STDA-project

```
require(zoo)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
require(xts)
```

```
## Loading required package: xts
```

```
data <- read.csv("sims2018milan.csv",sep = ";")
data <- rbind(data, read.csv("foreignsim_2019-11-07.csv",sep = ";"))

data$prefix <- NULL
data$country <- NULL
data$num <- NULL
data$total.ita.sim <- NULL
```

```
ll <- aggregate.data.frame(data$total.foreign.sim,by=list(data$date),FUN=mean)
names(ll)[2] <- "total.foreign.sim"
names(ll)[1] <- "Date"
data <- ll

dim(data)
```

```
## [1] 658   2
```

```
data <- data[-c(656,657,658), ]
dim(data)
```

```
## [1] 655   2
```

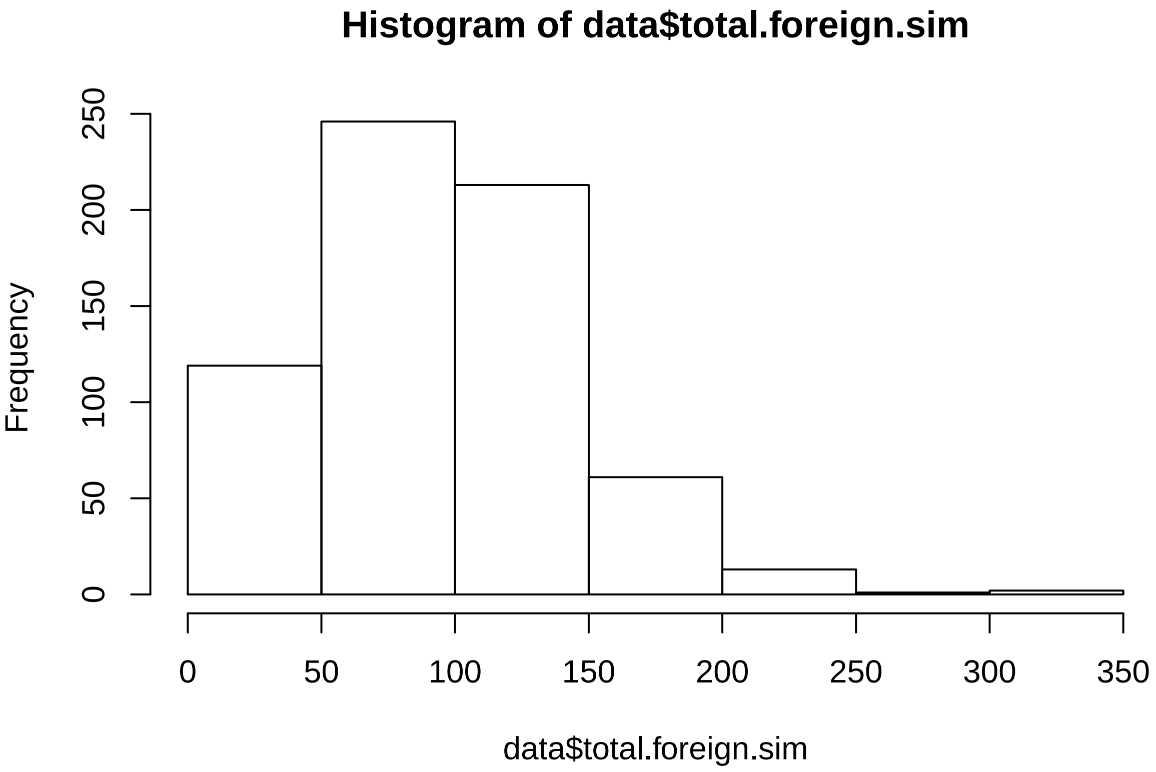
```
print("minimum, lower-hinge, median, upper-hinge, maximum)")
```

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

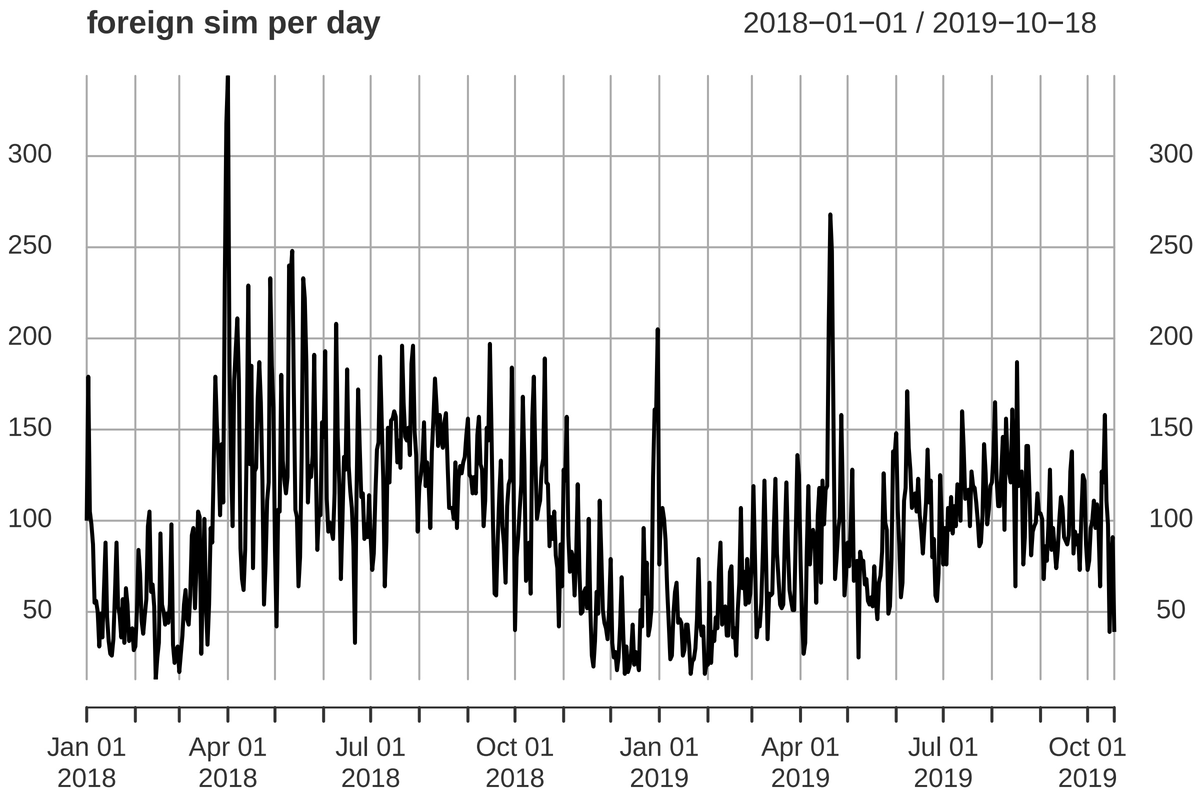
```
fivenum(data$total.foreign.sim)
```

```
## [1]  13  59  95 124 344
```

```
hist(data$total.foreign.sim)
```

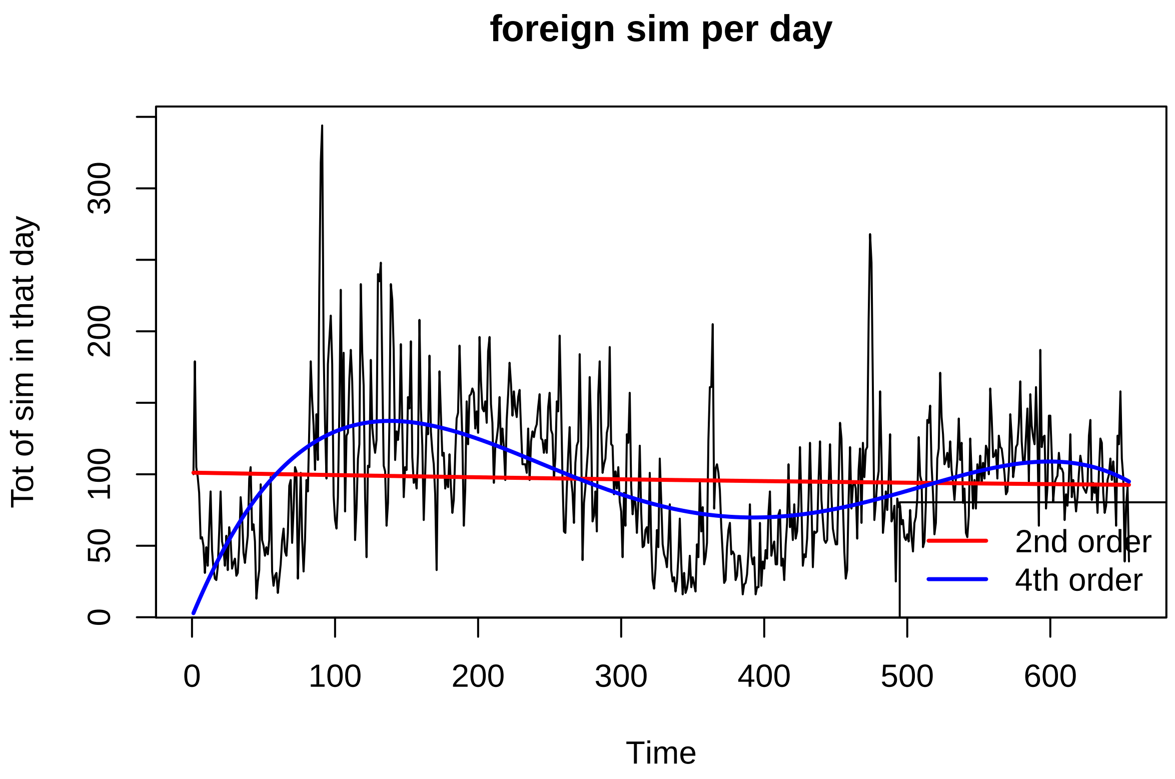


```
data$Date <- as.Date(data$Date, format = "%Y-%m-%d")
#typeof(data$date[1])
data.xts <- xts(data$total.foreign.sim, order.by=data$Date, frequency = 365)
data.ts <- ts(data$total.foreign.sim)

main <- "foreign sim per day"
ylab<-"Tot of sim in that day"
plot(data.xts,ylab=ylab,main=main)
```

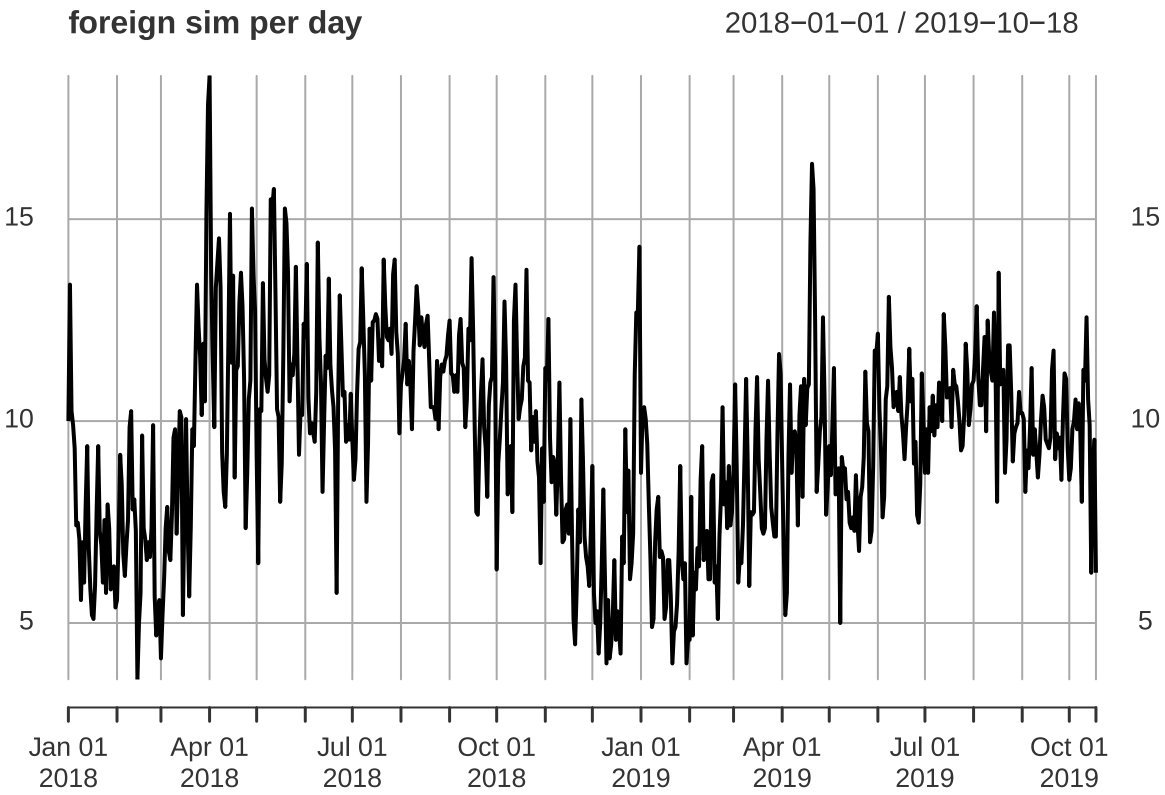


```r
tt<-as.numeric(time(data.ts))
fit2<-lm(data.ts~poly(tt,degree=2,raw=TRUE))
fit4<-lm(data.ts~poly(tt,degree=4,raw=TRUE))
```

```r
main <- "foreign sim per day"
plot(data.ts,ylab=ylab,main=main)
lines(tt,predict(fit2),col='red',lwd=2)
lines(tt,predict(fit4),col='blue',lwd=2)
legend("bottomright",legend = c("2nd order","4th order"),lwd=2,lty=1,col=c("red","blue"))
```



```
data.xts.log <- xts(sqrt(data$total.foreign.sim), order.by=data$Date, frequency = 365)

main <- "foreign sim per day"
ylab<-"Tot of sim in that day"
plot(data.xts.log,ylab=ylab,main=main)
```

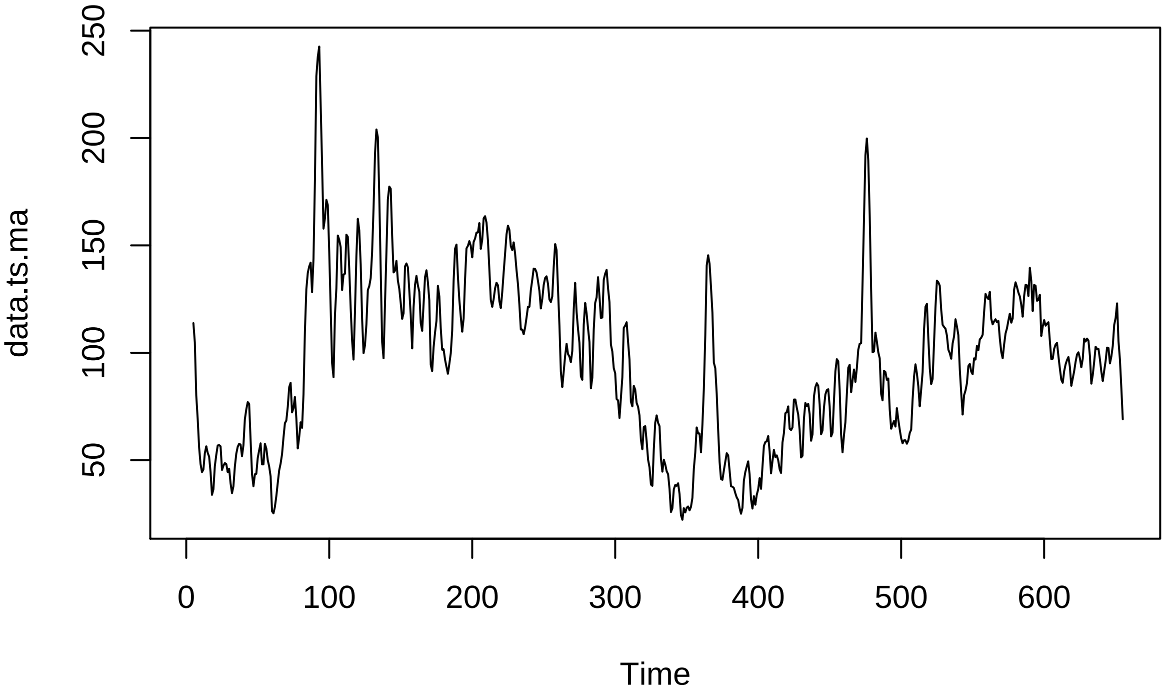


Decomposing non seasonal data

```
require(TTR)
```

```
## Loading required package: TTR
```

```
data.ts.ma <- SMA(data.ts, n=5)
plot.ts(data.ts.ma)
```



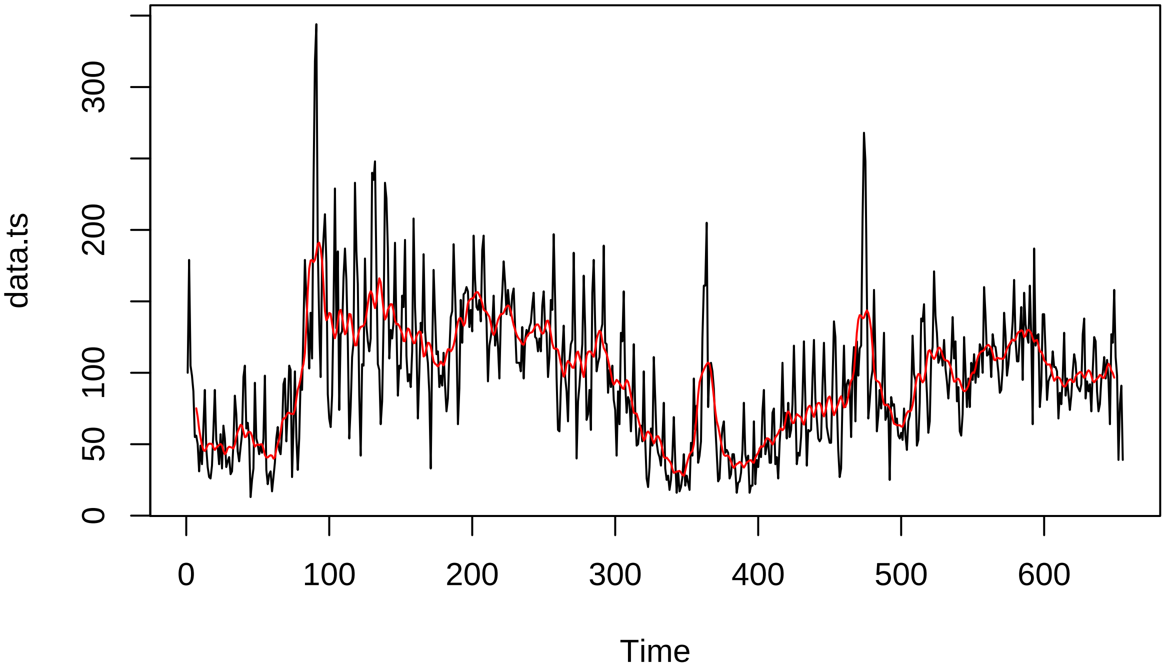
```
# Decomposing seasonal data
#data.ts.dec <- decompose(data.ts)
#plot.ts(data.ts.dec)
#Error in decompose(data.ts) : time series has no or less than 2 periods
# so no seasonality ca be fou d by R
```

```
#install.packages("forecast")
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.6.2
```

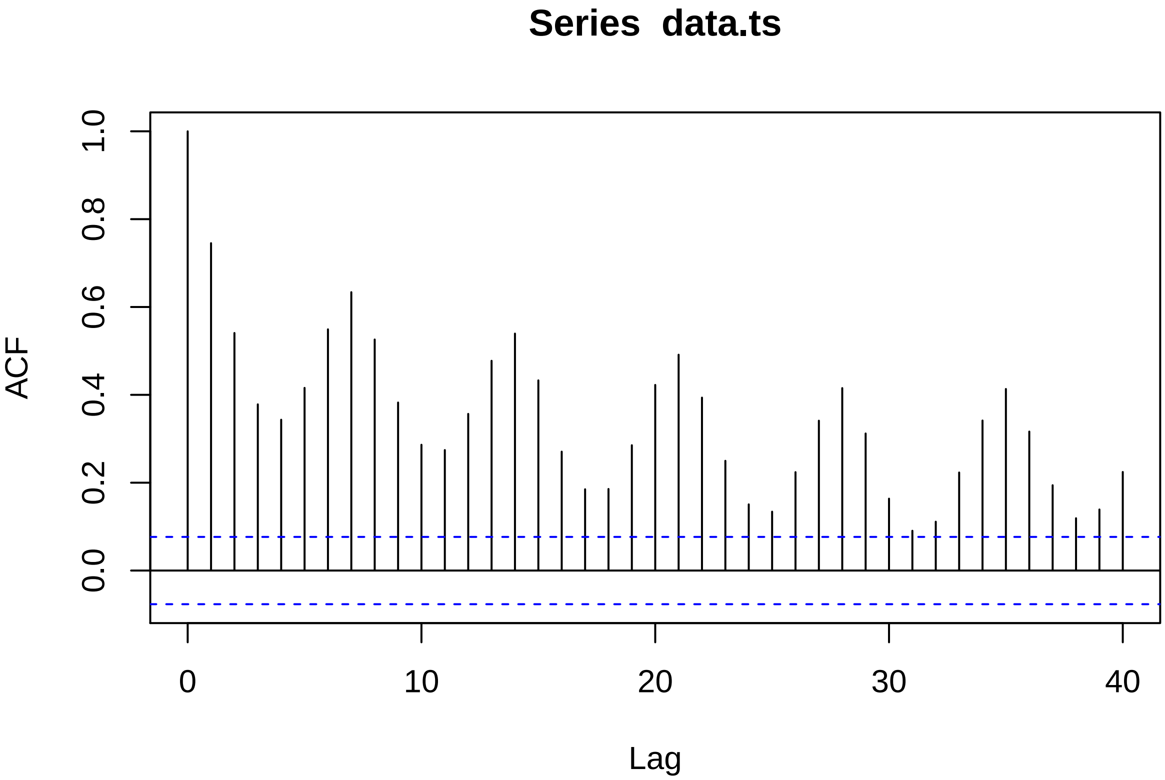
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
trend.data = ma(data.ts, order = 12, centre = T)
plot(data.ts)
lines(trend.data, col="red")
```



acf & pacf

```
acf(data.ts, 40)
```

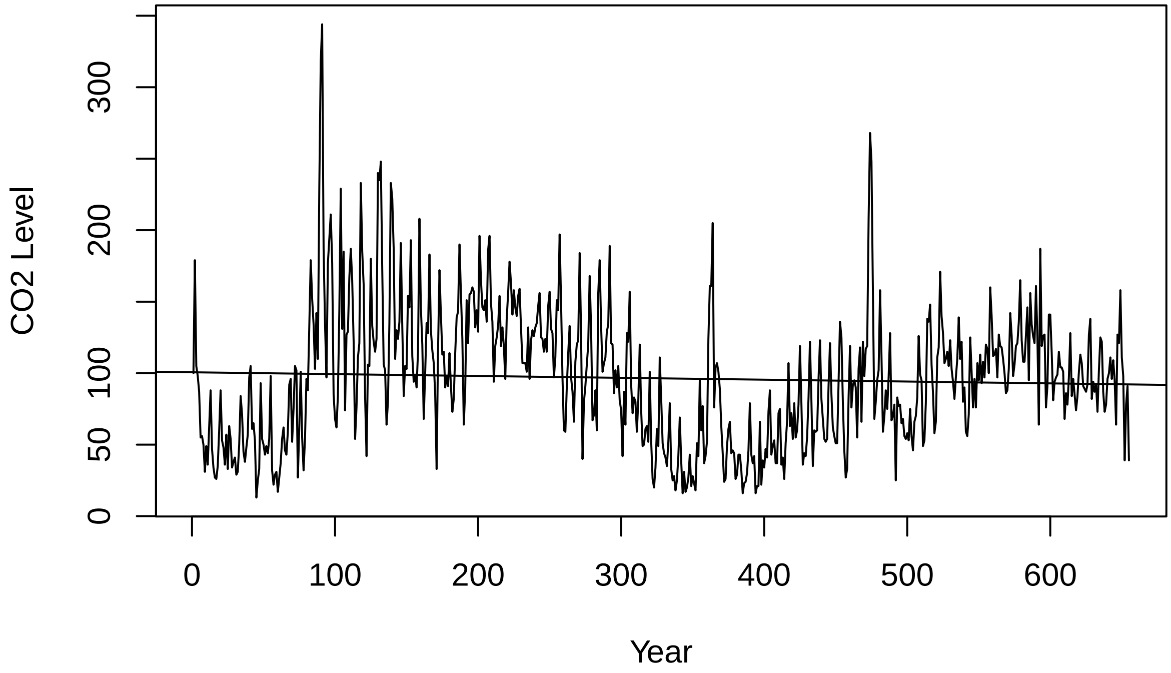


Every 7 lags the peak recurs

```r
# Part (c)
t = time(data.ts)
FIT = lm(data.ts ~ t)
summary(FIT)
```

```
##
## Call:
## lm(formula = data.ts ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -87.040 -37.594  -0.507  28.413 244.556
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 100.623489   3.729419  26.981   <2e-16 ***
## t            -0.012965   0.009851  -1.316    0.189
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.67 on 653 degrees of freedom
## Multiple R-squared:  0.002646,   Adjusted R-squared:  0.001118
## F-statistic: 1.732 on 1 and 653 DF,  p-value: 0.1886
```

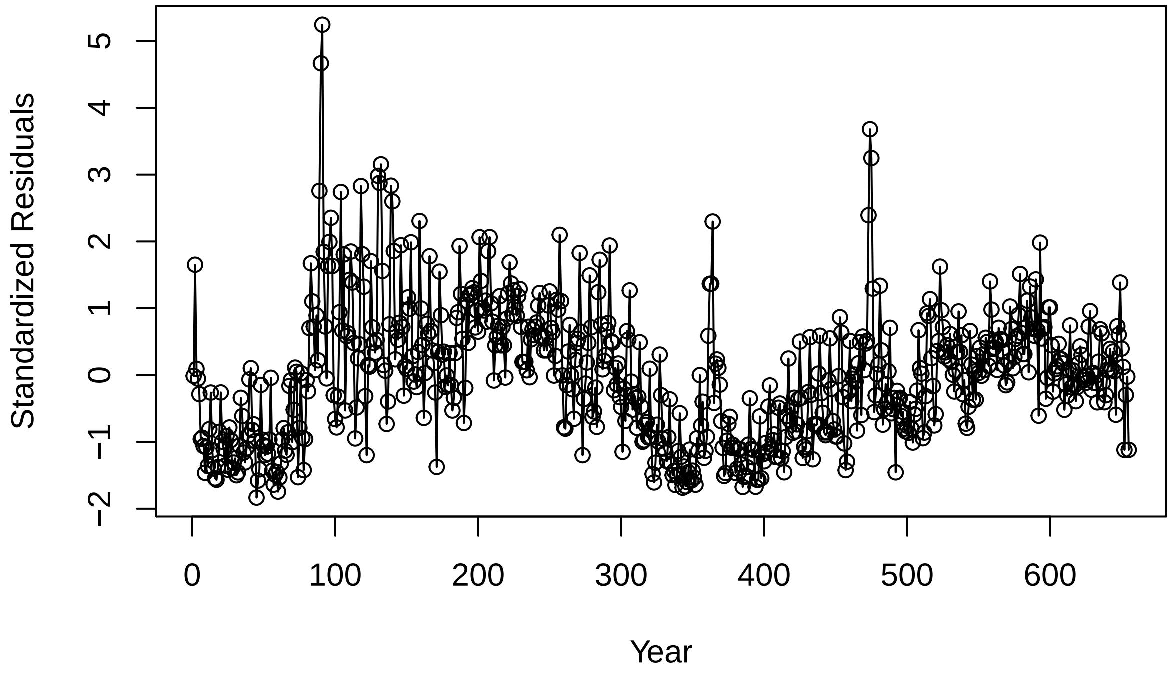
```r
X = as.vector(time(data.ts))
plot(X, data.ts, xlab = "Year", ylab = "CO2 Level", type = "l")
abline(FIT)
```



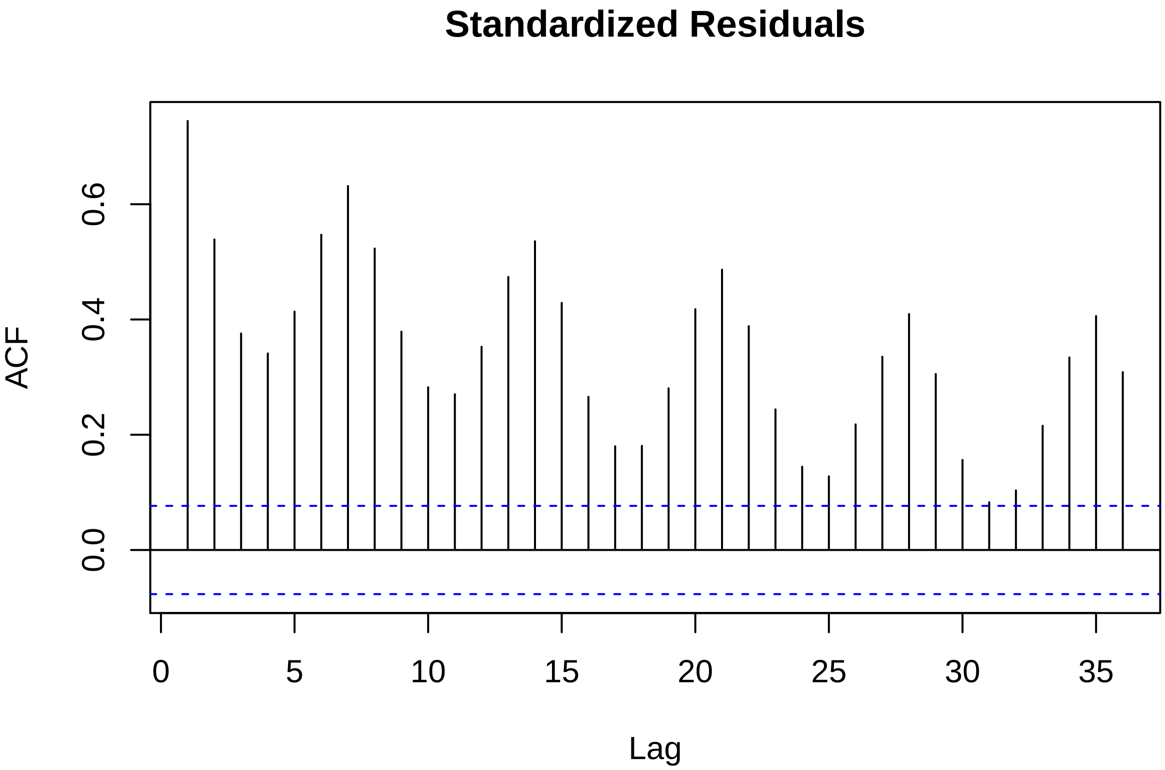
```r
require(TSA)
```

```
## Loading required package: TSA

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

##
## Attaching package: 'TSA'

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

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

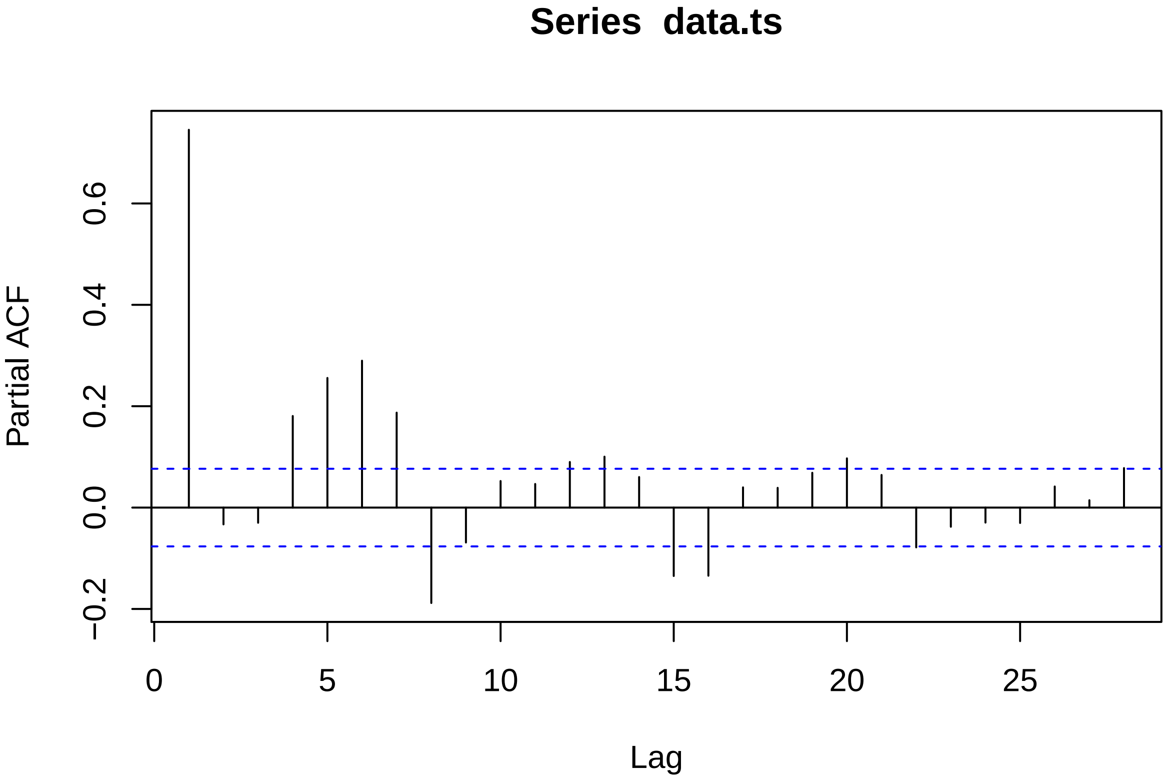
```
RES = rstudent(FIT)
plot(X, RES, xlab = "Year", ylab = "Standardized Residuals", type = "l")
points(X, RES)
```



```
acf(RES, main = "Standardized Residuals", 36)
```



```r
pacf(data.ts)
```

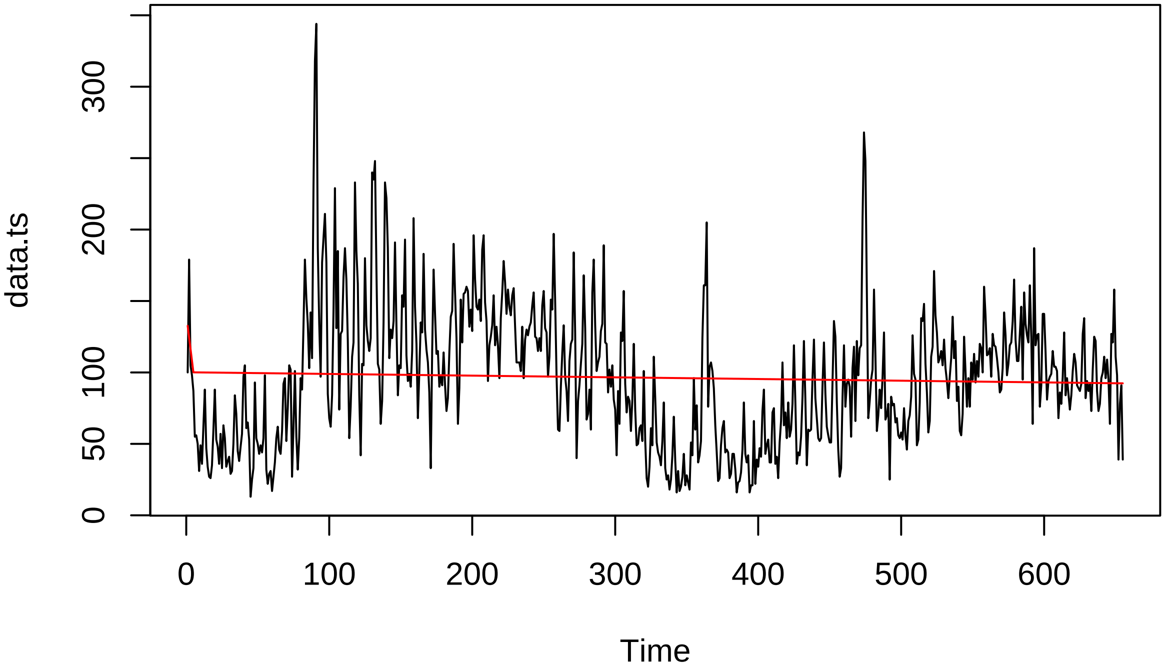


Detrending

```
require(pracma)
```

```
## Loading required package: pracma
```

```
#detrend.data = data.ts - trend.data
detrend.data = detrend(as.matrix(data.ts), 'linear', 5)
plot(data.ts)
lines(data.ts - detrend.data, col="red")
```

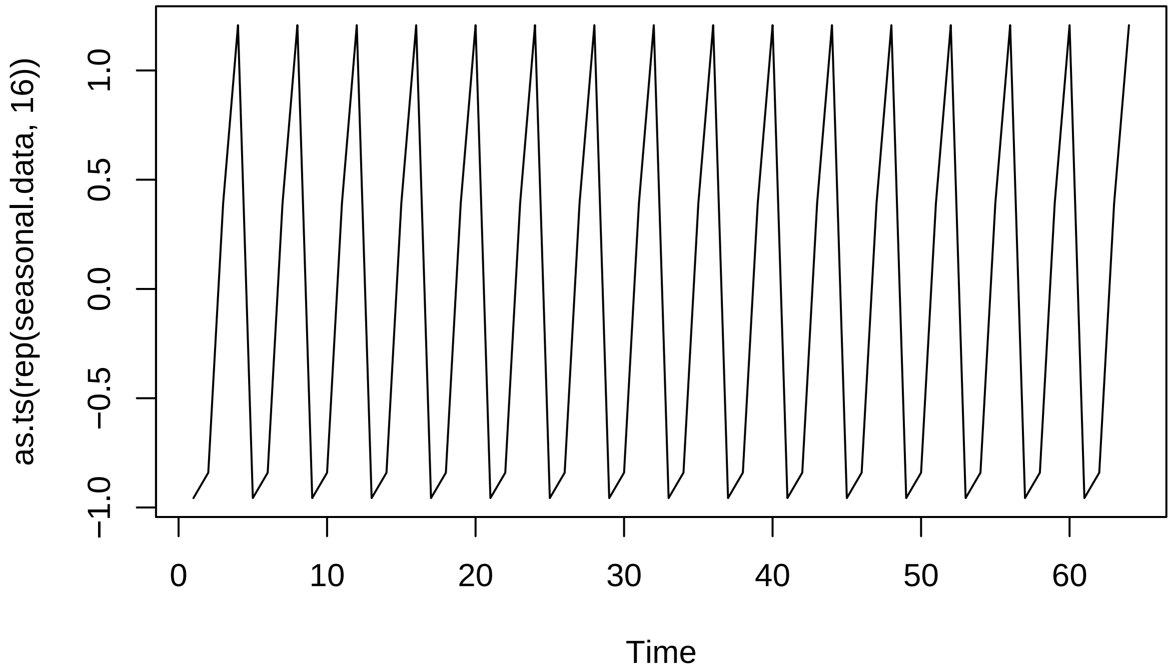


Finding seasonality

```
m_data = t(matrix(data = detrend.data, nrow = 4))
```

```
## Warning in matrix(data = detrend.data, nrow = 4): data length [655] is not a
## sub-multiple or multiple of the number of rows [4]
```

```
seasonal.data = colMeans(m_data, na.rm = T)
plot(as.ts(rep(seasonal.data,16)))
```

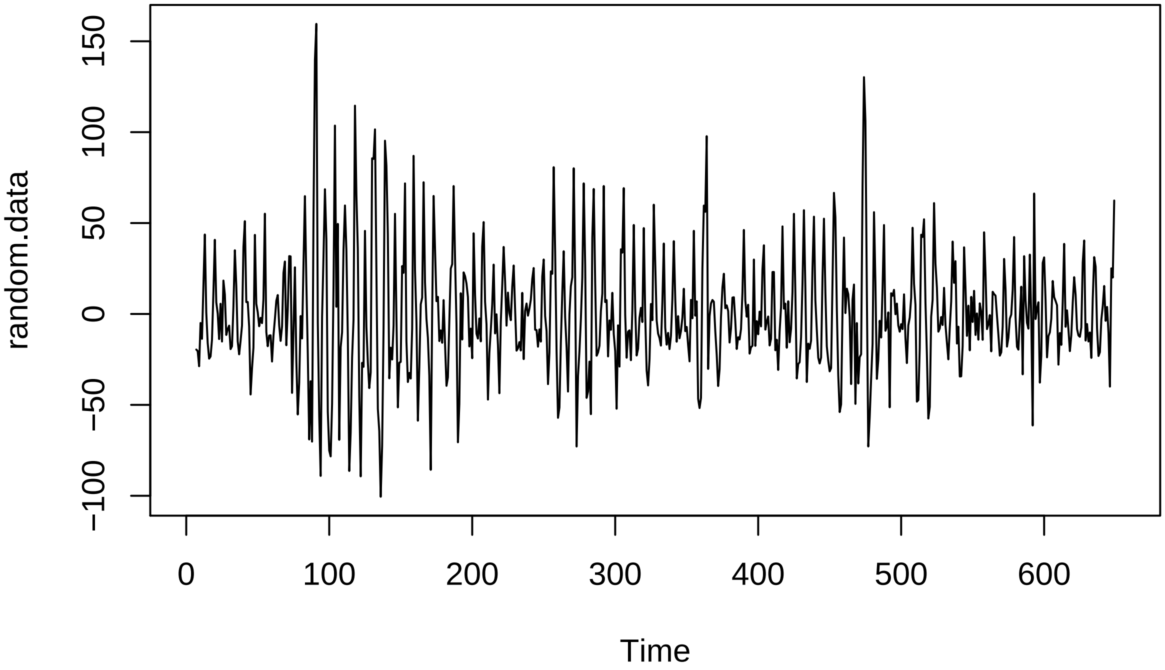


the rest is residuals

```
random.data = data.ts - trend.data - seasonal.data
```

```
## Warning in `-.default`(data.ts - trend.data, seasonal.data): longer object
## length is not a multiple of shorter object length
```

```
plot(random.data)
```



```
Box.test(random.data, lag=10, fitdf=0)
```

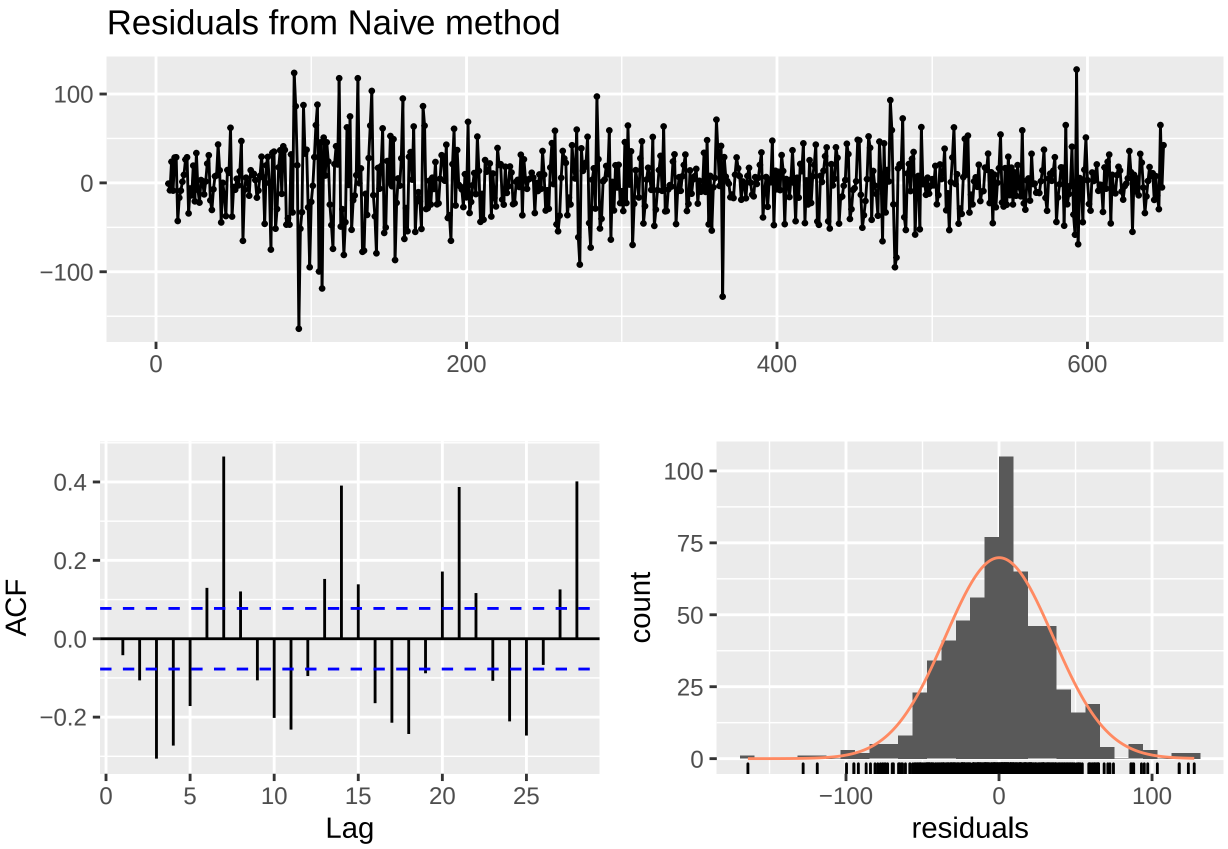
```
##
##  Box-Pierce test
##
## data:  random.data
## X-squared = 826.14, df = 10, p-value < 2.2e-16
```

```
Box.test(random.data, lag=10, fitdf=0, type="Lj")
```

```
##
##  Box-Ljung test
##
## data:  random.data
## X-squared = 834.97, df = 10, p-value < 2.2e-16
```

There is strong evidence that data is non stationary but im not able to decompose the seasonality

```
checkresiduals(naive(random.data))
```



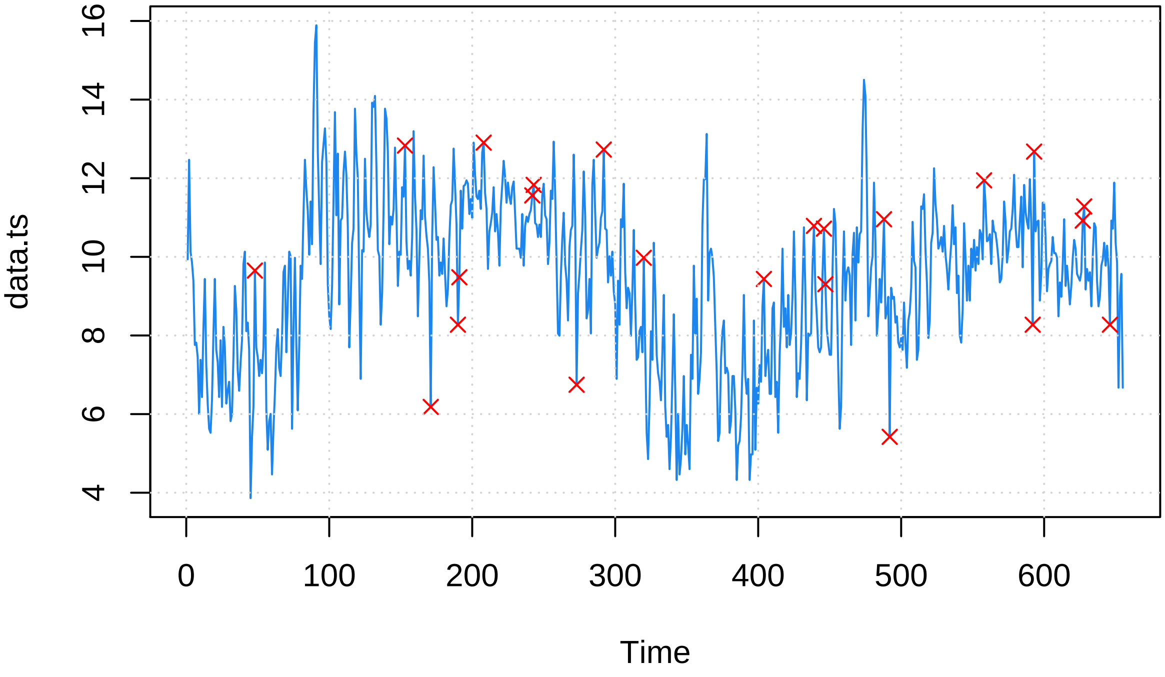
```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 331.5, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
```

trying to fix

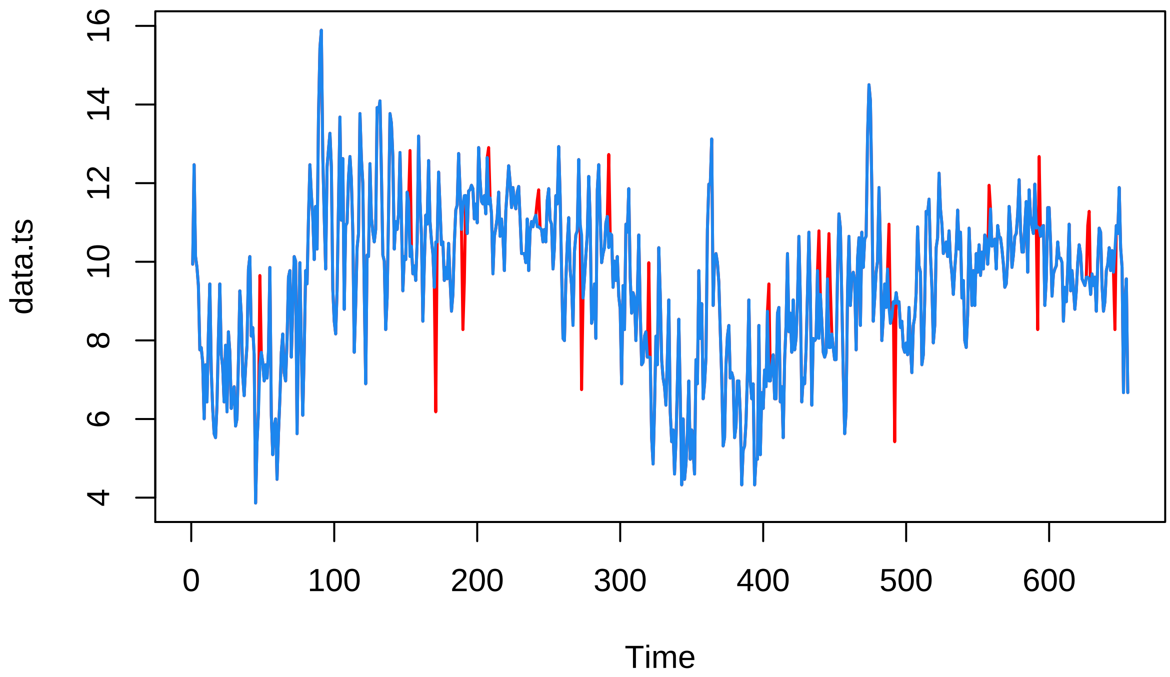
```
library(pracma)
data.ts.box <- BoxCox(data.ts,lambda=0.3)
data.ts <- data.ts.box
SP.hampel <- hampel(data.ts, 5, 3)
SP.hampel$ind
```

```
##  [1]  48 153 171 190 191 208 242 243 273 292 320 404 439 446 447 488 492 558 592
## [20] 593 627 628 646
```

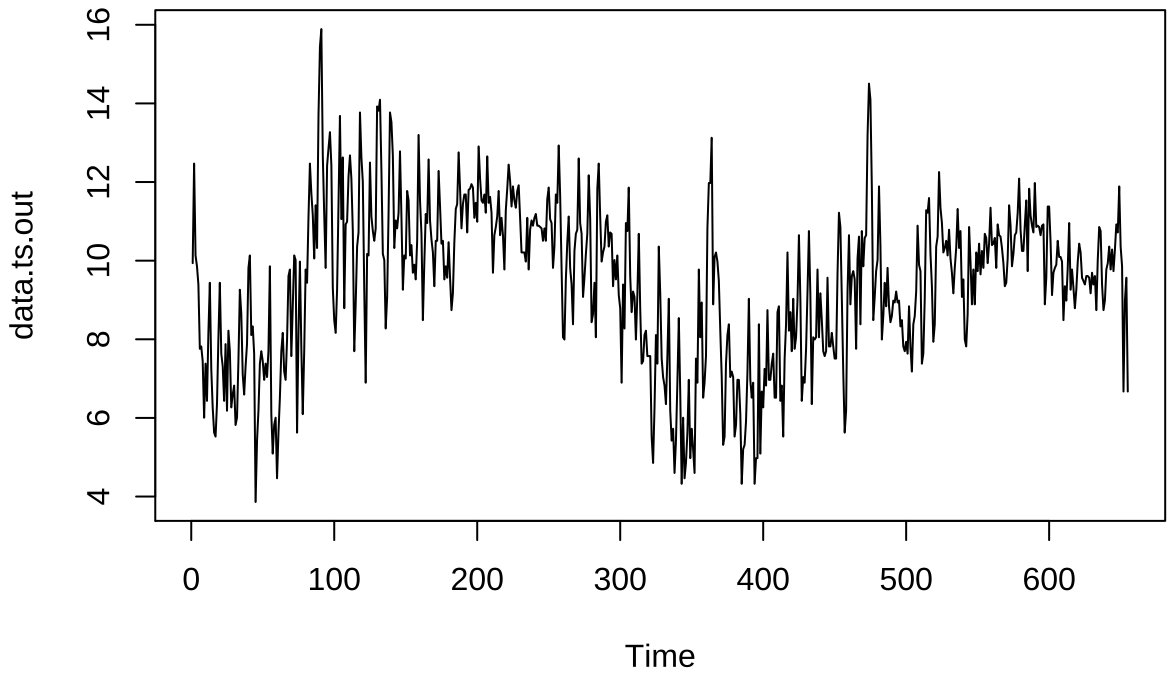
```
SP.hampel.outlier.times <- vector()
SP.hampel.outlier.values <- vector()
i <- 1
for (ind in SP.hampel$ind) {
    SP.hampel.outlier.times[i] <- time(data.ts)[ind]
    SP.hampel.outlier.values[i] <- data.ts[ind]
    i <- i + 1
}
options(repr.plot.width=8, repr.plot.height=4)
plot(data.ts, col="dodgerblue2")
points(SP.hampel.outlier.times, SP.hampel.outlier.values,
    pch=4, col="red")
grid()
```



```
plot(data.ts, lwd=1.5, col="red")
lines(SP.hampel$y, lwd=1.5, col="dodgerblue2")
```



```
data.ts.out <- ts(SP.hampel$y)
plot(data.ts.out)
```



```
data.ts <- data.ts.out
```

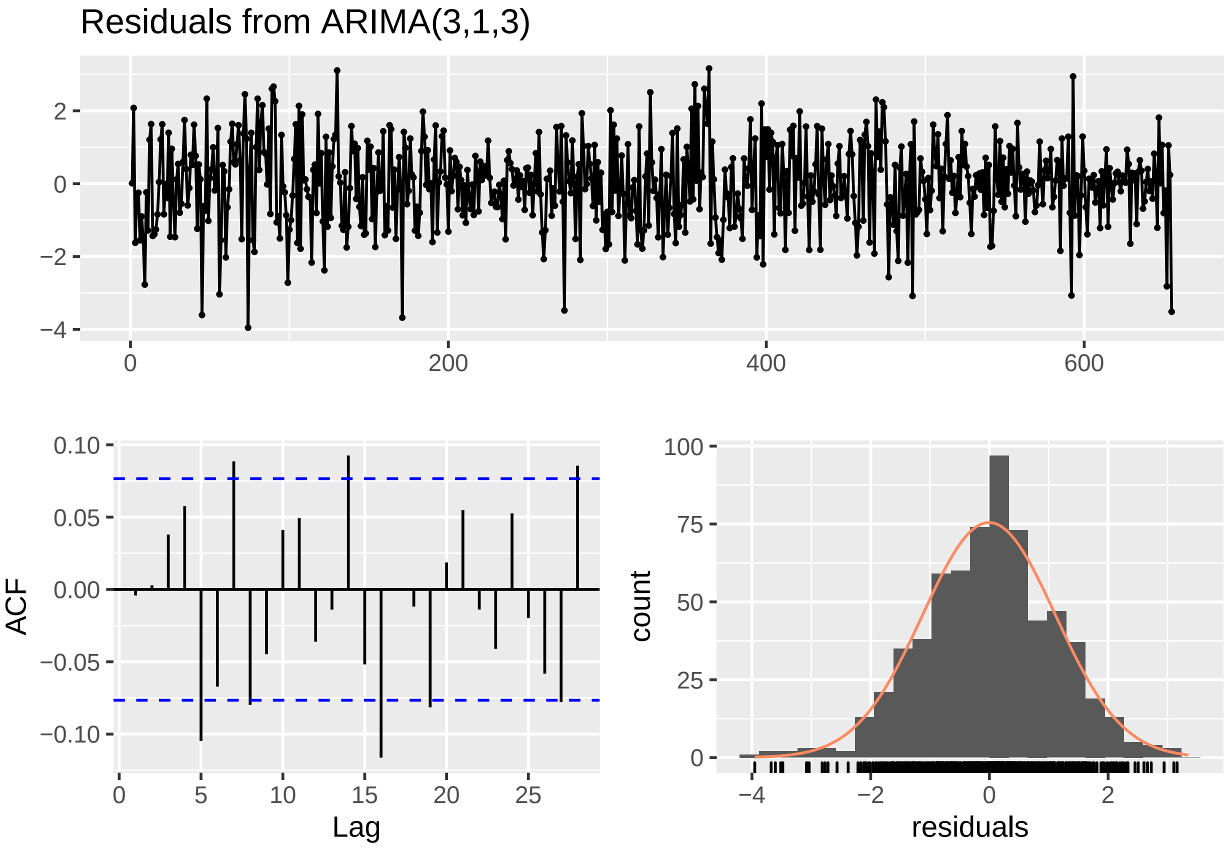
```
require(fpp)
```

```
## Loading required package: fpp
## Loading required package: fma
## Loading required package: expsmooth
## Loading required package: lmtest
## Loading required package: tseries
```

```
# data.ts.box <- BoxCox(data.ts,lambda=0.4)
fit <- Arima(data.ts.box, order=c(3,1,3))
summary(fit)
```

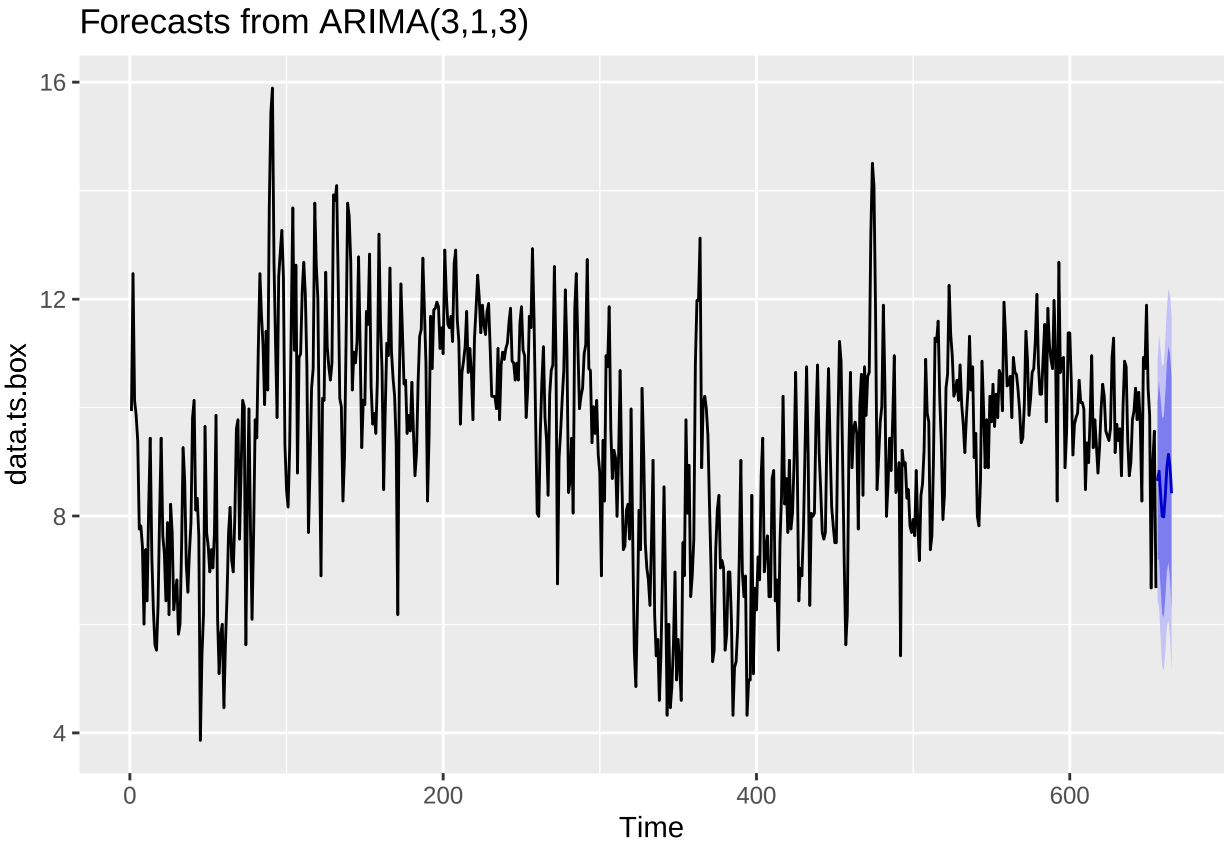
```
## Series: data.ts.box
## ARIMA(3,1,3)
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2      ma3
##       1.4616  -1.2705  0.2248  -1.9163  1.8068  -0.7039
## s.e.  0.0881   0.1087  0.0851   0.0708  0.0862   0.0563
##
## sigma^2 estimated as 1.265:  log likelihood=-1003.34
## AIC=2020.68   AICc=2020.85   BIC=2052.06
##
## Training set error measures:
##                        ME     RMSE       MAE       MPE     MAPE      MASE
## Training set -0.007186594 1.118824 0.8731546 -1.566833 10.16686 0.8098616
##                      ACF1
## Training set -0.004136471
```

```
checkresiduals(fit)
```

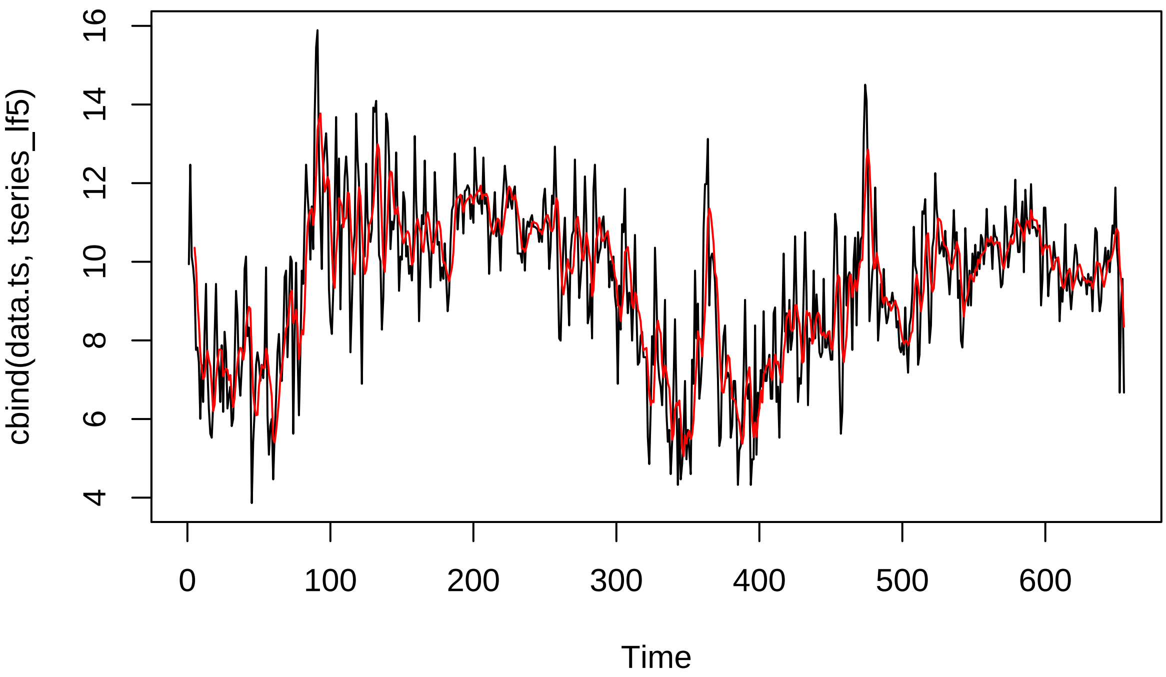


```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(3,1,3)
## Q* = 25.301, df = 4, p-value = 4.376e-05
##
## Model df: 6.   Total lags used: 10
```

```
autoplot(forecast(fit))
```



```
tseries_lf5 <- filter(data.ts, filter = rep(1/5, 5), sides = 1)
plot.ts(cbind(data.ts, tseries_lf5), plot.type = 'single', col = c('black', 'red'))
```


cbind(data.ts, tseries_lf5)
4
6
8
10
12
14
16
0
100
200
300
400
500
600
Time