PRE-PROCESSING STEPS TO REMOVE NON-STATIONARITY

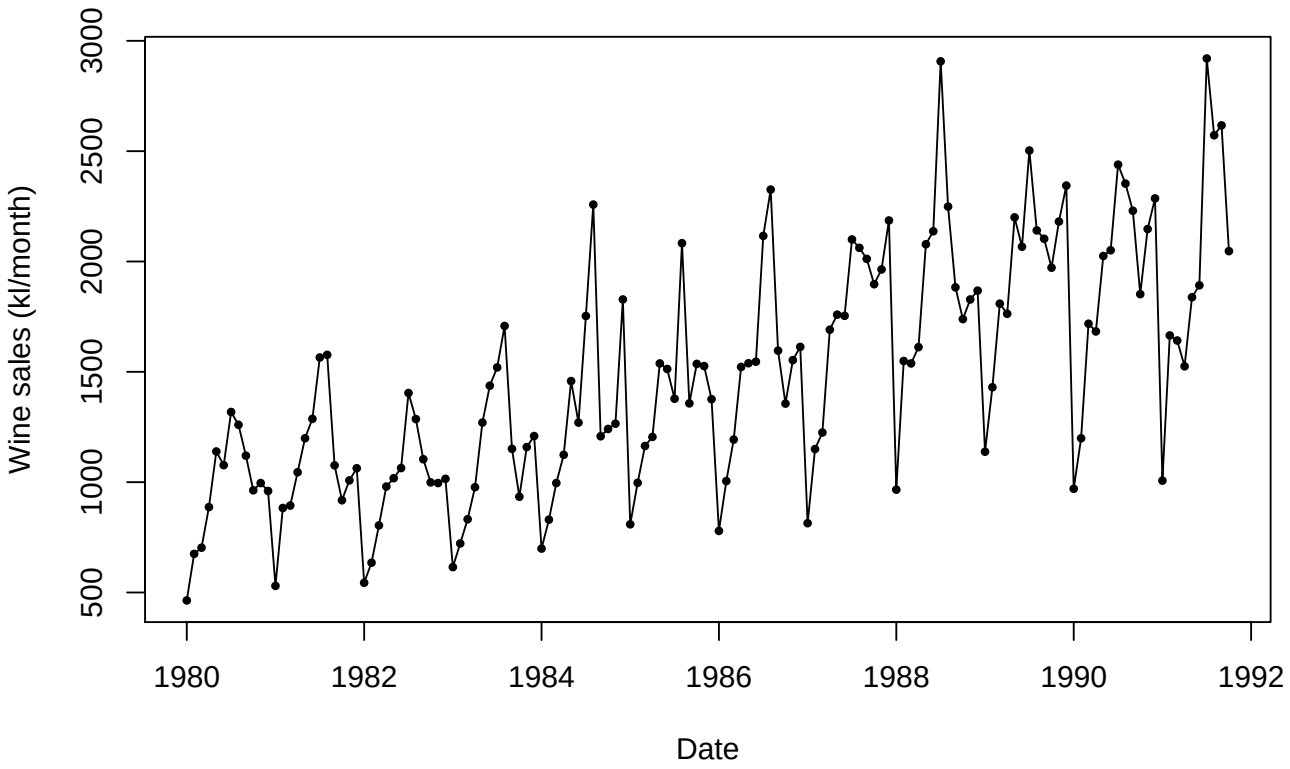
Typically, real data are not plausibly represented by a stationary time series model, as on inspection we typically observe non-stationarity in mean or variance. In addition, there is very often evident *seasonality*, that is, an apparent systematic periodic variation in mean for the process. There are standard approaches to removing these *deterministic* non-stationarities that can be applied as pre-processing steps that achieve the decomposition for observed process $\{X_t\}$

$$X_t = m_t + s_t + Y_t \qquad t \in \mathbb{Z}$$

- m_t is deterministic time-varying **mean**;
- s_t is a deterministic **seasonality** with period d, say (so that $s_t = s_{t+kd}$, $k = 1, 2, \ldots$);
- Y_t is stationary zero mean process.

To illustrate the steps, consider the Australian wine sales data from Brockwell & Davis (see p. 20). The data are available on the course website and comprise 142 monthly observations from January 1980 to October 1991. The data are plotted as follows:

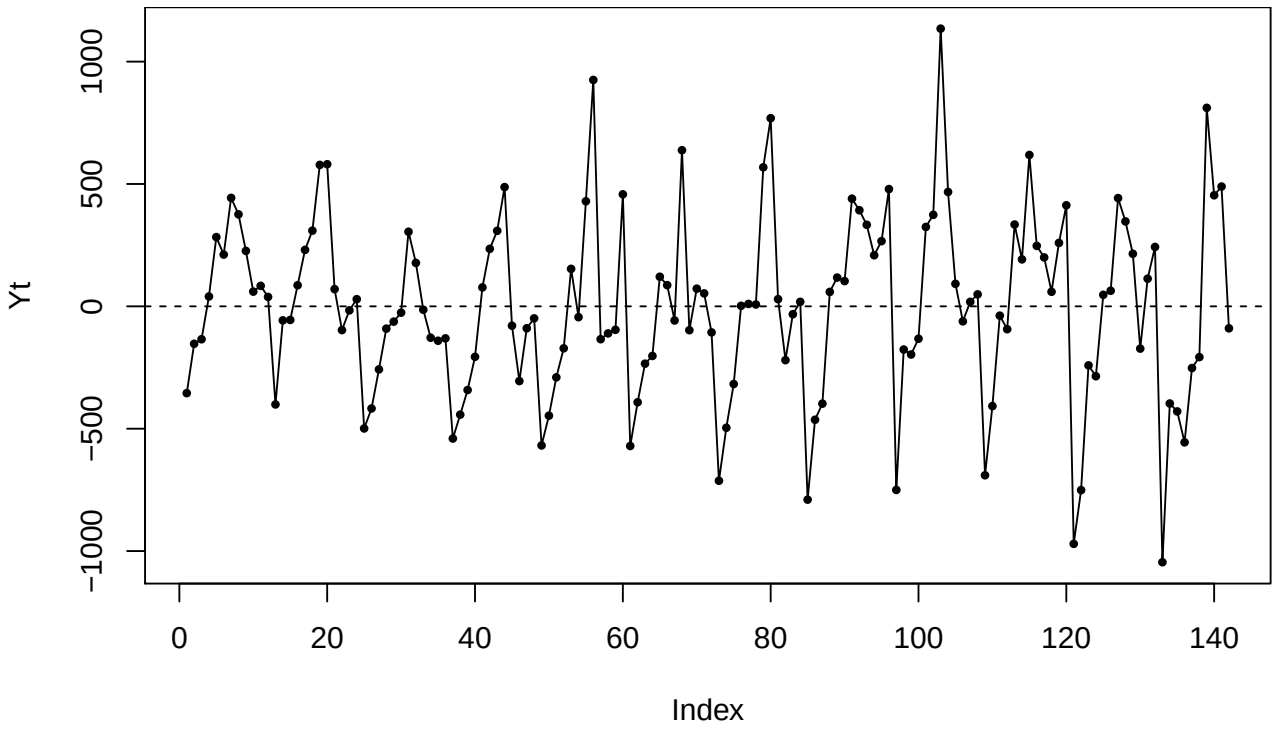
```
#Read in data: Jan 1980 - Oct 1991
Wine<-scan("http://www.math.mcgill.ca//dstephens//545//Data//wine.txt")
n<-length(Wine); tvec<-c(1:n)
Xt<-ts(Wine,frequency=12,start=c(1980,1))
par(mar=c(4,4,1,1))
plot(Xt,xlab='Date',ylab='Wine sales (kl/month)'); points(Xt,pch=19,cex=0.5)
```



There is an apparent linear increasing trend in these data, meaning that the process is non-stationary. We may attempt to remove the trend by fitting a simple linear regression model in time (using the `lm` function in `R`) to represent m_t as $m_t = \beta_0 + \beta_1 t$ to use the form

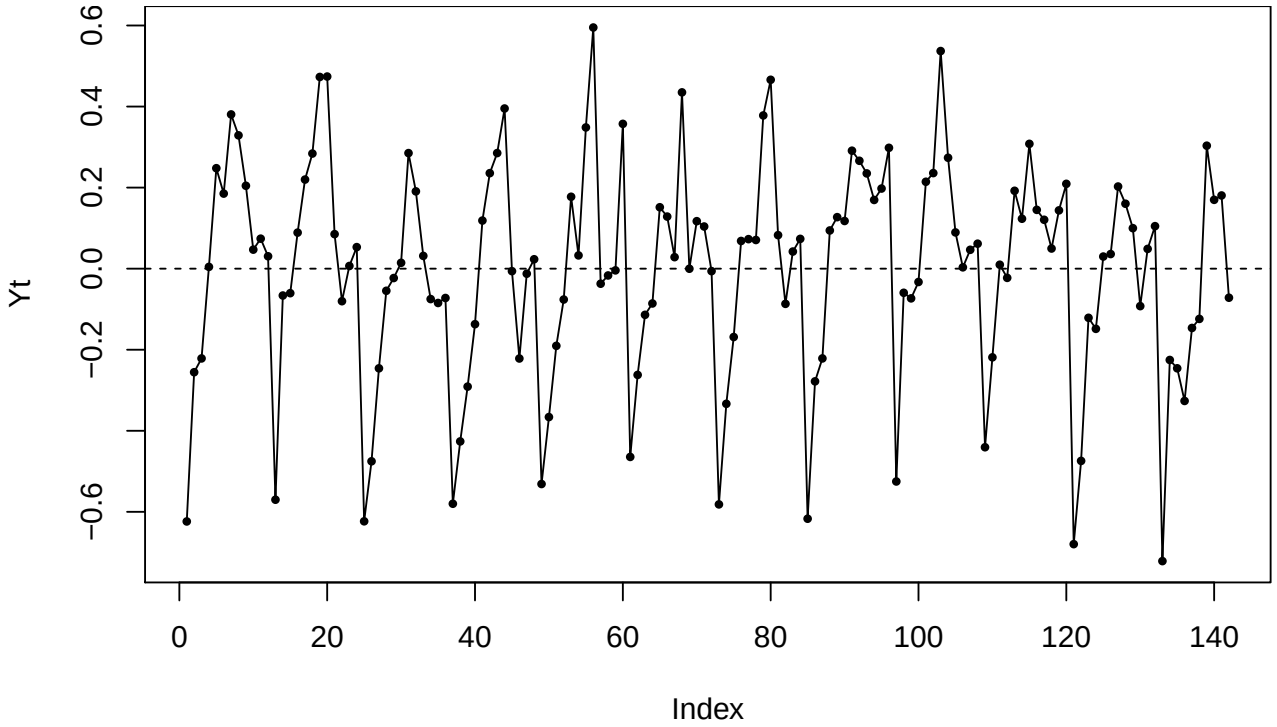
$$X_t = m_t + Y_t = \beta_0 + \beta_1 t + Y_t$$

```
Wine.fit<-lm(Wine~tvec)
mt<-Wine.fit$coef[1]+Wine.fit$coef[2]*tvec
Yt<-Wine-mt
par(mar=c(4,4,0,1))
plot(Yt,type='l'); points(Yt,pch=19,cex=0.5);abline(h=0,lty=2)
```



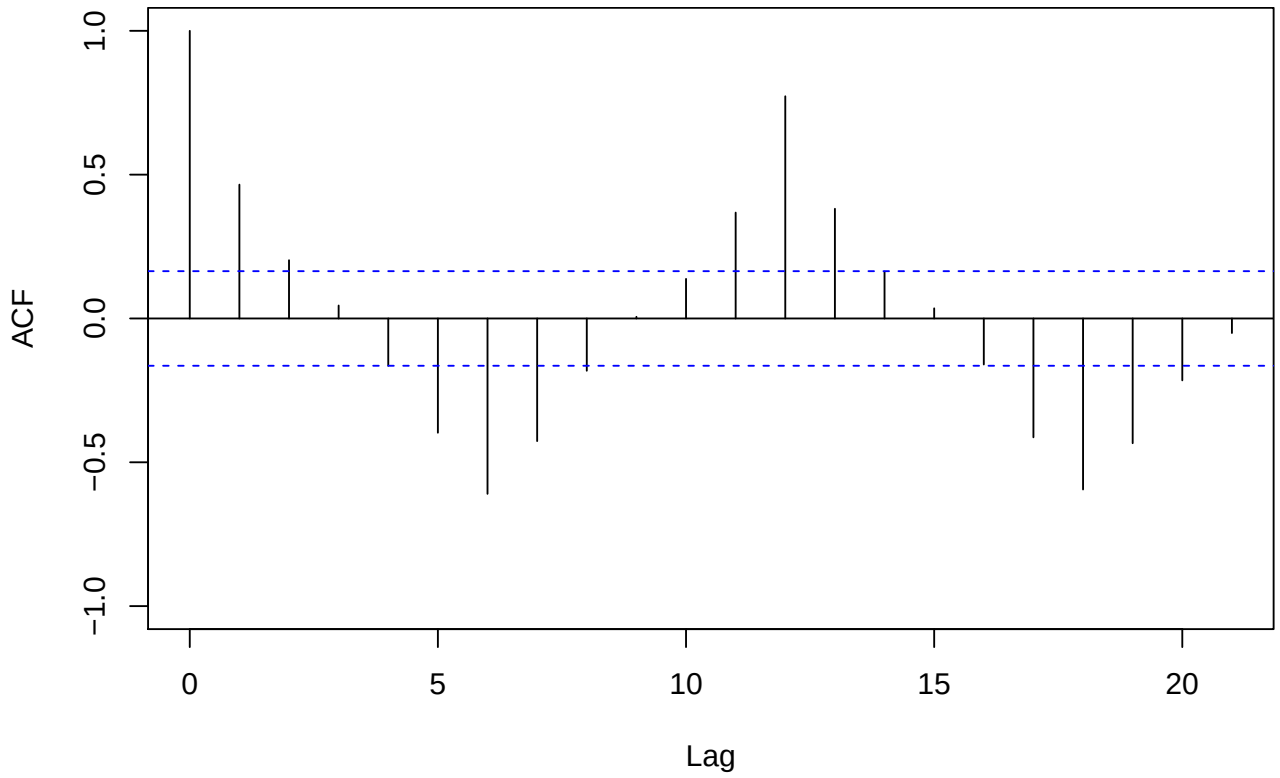
The mean level now appears constant, but there is an apparent increasing trend in the variance of the process. To address this, we take a (natural) log transform of the original data, and repeat the fit.

```
lWine<-log(Wine)
lWine.fit<-lm(lWine~tvec)
mt<-lWine.fit$coef[1]+lWine.fit$coef[2]*tvec
Yt<-lWine-mt
par(mar=c(4,4,0,1))
plot(Yt,type='l'); points(Yt,pch=19,cex=0.5); abline(h=0,lty=2)
```



The variance now appears approximately constant. If we plot the acf of the resulting $\{Y_t\}$ process, we see that there is a notable pattern:

```
par(mar=c(4,4,0,1))
acf(Yt,ylim=range(-1,1))
```

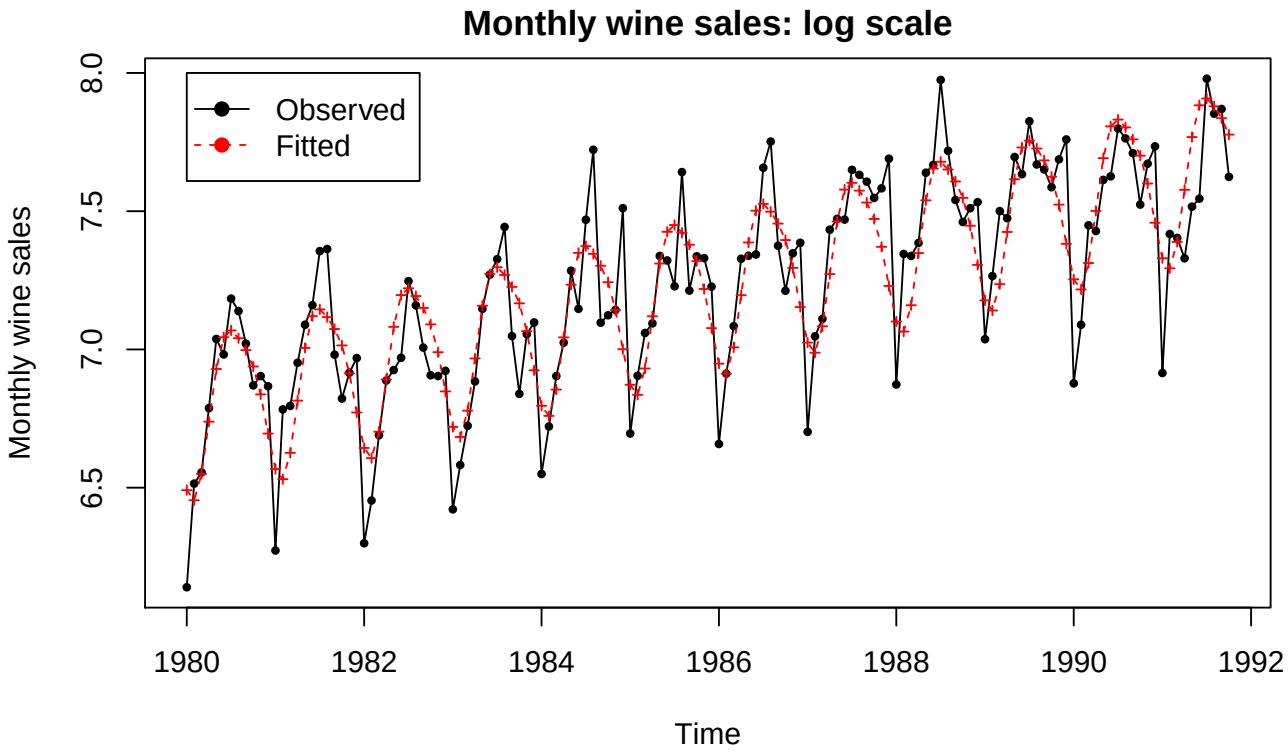


From this we deduce a periodic component is present in the $\{Y_t\}$ process – this is evident from the data plot also. We can attempt to address this by using a deterministic mean model for m_t that also has periodic components, such as the so-called *harmonic regression* (with trend). Here a model with two periodic components with cycles of 12 months and 6 months, for which the model would be

$$m_t = \beta_0 + \beta_1 t + \beta_2 \cos(2\pi t/12) + \beta_3 \sin(2\pi t/12) + \beta_4 \cos(2\pi t/6) + \beta_5 \sin(2\pi t/6).$$

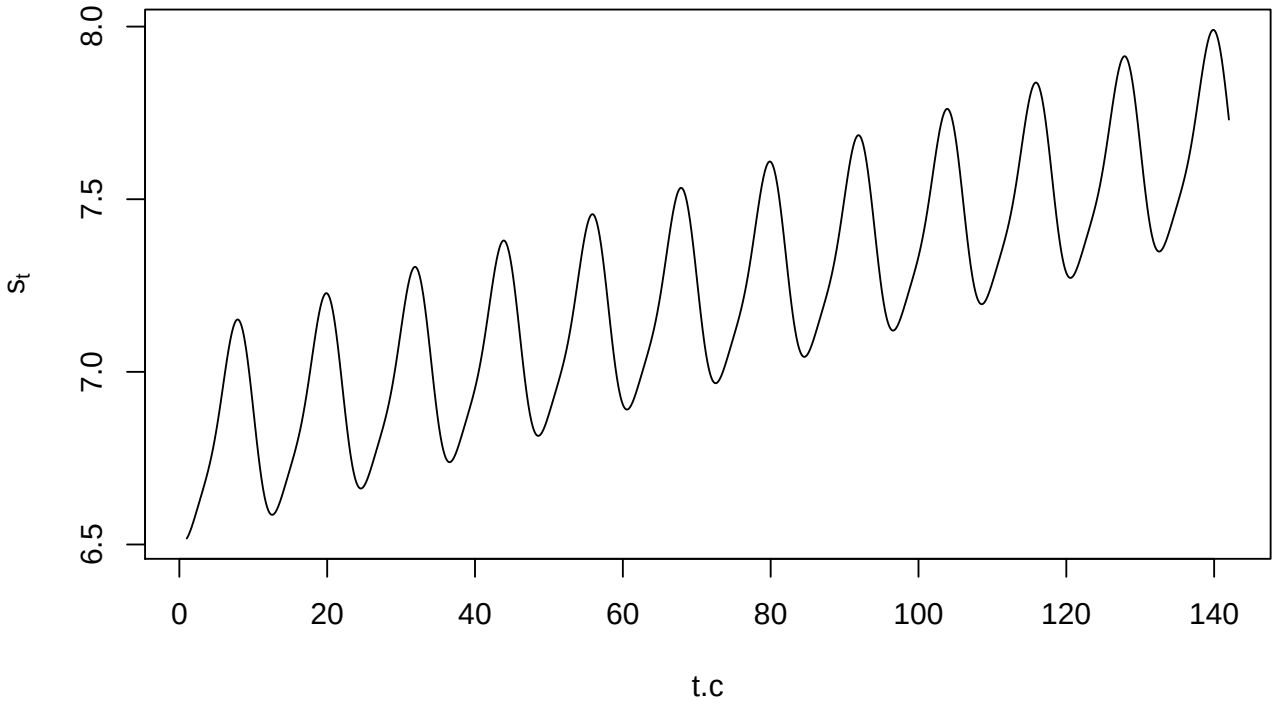
We can fit this model in a straightforward fashion using `lm` in `R`:

```
t.v<-c(1:n);Time<-t.v
C1<-cos(2*pi*t.v/12);S1<-sin(2*pi*t.v/12)
C2<-cos(2*pi*t.v/6);S2<-sin(2*pi*t.v/6)
lWine.data<-data.frame(lWine,Time,S1,C1,S2,C2)
lWine.fit<-lm(lWine~Time+S1+C1+S2+C2,data=lWine.data)
lWine.fitted<-lWine.fit$fitted
lWine.coef<-lWine.fit$coef
par(mar=c(4,4,2,1))
plot(log(Xt),ylab="Monthly wine sales",type='l');points(log(Xt),pch=19,cex=0.5)
title("Monthly wine sales: log scale")
lWineFit<-ts(lWine.fitted, frequency = 12, start = c(1980, 1))
points(lWineFit,pch=3,cex=0.5,col='red');lines(lWineFit,lty=2,col='red')
legend(1980,8.0,c("Observed","Fitted"),lty=c(1,2),pch=19,col=c('black','red'))
```



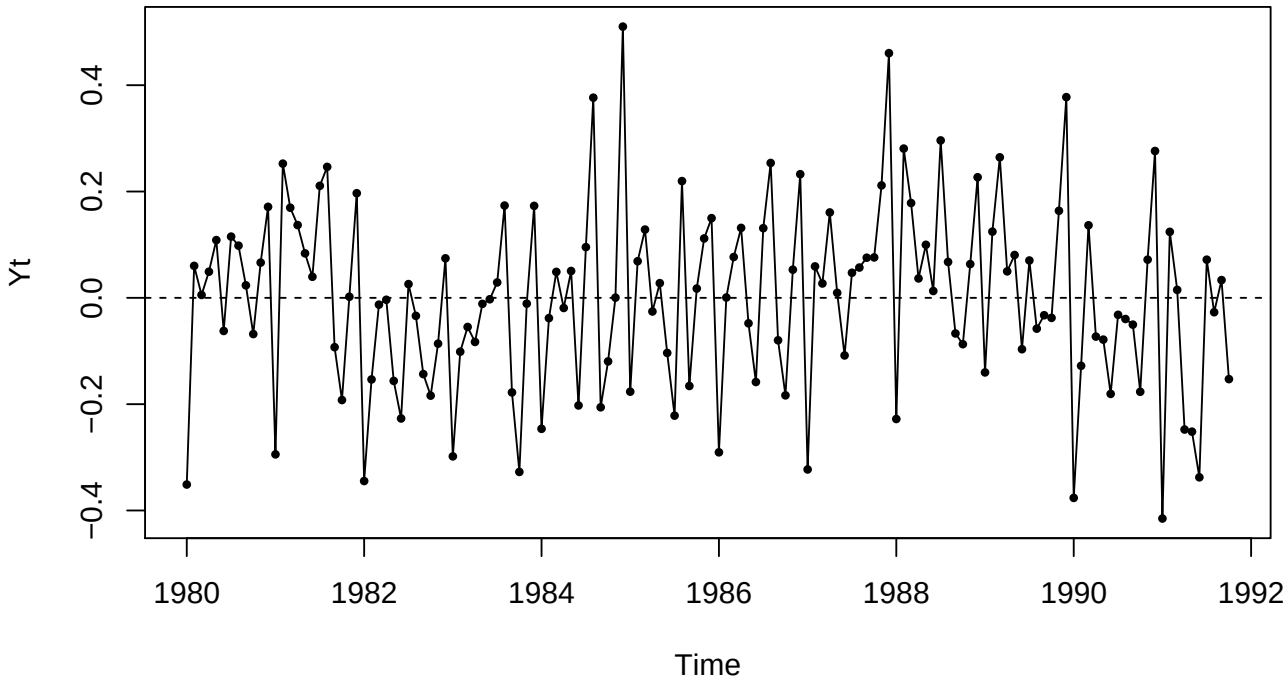
To inspect the fitted m_t:

```
t.c<-seq(from=1,to=142,by=0.01);T.c<-t.c
C1.c<-cos(2*pi*t.c/12);S1.c<-sin(2*pi*t.c/12)
C2.c<-cos(2*pi*t.c/6);S2.c<-sin(2*pi*t.c/6)
y.c<-cbind(1,T.c,C1.c,S1.c,C2.c,S2.c) %*% lWine.coef
par(mar=c(4,4,2,1))
plot(t.c,y.c,type="l",ylab=expression(s[t]))
```



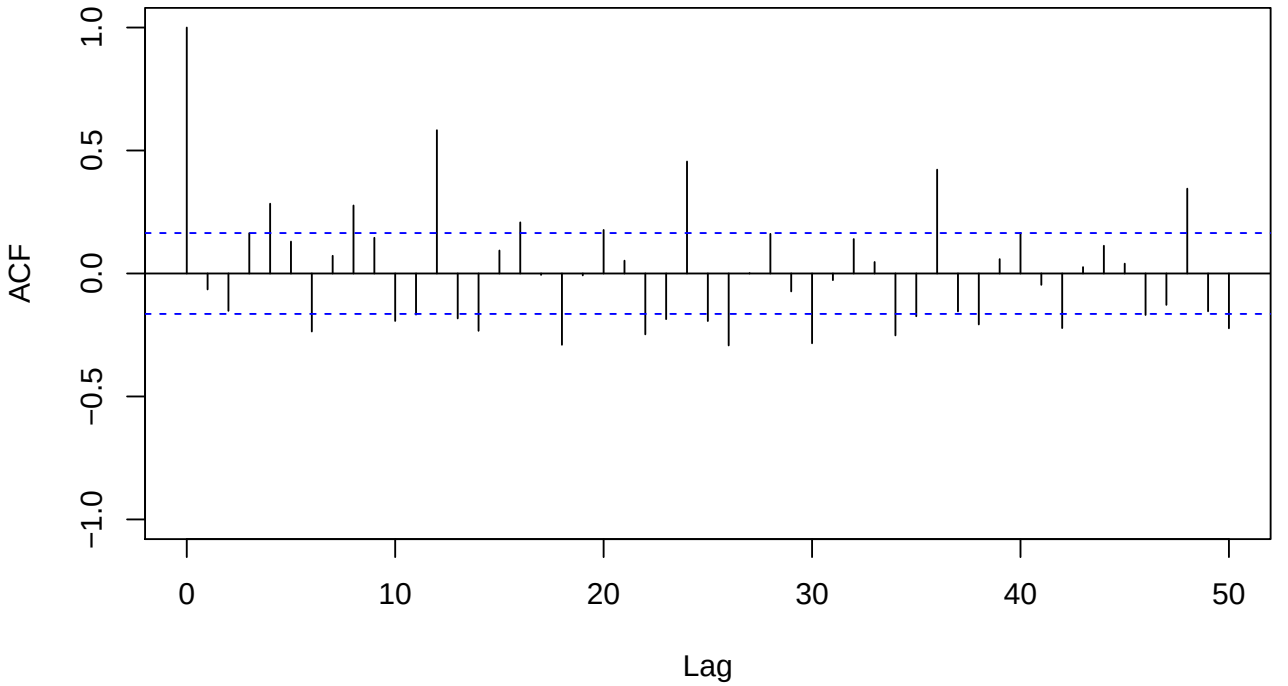
We also inspect the new "residual" series $Y_t = \log(X_t) - \widehat{m}_t$

```
lWine.res<-lWine-lWine.fitted
Yt<-ts(lWine.res, frequency = 12, start = c(1980, 1))
par(mar=c(4,4,0,1))
plot(Yt,ylab="Yt");points(Yt,pch=19,cex=0.5);abline(h=0,lty=2)
```



for which the acf is

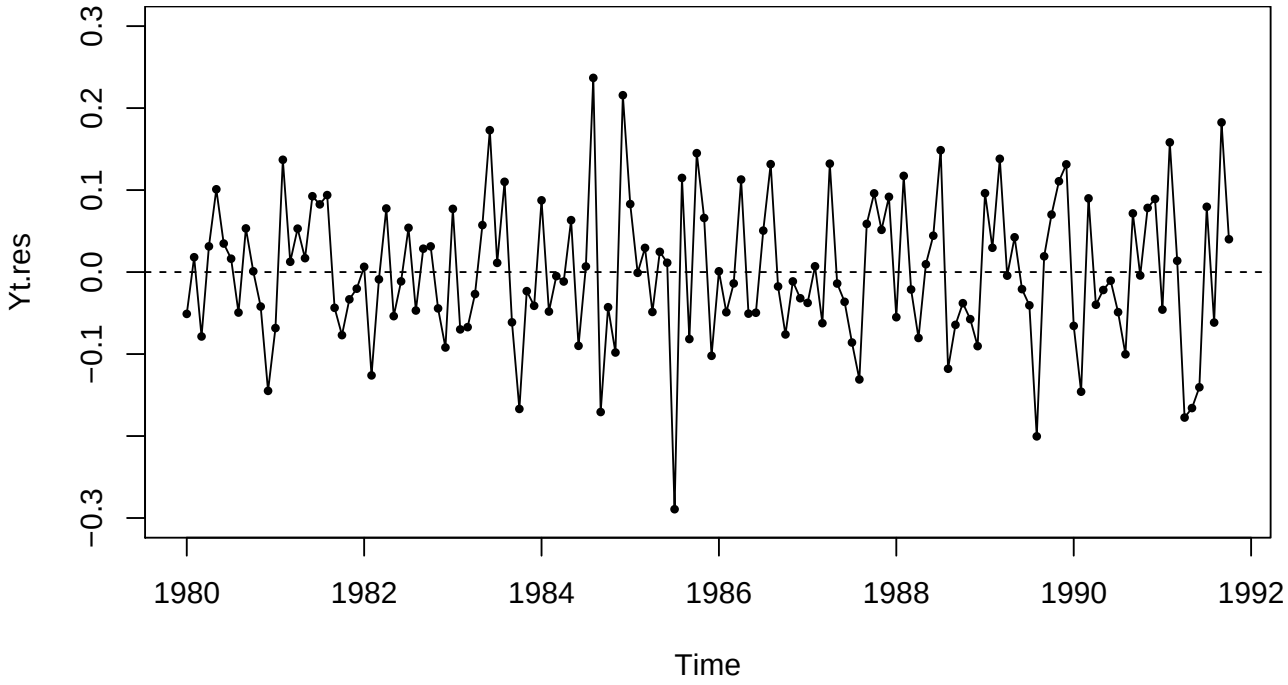
```
par(mar=c(4,4,0,1))
acf(lWine.res,ylim=range(-1,1),lag.max=50)
```



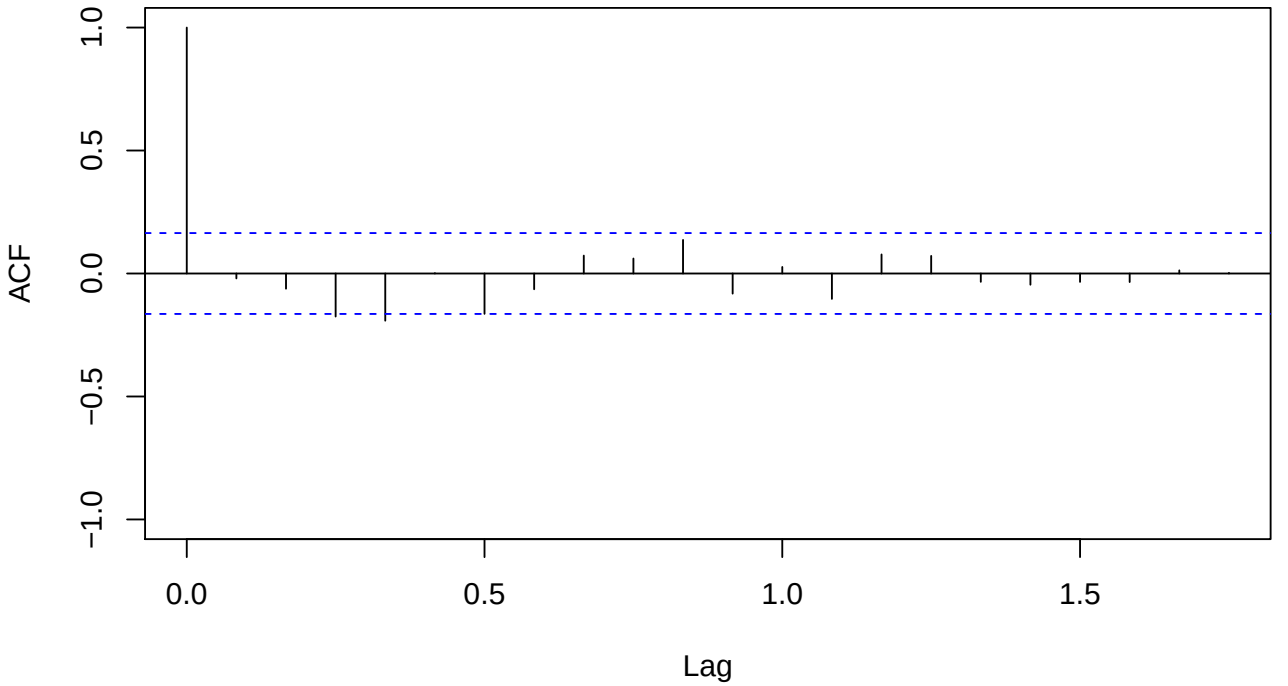
There still seems to be a significant correlation at lags which are multiples of 12, but apart from that the resulting process seems well-represented by a stationary process.

The `stl` function in `R` can be used to complete the decomposition into trend and periodic components in an automatic fashion:

```
tsl.lWine<-stl(ts(lWine,frequency=12,start=c(1980,1)),"per")
Yt<-ts(lWine,frequency=12,start=c(1980,1))
st<-tsl.lWine$time.series[,1]; mt<-tsl.lWine$time.series[,2]
Yt.res<-ts(tsl.lWine$time.series[,3],frequency=12,start=c(1980,1))
par(mar=c(4,4,0,1))
plot(Yt.res,ylim=range(-0.3,0.3));points(Yt.res,pch=19,cex=0.5);abline(h=0,lty=2)
```

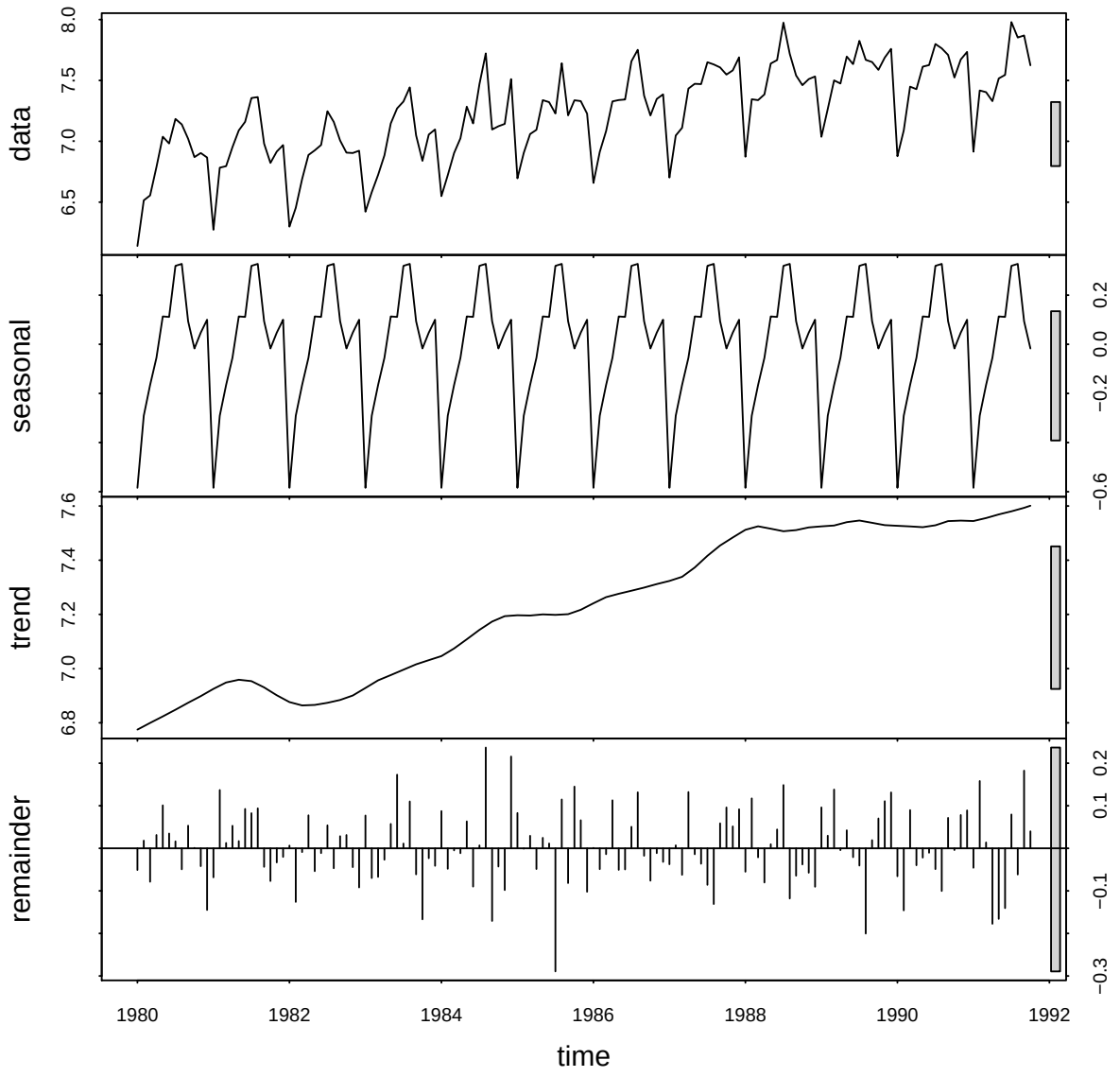



```
par(mar=c(4,4,0,1))
acf(Yt.res,ylim=range(-1,1))
```



A plot of the decomposition into seasonal component and trend can be obtained using the `plot` command in `R`:

```
par(mar=c(4,4,0,1))
plot(tsl.lWine)
```



The fit of the model for m_t is achieved using flexible spline functions that allow for deviation from a simple parametric form. Other forms of flexible modelling can be obtained using

- **weighted (moving) averages**, where m_t is estimated from the data in a time window surrounding t

$$\widehat{m}_t = \frac{1}{2q+1} \sum_{j=-q}^{q} w_j x_{t-j}$$

  for weights $w_j, j = -q, \ldots, q$;
- **exponential smoothing**, based on the one-step ahead procedure

$$\widehat{m}_t = \alpha x_t + (1-\alpha)\widehat{m}_{t-1}$$

  with $m_1 = x_1$, for parameter α $(0 < \alpha < 1)$.

- **Holt-Winters smoothing**: an extension of exponential smoothing that also accounts for trends and seasonalities in the data. This fits the model based on the one step ahead prediction error based on the prediction

$$\widetilde{y}_{t+1} = a_t + b_t + s_{(t-d+1) \mod d}$$

where

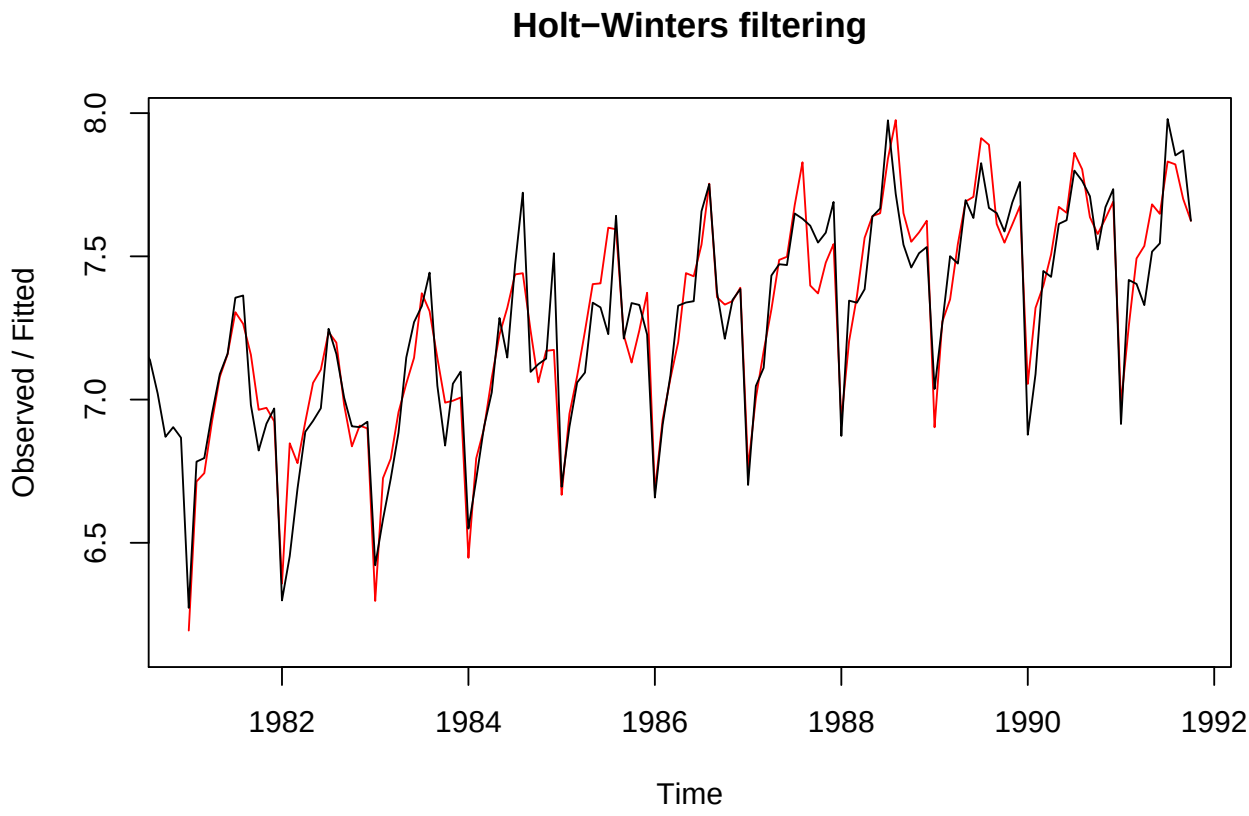
$$a_t = \alpha(y_t - s_{t-d}) + (1-\alpha)(a_{t-1} - b_{t-1})$$
$$b_t = \beta(a_t - a_{t-1}) + (1-\beta)b_{t-1}$$
$$s_t = \gamma(y_t - a_t) + (1-\gamma)s_{t-d}$$

for parameters α, β, γ to be estimated. It can be fitted using the `HoltWinters` function in `R`:

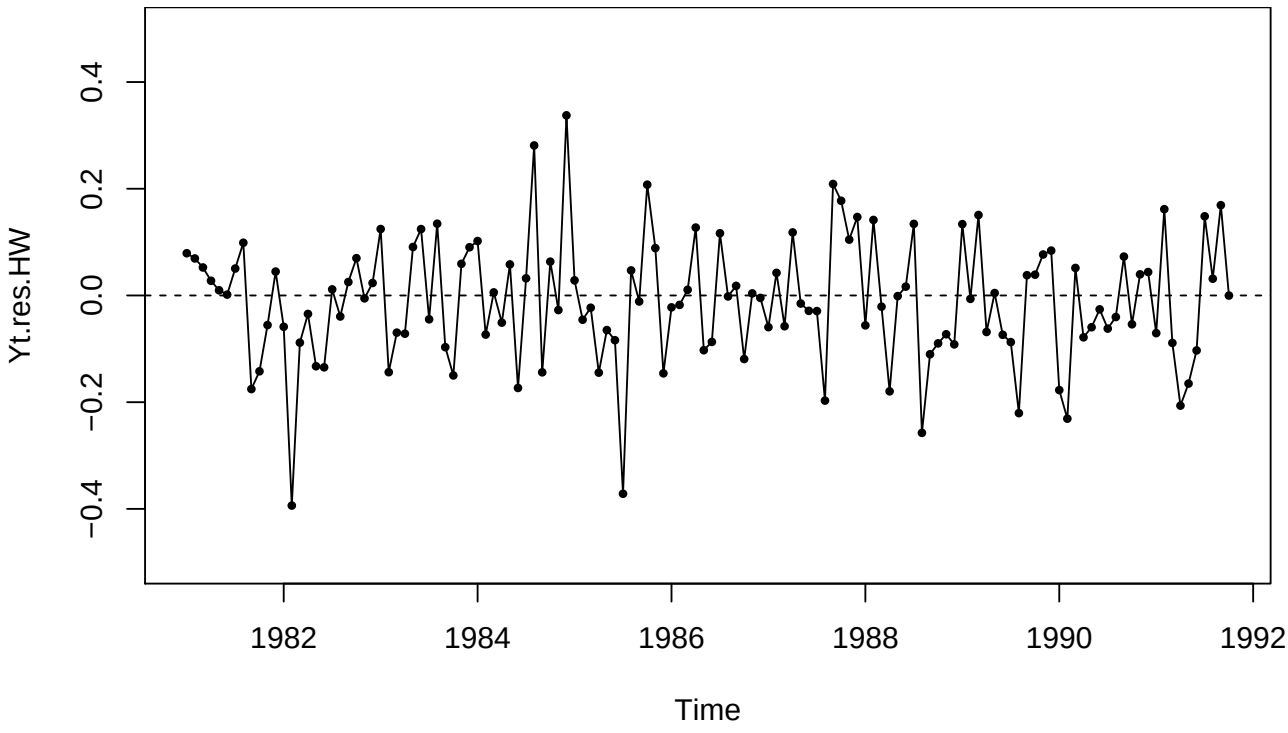
```
(m <- HoltWinters(Yt))

+ Holt-Winters exponential smoothing with trend and additive seasonal component.
+
+ Call:
+ HoltWinters(x = Yt)
+
+ Smoothing parameters:
+  alpha: 0.199782
+  beta : 0.01121394
+  gamma: 0.359314
+
+ Coefficients:
+             [,1]
+ a    7.654880596
+ b    0.006788425
+ s1   0.054390182
+ s2   0.099139876
+ s3  -0.654899960
+ s4  -0.310388773
+ s5  -0.185248194
+ s6  -0.164760271
+ s7   0.027097028
+ s8   0.038562492
+ s9   0.307969137
+ s10  0.228436103
+ s11  0.134917130
+ s12 -0.030679174

plot(m)
```




```
Yt.res.HW<-m$x-m$fitted[,1]
par(mar=c(4,4,0,1))
plot(Yt.res.HW,ylim=range(-0.5,0.5));points(Yt.res.HW,pch=19,cex=0.5);abline(h=0,lty=2)
```



```
plot(fitted(m))
```

