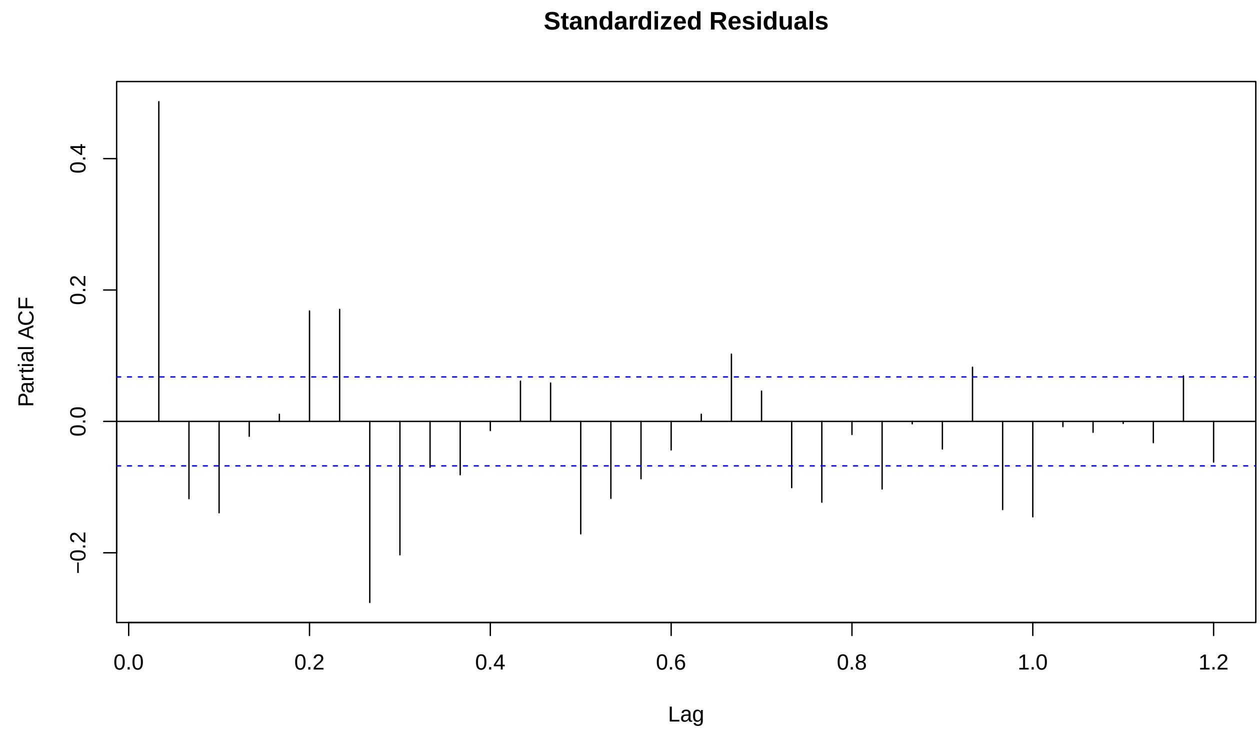

```
##
##  Box-Pierce test
##
## data:  decomposed$random
## X-squared = 244, df = 5, p-value < 2.2e-16
```

```
##
##  Box-Ljung test
##
## data:  decomposed$random
## X-squared = 244.99, df = 5, p-value < 2.2e-16
```

Every 7 lags the peak recurs

7 Check Residuals

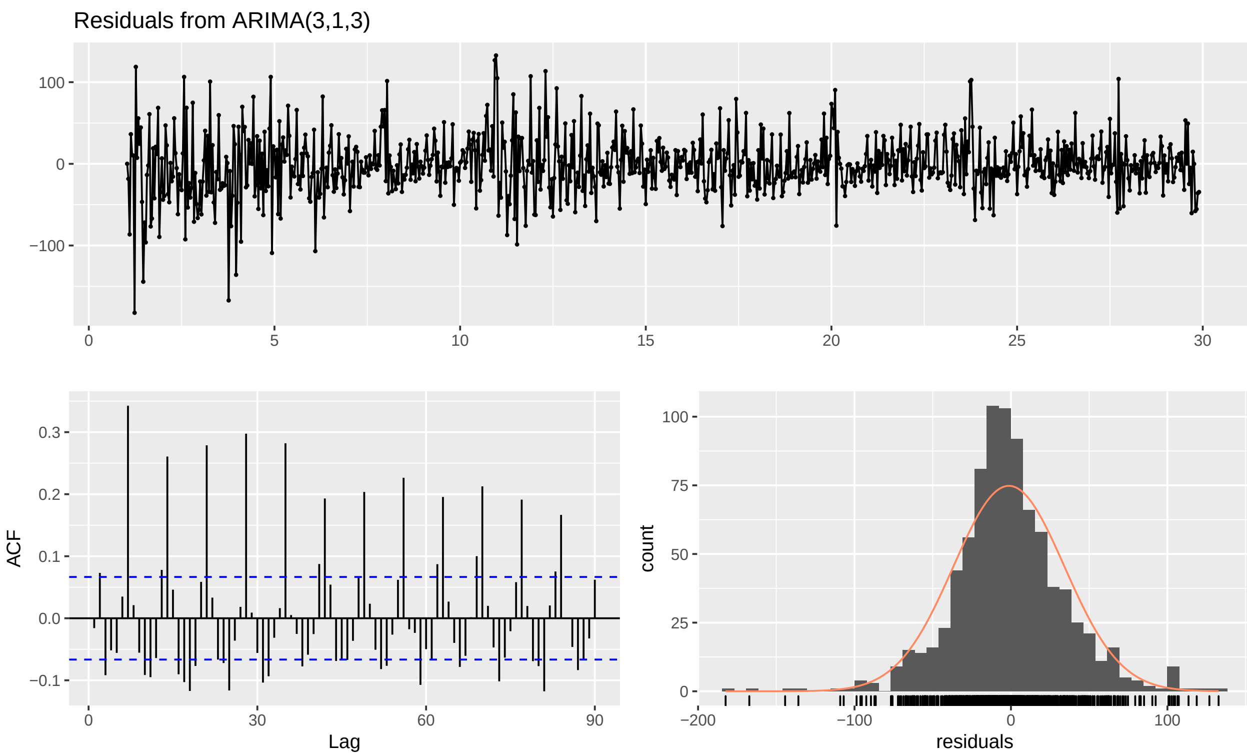




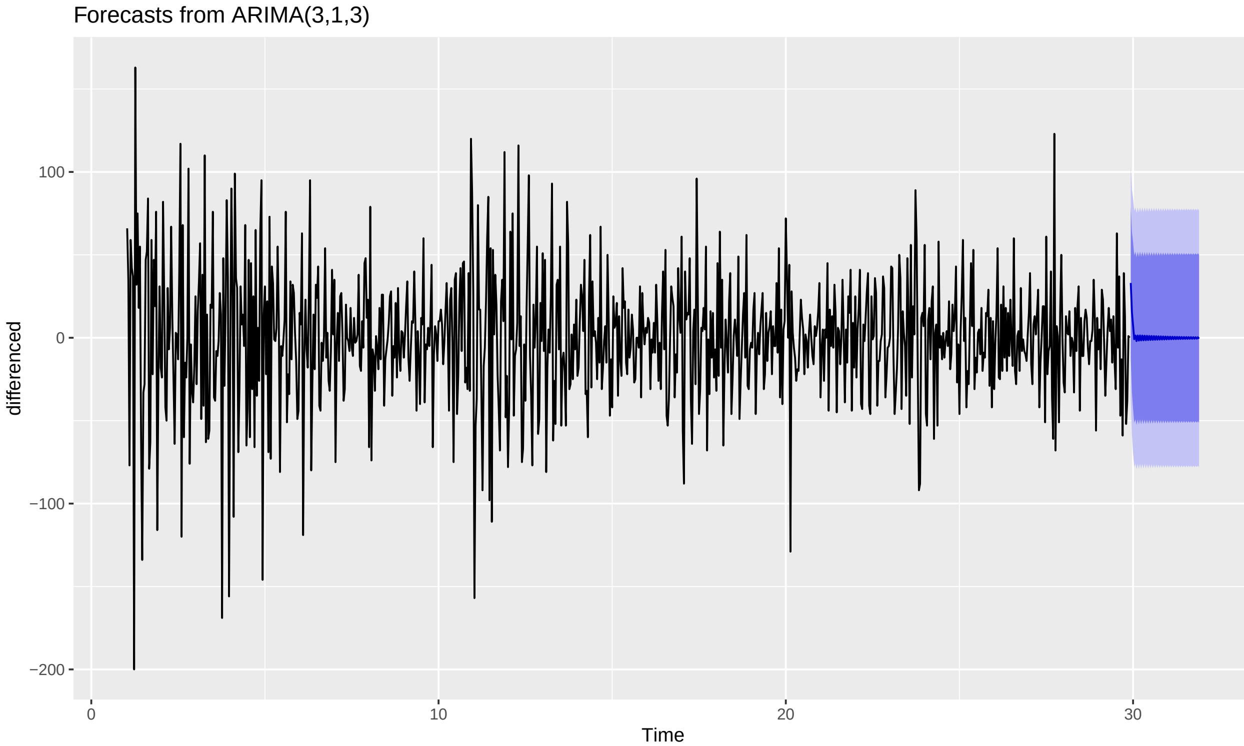
8 Arima

```
## Series: differenced
```

```
## ARIMA(3,1,3)
##
## Coefficients:
##           ar1     ar2      ar3      ma1      ma2     ma3
##       -0.4107  0.4209  -0.1268  -0.9264  -0.9235  0.8499
## s.e.   0.3868  0.2064   0.0512   0.4041   0.7835  0.3810
##
## sigma^2 estimated as 1273:  log likelihood=-4326.94
## AIC=8667.87   AICc=8668   BIC=8701.22
##
## Training set error measures:
##                     ME     RMSE      MAE MPE MAPE      MASE        ACF1
## Training set -1.314353 35.54053 25.84148 NaN  Inf 0.5948003 -0.01590064
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,3)
## Q* = 702.82, df = 54, p-value < 2.2e-16
##
## Model df: 6.   Total lags used: 60
```



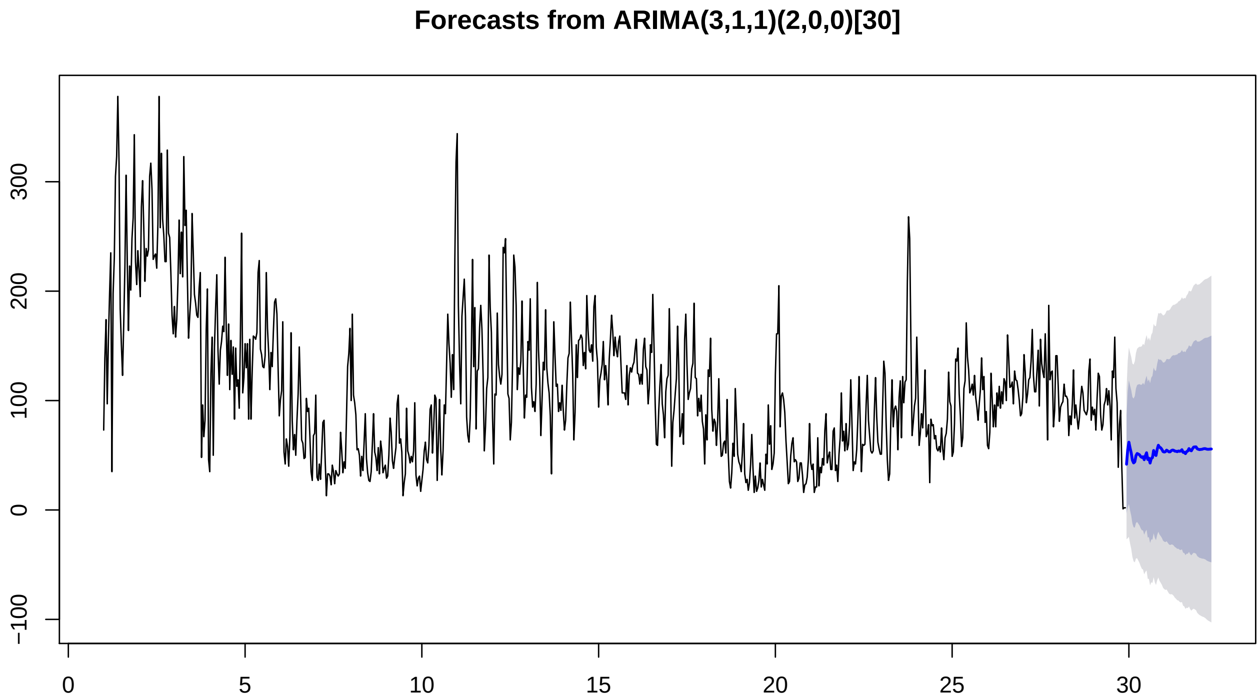
9 Auto Arima

```
##
## Fitting models using approximations to speed things up...
##
## ARIMA(2,1,2)(1,0,1)[30] with drift         : 8610.806
## ARIMA(0,1,0)            with drift         : 8834.014
## ARIMA(1,1,0)(1,0,0)[30] with drift         : 8730.36
## ARIMA(0,1,1)(0,0,1)[30] with drift         : 8800.342
## ARIMA(0,1,0)                               : 8827.253
## ARIMA(2,1,2)(0,0,1)[30] with drift         : Inf
## ARIMA(2,1,2)(1,0,0)[30] with drift         : 8610.141
## ARIMA(2,1,2)            with drift         : 8671.879
## ARIMA(2,1,2)(2,0,0)[30] with drift         : Inf
## ARIMA(2,1,2)(2,0,1)[30] with drift         : Inf
## ARIMA(1,1,2)(1,0,0)[30] with drift         : 8608.827
## ARIMA(1,1,2)            with drift         : 8678.032
## ARIMA(1,1,2)(2,0,0)[30] with drift         : 8579.411
## ARIMA(1,1,2)(2,0,1)[30] with drift         : 8583.577
## ARIMA(1,1,2)(1,0,1)[30] with drift         : 8609.229
## ARIMA(0,1,2)(2,0,0)[30] with drift         : Inf
## ARIMA(1,1,1)(2,0,0)[30] with drift         : 8575.506
## ARIMA(1,1,1)(1,0,0)[30] with drift         : 8606.677
## ARIMA(1,1,1)(2,0,1)[30] with drift         : 8580.196
```

```
##  ARIMA(1,1,1)(1,0,1)[30] with drift         : 8607.332
##  ARIMA(0,1,1)(2,0,0)[30] with drift         : 8665.059
##  ARIMA(1,1,0)(2,0,0)[30] with drift         : 8689.61
##  ARIMA(2,1,1)(2,0,0)[30] with drift         : 8567.804
##  ARIMA(2,1,1)(1,0,0)[30] with drift         : 8603.453
##  ARIMA(2,1,1)(2,0,1)[30] with drift         : 8574.386
##  ARIMA(2,1,1)(1,0,1)[30] with drift         : 8604.228
##  ARIMA(2,1,0)(2,0,0)[30] with drift         : 8688.443
##  ARIMA(3,1,1)(2,0,0)[30] with drift         : 8540.098
##  ARIMA(3,1,1)(1,0,0)[30] with drift         : 8566.621
##  ARIMA(3,1,1)(2,0,1)[30] with drift         : 8546.123
##  ARIMA(3,1,1)(1,0,1)[30] with drift         : 8571.574
##  ARIMA(3,1,0)(2,0,0)[30] with drift         : 8638.097
##  ARIMA(4,1,1)(2,0,0)[30] with drift         : 8538.185
##  ARIMA(4,1,1)(1,0,0)[30] with drift         : 8561.636
##  ARIMA(4,1,1)(2,0,1)[30] with drift         : 8541.546
##  ARIMA(4,1,1)(1,0,1)[30] with drift         : 8566.49
##  ARIMA(4,1,0)(2,0,0)[30] with drift         : 8593.136
##  ARIMA(5,1,1)(2,0,0)[30] with drift         : 8527.192
##  ARIMA(5,1,1)(1,0,0)[30] with drift         : 8548.369
##  ARIMA(5,1,1)(2,0,1)[30] with drift         : 8532.352
##  ARIMA(5,1,1)(1,0,1)[30] with drift         : 8553.149
##  ARIMA(5,1,0)(2,0,0)[30] with drift         : 8537.623
##  ARIMA(5,1,2)(2,0,0)[30] with drift         : 8499.702
##  ARIMA(5,1,2)(1,0,0)[30] with drift         : Inf
##  ARIMA(5,1,2)(2,0,1)[30] with drift         : 8506.464
##  ARIMA(5,1,2)(1,0,1)[30] with drift         : Inf
##  ARIMA(4,1,2)(2,0,0)[30] with drift         : 8546.42
##  ARIMA(5,1,3)(2,0,0)[30] with drift         : 8505.967
##  ARIMA(4,1,3)(2,0,0)[30] with drift         : 8543.06
##  ARIMA(5,1,2)(2,0,0)[30]                    : 8493.095
##  ARIMA(5,1,2)(1,0,0)[30]                    : Inf
##  ARIMA(5,1,2)(2,0,1)[30]                    : 8499.858
##  ARIMA(5,1,2)(1,0,1)[30]                    : Inf
##  ARIMA(4,1,2)(2,0,0)[30]                    : 8540.416
##  ARIMA(5,1,1)(2,0,0)[30]                    : 8520.844
##  ARIMA(5,1,3)(2,0,0)[30]                    : 8499.324
##  ARIMA(4,1,1)(2,0,0)[30]                    : 8532.057
##  ARIMA(4,1,3)(2,0,0)[30]                    : 8536.662
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(5,1,2)(2,0,0)[30]                    : Inf
##  ARIMA(5,1,3)(2,0,0)[30]                    : Inf
##  ARIMA(5,1,2)(2,0,0)[30] with drift         : Inf
```

```
##  ARIMA(5,1,2)(2,0,1)[30]                    : Inf
##  ARIMA(5,1,3)(2,0,0)[30] with drift         : Inf
##  ARIMA(5,1,2)(2,0,1)[30] with drift         : Inf
##  ARIMA(5,1,1)(2,0,0)[30]                    : 8644.43
##
##  Best model: ARIMA(5,1,1)(2,0,0)[30]
```

```
## Series: data.ts
## ARIMA(5,1,1)(2,0,0)[30]
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ma1     sar1     sar2
##       -0.0582  -0.2057  -0.3041  -0.2182  -0.2583  -0.3268  -0.0639  -0.0228
## s.e.   0.0683   0.0385   0.0338   0.0382   0.0391   0.0663   0.0383   0.0386
##
## sigma^2 estimated as 1176:  log likelihood=-4291.77
## AIC=8601.54   AICc=8601.75   BIC=8644.43
```

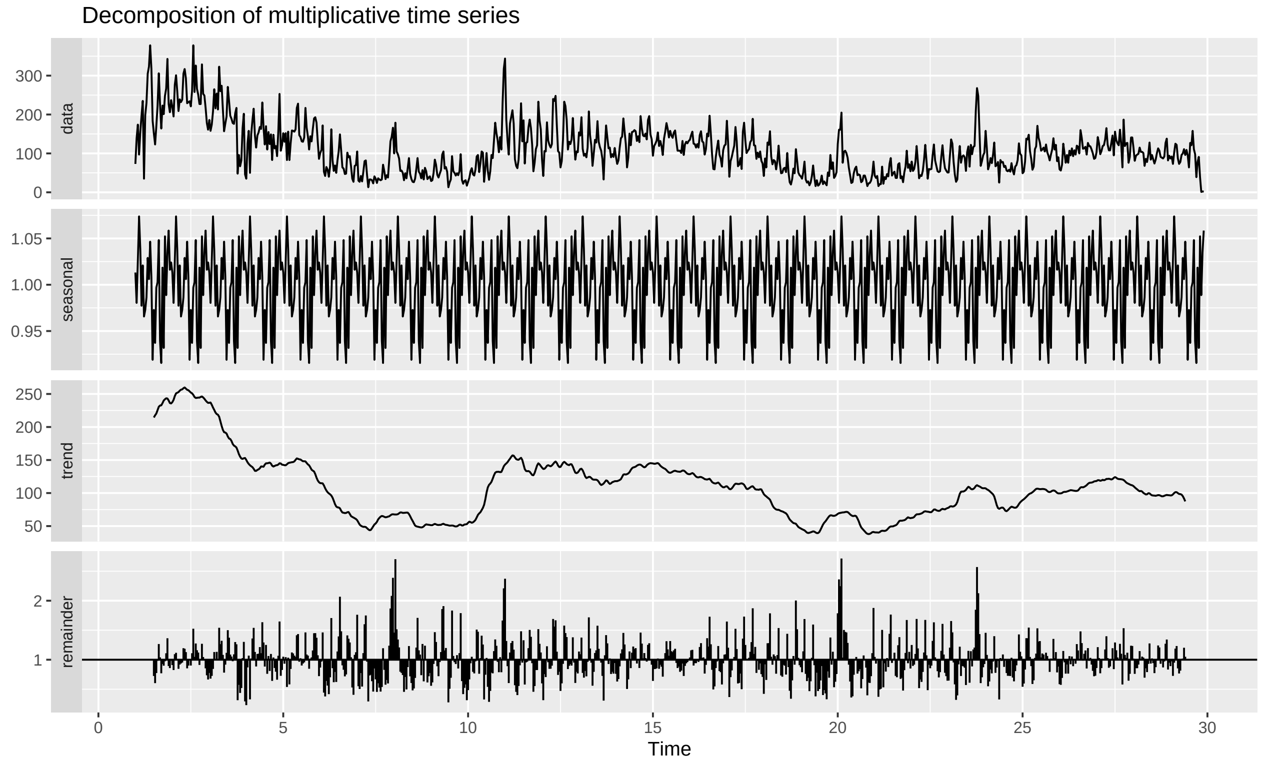


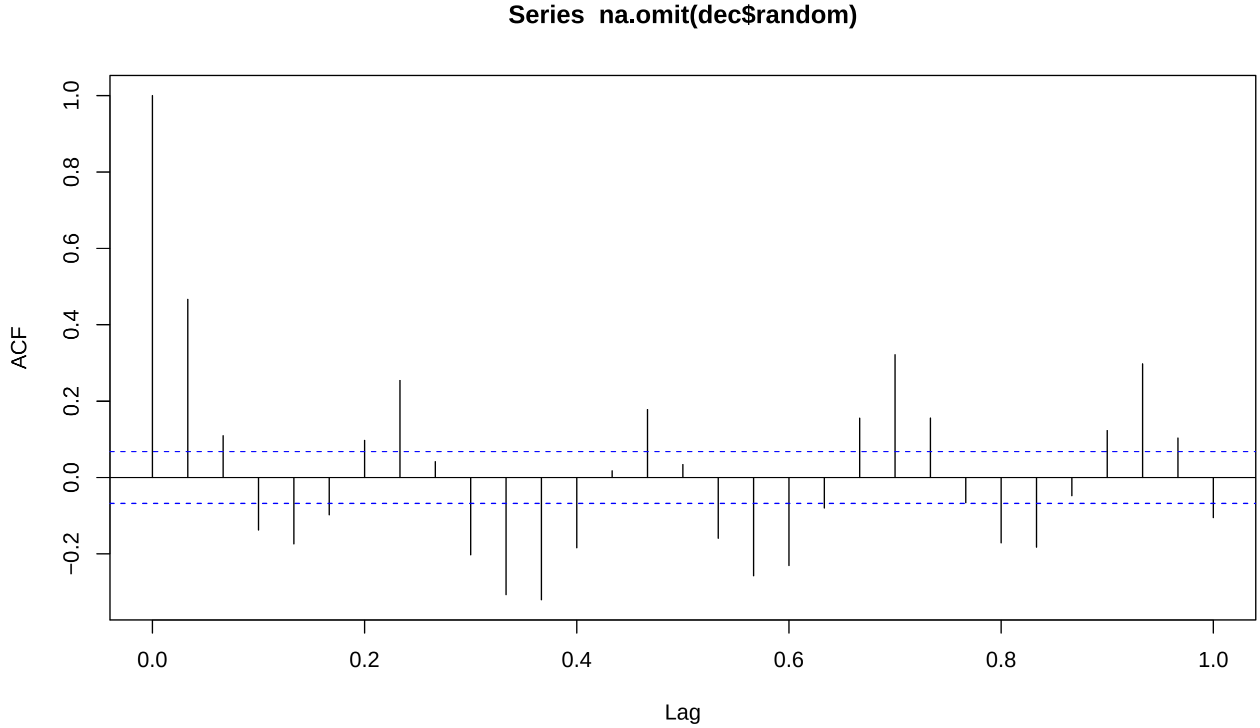
The plot is not good but AIC and BIC are very high, we should try with a multi seasonal decomposition

```
## [1] 30
```

10 Searching for multi seasonalities

without differentiation residuals looks pretty bad

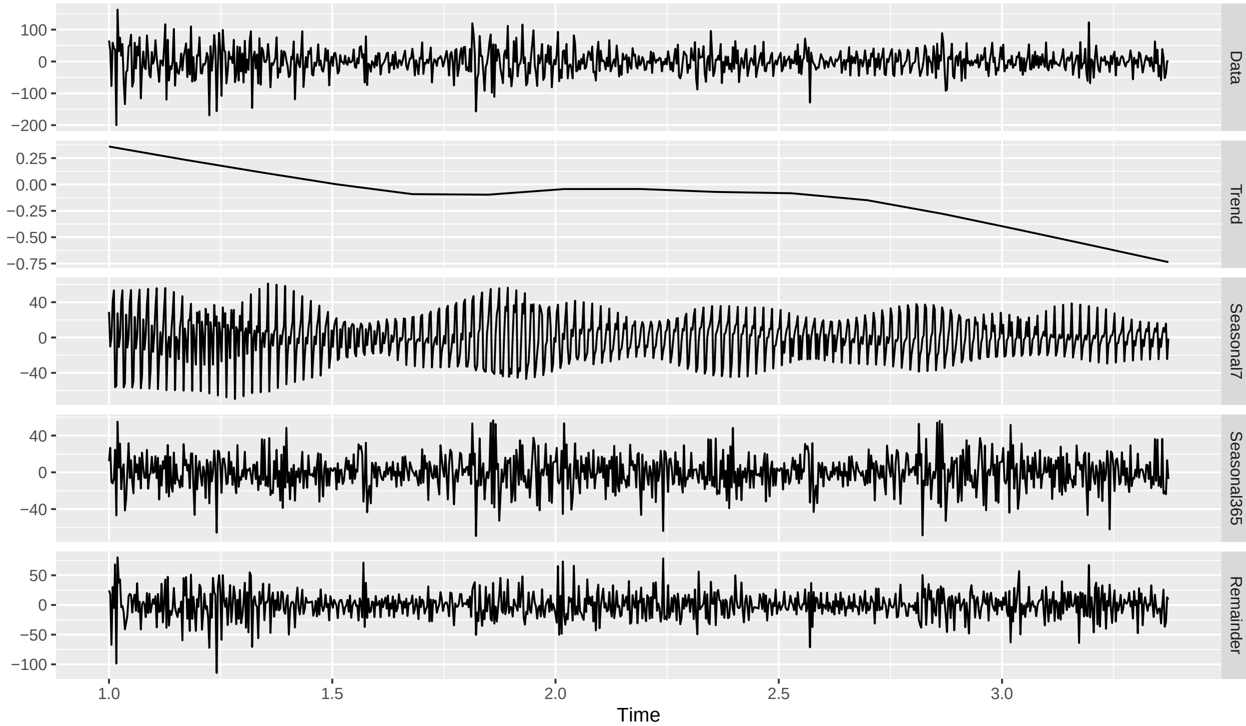


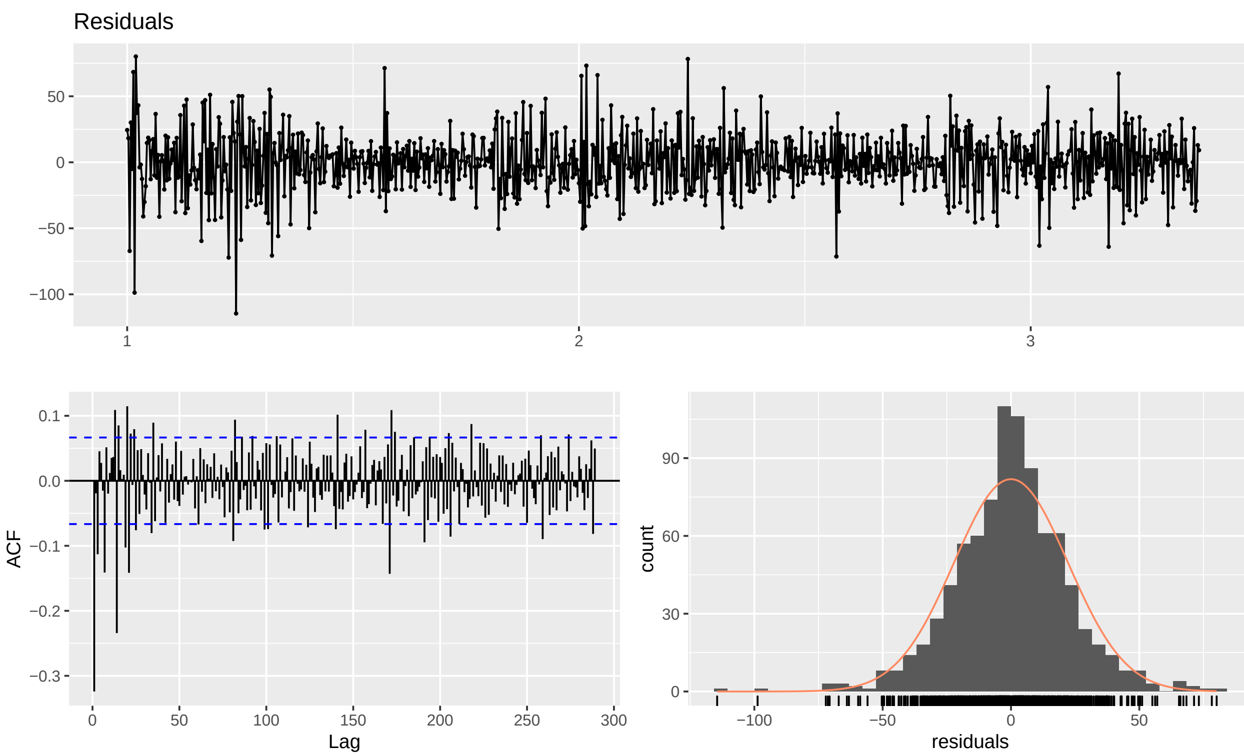


trying with differentiation and a multiplicative model:

Looks better than before but we can still see every 5(*7) a seasonality/trend left. 5*7 is about a month, probably there is a monthly seasonality

11 Transforming into msts



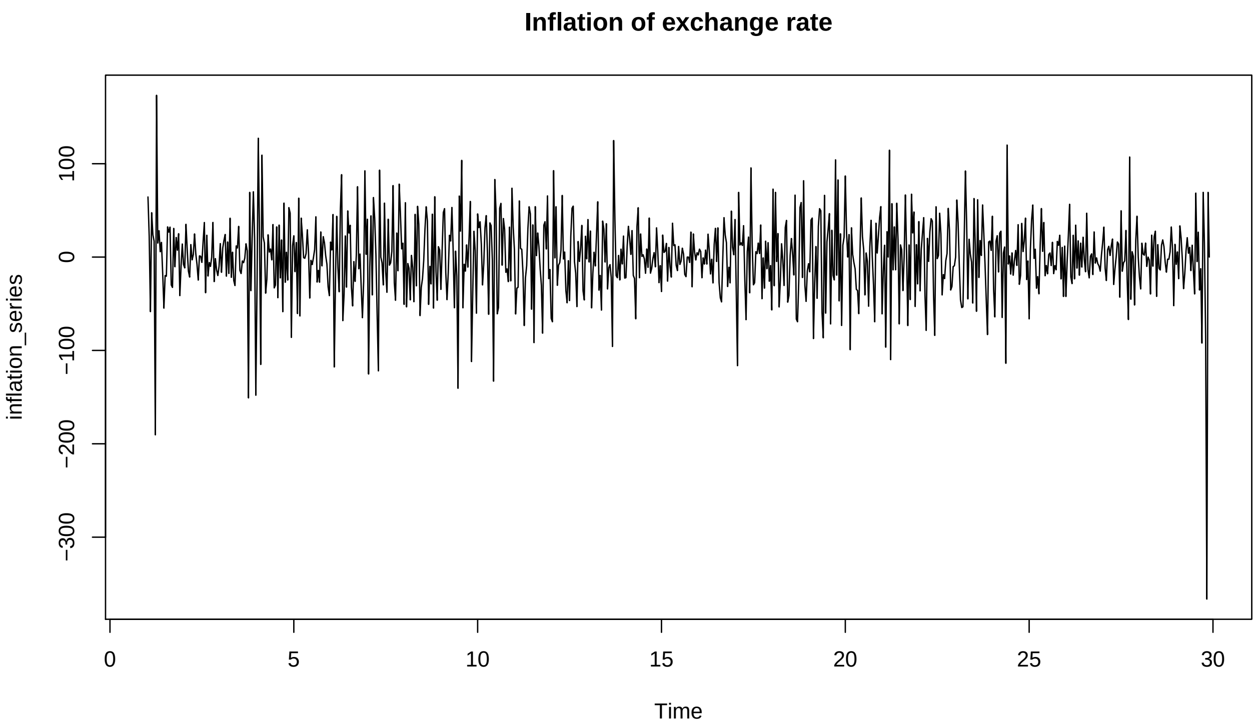



```
##
##  Box-Pierce test
##
```

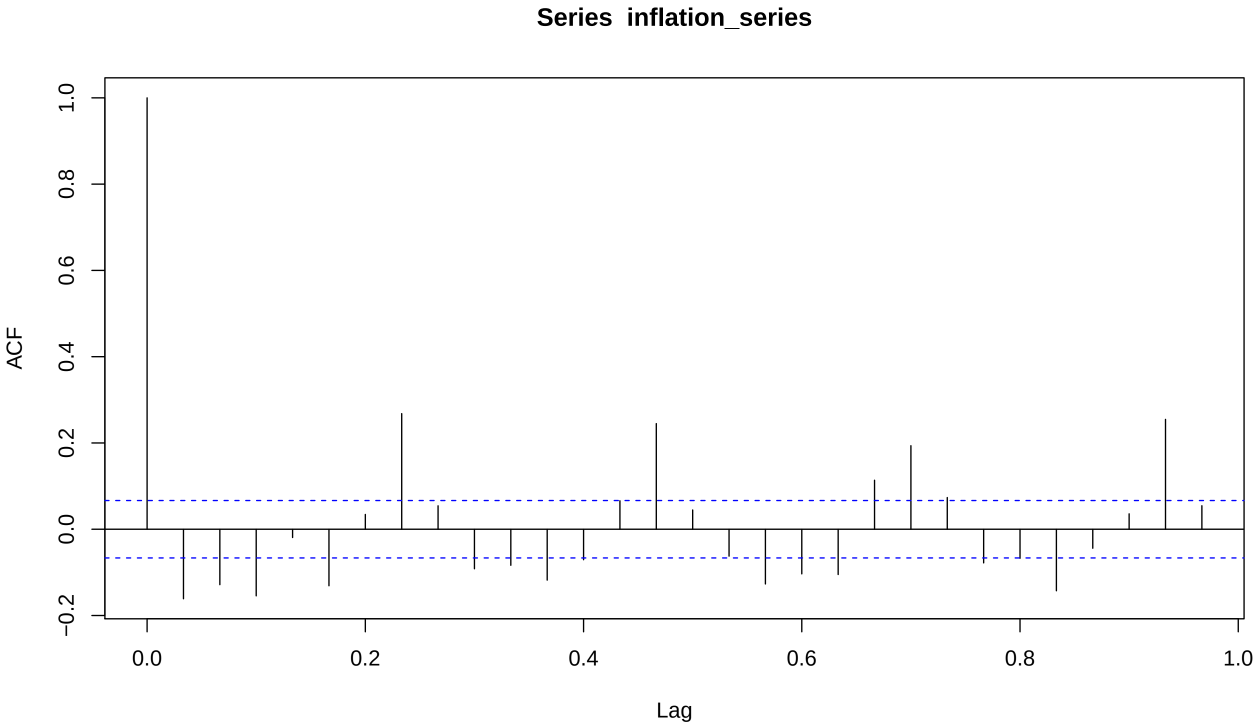
```
## data:  remainder(decomposed)
## X-squared = 104.9, df = 5, p-value < 2.2e-16
```

```
##
##  Box-Ljung test
##
## data:  remainder(decomposed)
## X-squared = 105.3, df = 5, p-value < 2.2e-16
```

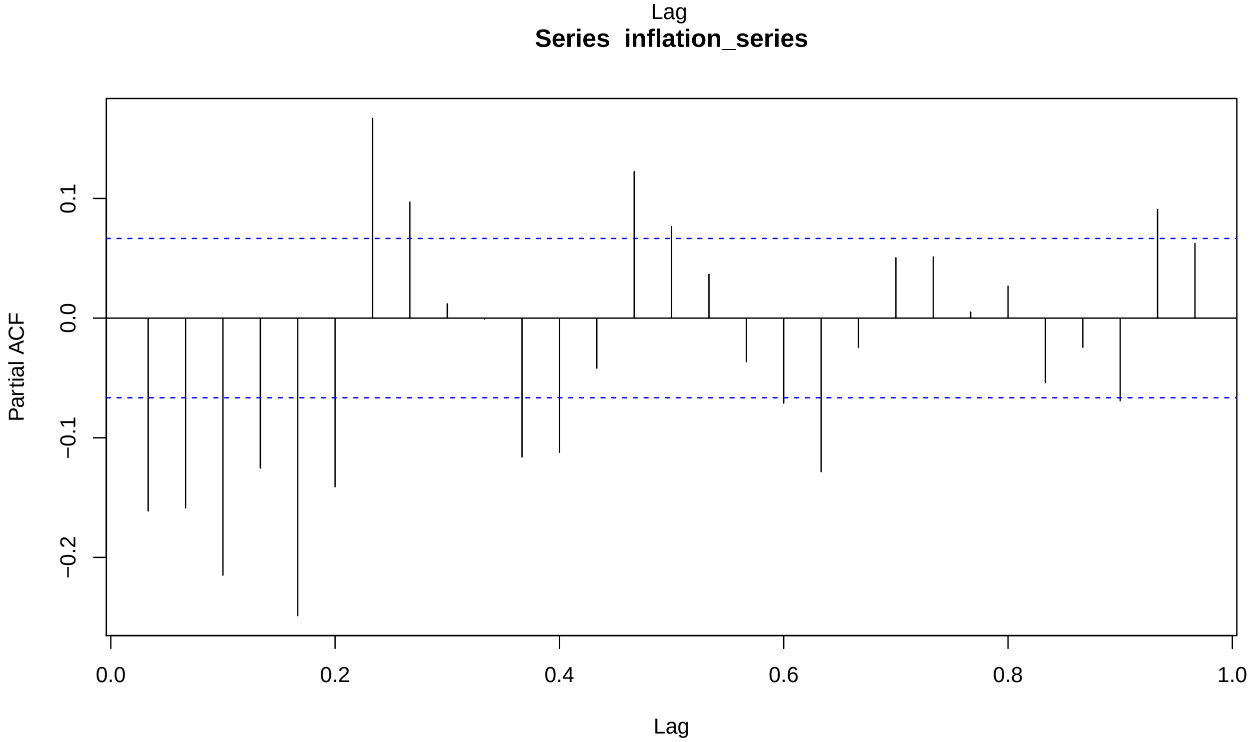
12 Garch



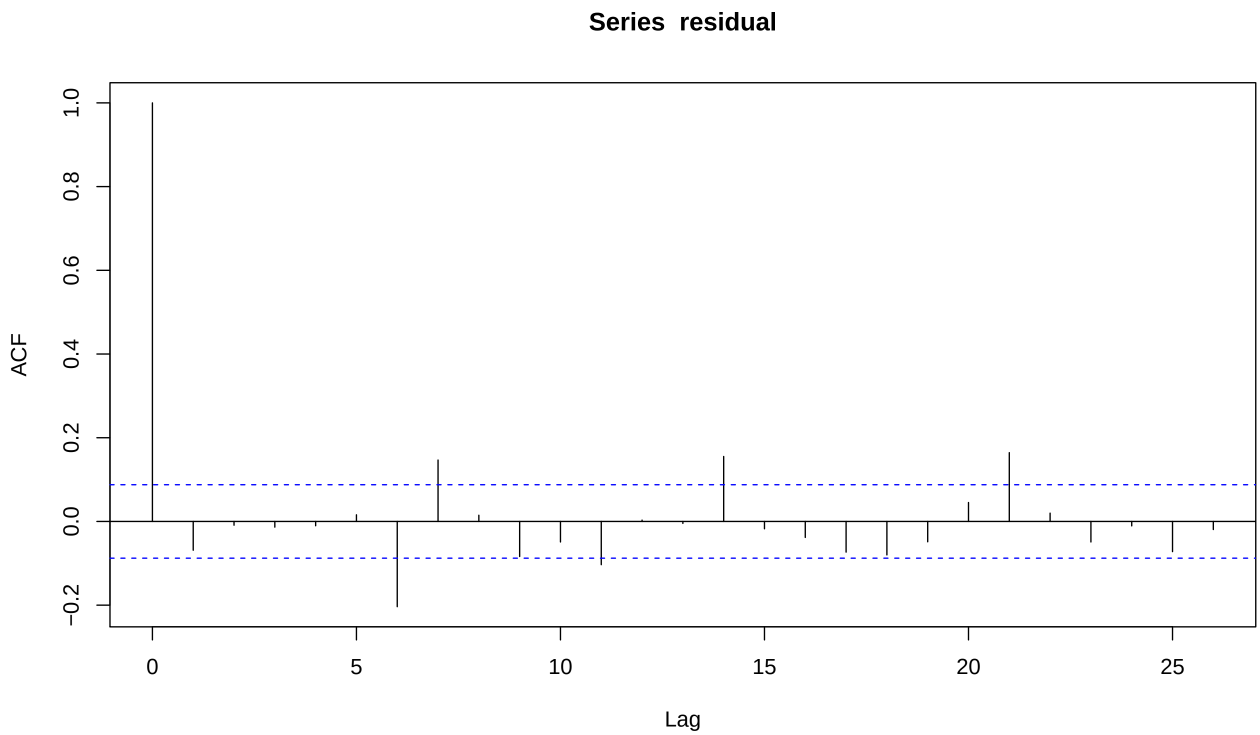

```
##       Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -366.3562  -22.2343    1.2903   -0.4149   23.6858  173.2919
```

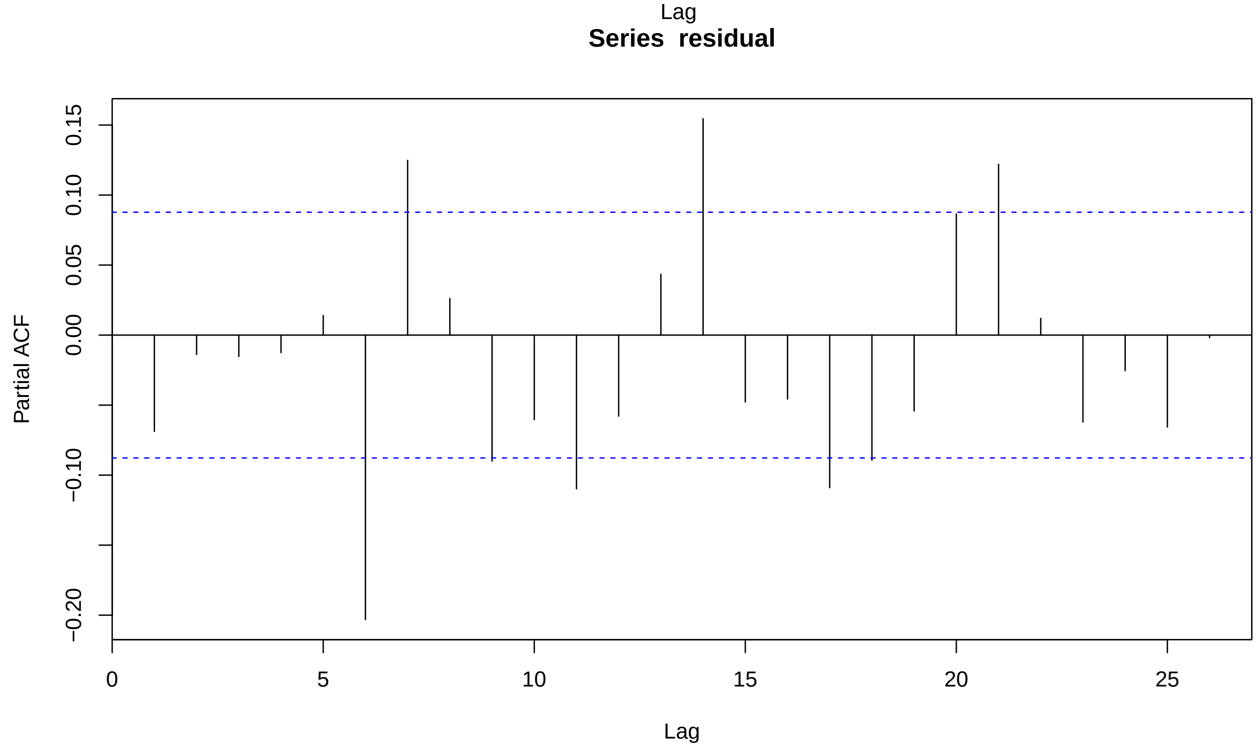


Series inflation_series
ACF
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Lag



Series inflation_series
Partial ACF
0.1
0.0
−0.1
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Lag



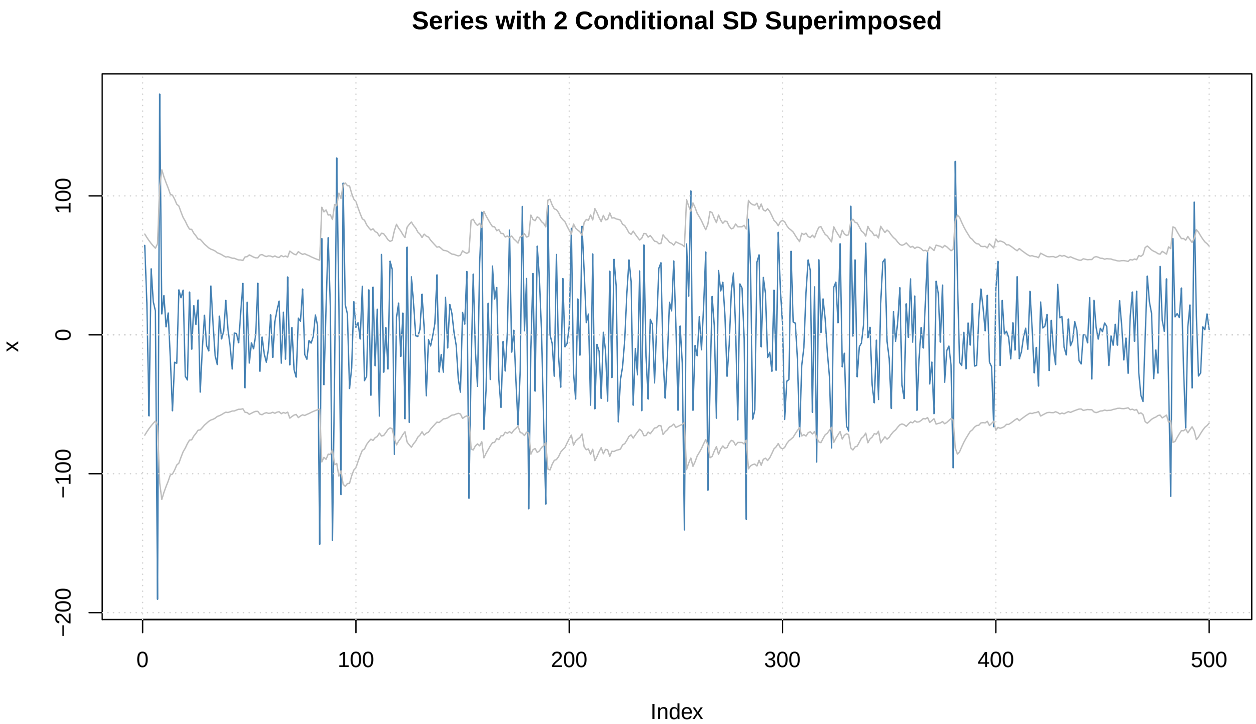



```
##
##  Box-Ljung test
##
## data:  residual
## X-squared = 66.837, df = 20, p-value = 5.929e-07

##
```

```
## Series Initialization:
##  ARMA Model:                arma
##  Formula Mean:              ~ arma(5, 0)
##  GARCH Model:               garch
##  Formula Variance:          ~ garch(1, 1)
##  ARMA Order:                5 0
##  Max ARMA Order:            5
##  GARCH Order:               1 1
##  Max GARCH Order:           1
##  Maximum Order:             5
##  Conditional Dist:          norm
##  h.start:                   6
##  llh.start:                 1
##  Length of Series:          500
##  Recursion Init:            mci
##  Series Scale:              41.0543
##
## Parameter Initialization:
##  Initial Parameters:          $params
##  Limits of Transformations:   $U, $V
##  Which Parameters are Fixed?  $includes
##  Parameter Matrix:
##                      U            V        params includes
##     mu     -0.02958912   0.02958912  0.0004379123     TRUE
##     ar1    -0.99999999   0.99999999 -0.3416401219     TRUE
##     ar2    -0.99999999   0.99999999 -0.3773843846     TRUE
##     ar3    -0.99999999   0.99999999 -0.3537906753     TRUE
##     ar4    -0.99999999   0.99999999 -0.2437663341     TRUE
##     ar5    -0.99999999   0.99999999 -0.2757320301     TRUE
##     omega   0.00000100 100.00000000  0.1000000000     TRUE
##     alpha1  0.00000001   0.99999999  0.1000000000     TRUE
##     gamma1 -0.99999999   0.99999999  0.1000000000    FALSE
##     beta1   0.00000001   0.99999999  0.8000000000     TRUE
##     delta   0.00000000   2.00000000  2.0000000000    FALSE
##     skew    0.10000000  10.00000000  1.0000000000    FALSE
##     shape   1.00000000  10.00000000  4.0000000000    FALSE
##  Index List of Parameters to be Optimized:
##     mu    ar1    ar2    ar3    ar4    ar5  omega alpha1  beta1
##      1      2      3      4      5      6      7      8     10
##  Persistence:                  0.9
##
##
## --- START OF TRACE ---
## Selected Algorithm: nlminb
##
```

```
## R coded nlminb Solver:
##
##   0:     637.07469: 0.000437912 -0.341640 -0.377384 -0.353791 -0.243766 -0.275732 0.1
##   1:     635.12577: 0.000437904 -0.341341 -0.376830 -0.354027 -0.244424 -0.276033 0.0
##   2:     634.80126: 0.000437818 -0.338443 -0.371898 -0.356251 -0.250399 -0.278932 0.0
##   3:     634.59553: 0.000437685 -0.335492 -0.366346 -0.358949 -0.257304 -0.282403 0.0
##   4:     634.43531: 0.000437440 -0.334945 -0.363242 -0.361368 -0.262225 -0.285280 0.0
##   5:     634.29527: 0.000437110 -0.336695 -0.363922 -0.362277 -0.263617 -0.287243 0.0
##   6:     634.10259: 0.000435251 -0.342599 -0.370069 -0.363278 -0.265341 -0.288119 0.0
##   7:     634.09567: 0.000435222 -0.342438 -0.369369 -0.363385 -0.265907 -0.288684 0.0
##   8:     633.95285: 0.000435217 -0.342408 -0.369244 -0.363411 -0.266009 -0.288767 0.0
##   9:     633.92536: 0.000435170 -0.342377 -0.368797 -0.363507 -0.266411 -0.289143 0.0
##  10:     633.88249: 0.000434986 -0.343018 -0.368924 -0.363452 -0.266446 -0.289435 0.0
##  11:     633.85447: 0.000434730 -0.343809 -0.368578 -0.363355 -0.266495 -0.289856 0.0
##  12:     633.83116: 0.000434129 -0.344500 -0.365659 -0.363329 -0.267284 -0.290947 0.0
##  13:     633.81312: 0.000433561 -0.342646 -0.363251 -0.363958 -0.269531 -0.291648 0.0
##  14:     633.80956: 0.000432571 -0.339600 -0.365153 -0.365672 -0.272564 -0.291857 0.0
##  15:     633.79963: 0.000430781 -0.342500 -0.363870 -0.367087 -0.273552 -0.295338 0.0
##  16:     633.79769: 0.000428904 -0.343519 -0.361462 -0.365682 -0.274480 -0.292216 0.0
##  17:     633.79762: 0.000428896 -0.343474 -0.361441 -0.365702 -0.274491 -0.292271 0.0
##  18:     633.79658: 0.000428893 -0.343455 -0.361432 -0.365711 -0.274496 -0.292296 0.0
##  19:     633.79503: 0.000428776 -0.342811 -0.361125 -0.366001 -0.274654 -0.293094 0.0
##  20:     633.79351: 0.000427443 -0.341675 -0.360864 -0.365444 -0.273360 -0.293438 0.0
##  21:     633.79313: 0.000424208 -0.342252 -0.362235 -0.366231 -0.274030 -0.294301 0.0
##  22:     633.79228: 0.000420478 -0.342806 -0.361849 -0.366941 -0.274341 -0.295331 0.0
##  23:     633.79226: 0.000415090 -0.341786 -0.361096 -0.365972 -0.274747 -0.294907 0.0
##  24:     633.79170: 0.000412525 -0.341499 -0.360951 -0.366486 -0.274835 -0.294343 0.0
##  25:     633.79152: 0.000410155 -0.341405 -0.360781 -0.367274 -0.274666 -0.293930 0.0
##  26:     633.79105: 0.000408301 -0.341656 -0.361189 -0.366587 -0.275147 -0.294405 0.0
##  27:     633.79099: 0.000408297 -0.341656 -0.361184 -0.366588 -0.275147 -0.294412 0.0
##  28:     633.79094: 0.000408235 -0.341660 -0.361110 -0.366617 -0.275151 -0.294524 0.0
##  29:     633.79085: 0.000407299 -0.341653 -0.361063 -0.366584 -0.275064 -0.294567 0.0
##  30:     633.78976: 0.000362647 -0.342148 -0.361117 -0.366722 -0.274607 -0.294925 0.0
##  31:     633.78922: 0.000317842 -0.342332 -0.361735 -0.367411 -0.275965 -0.294633 0.0
##  32:     633.78405: -6.51134e-05 -0.338943 -0.358712 -0.365868 -0.276065 -0.295330 0.
##  33:     633.77528: -0.000448109 -0.340934 -0.358869 -0.367827 -0.277582 -0.298220 0.
##  34:     633.76623: -0.000831325 -0.341745 -0.358715 -0.367301 -0.277351 -0.297903 0.
##  35:     633.68345: -0.00644891 -0.342576 -0.369286 -0.369043 -0.273869 -0.305214 0.0
##  36:     633.60773: -0.0125805 -0.352816 -0.363819 -0.363643 -0.273048 -0.302214 0.04
##  37:     633.49362: -0.0187121 -0.352985 -0.362901 -0.365004 -0.274065 -0.303516 0.04
##  38:     633.39483: -0.0248438 -0.348931 -0.362193 -0.366597 -0.275890 -0.301235 0.04
##  39:     633.36183: -0.0295891 -0.335642 -0.358716 -0.368295 -0.278140 -0.290324 0.04
##  40:     633.33878: -0.0295891 -0.342120 -0.360181 -0.367410 -0.277158 -0.295485 0.04
##  41:     633.33877: -0.0295891 -0.342129 -0.360224 -0.367430 -0.277163 -0.295509 0.04
##  42:     633.33876: -0.0295891 -0.342131 -0.360224 -0.367425 -0.277160 -0.295503 0.04
```

```
##  43:     633.33876: -0.0295891 -0.342132 -0.360224 -0.367425 -0.277159 -0.295503 0.04
##
## Final Estimate of the Negative LLH:
##  LLH:  2490.787    norm LLH:  4.981573
##          mu         ar1         ar2         ar3         ar4         ar5
## -1.21476072 -0.34213169 -0.36022407 -0.36742497 -0.27715949 -0.29550303
##       omega      alpha1       beta1
## 79.03691998  0.06601148  0.87205266
##
## R-optimhess Difference Approximated Hessian Matrix:
##                  mu           ar1           ar2           ar3           ar4
## mu     -0.457472849    0.15895878 -3.477774e-01    0.11047855  6.144113e-01
## ar1     0.158958784 -548.91316020  8.400644e+01   79.35724952  9.829975e+01
## ar2    -0.347777398   84.00643955 -5.196965e+02   88.64784166  5.481826e+01
## ar3     0.110478554   79.35724952  8.864784e+01 -567.65566137  9.577744e+01
## ar4     0.614411331   98.29975099  5.481826e+01   95.77744069 -5.374288e+02
## ar5    -0.257030691   35.02504586  8.889707e+01   95.36704488  1.022688e+02
## omega   0.004791553    0.01125851 -7.489342e-04   -0.03392825  2.506991e-03
## alpha1 -2.093524959  -40.03538677 -1.062704e+01   37.89851129  1.020662e+01
## beta1   4.286530629  -67.50809470 -3.862055e+01   20.72436214  4.337998e+01
##                 ar5         omega        alpha1         beta1
## mu      -0.25703069  4.791553e-03     -2.093525      4.286531
## ar1      35.02504586  1.125851e-02    -40.035387    -67.508095
## ar2      88.89706962 -7.489342e-04    -10.627045    -38.620549
## ar3      95.36704488 -3.392825e-02     37.898511     20.724362
## ar4     102.26884743  2.506991e-03     10.206617     43.379978
## ar5    -579.38482493  6.908145e-02    -68.159641     43.852679
## omega     0.06908145 -1.202629e-02    -10.157877    -13.712071
## alpha1  -68.15964056 -1.015788e+01 -13543.451549 -13587.644372
## beta1    43.85267923 -1.371207e+01 -13587.644372 -16805.902479
## attr(,"time")
## Time difference of 0.02876115 secs
##
## --- END OF TRACE ---
##
##
## Time to Estimate Parameters:
##  Time difference of 0.108247 secs
```



13 Conclusions

It was really interesting!

14 TODO

prima diff, poi prima diff seasonal, check acf pacf, check no trend(trend se con decadono a 0 velocemente) identificare i picchi identificare l estate doppia seasonality una settimanale e una annuale ARCH GARCH VAR<—- stabilizzare con trasformazioni