STAT4870 Chapter 3

Anton Yang

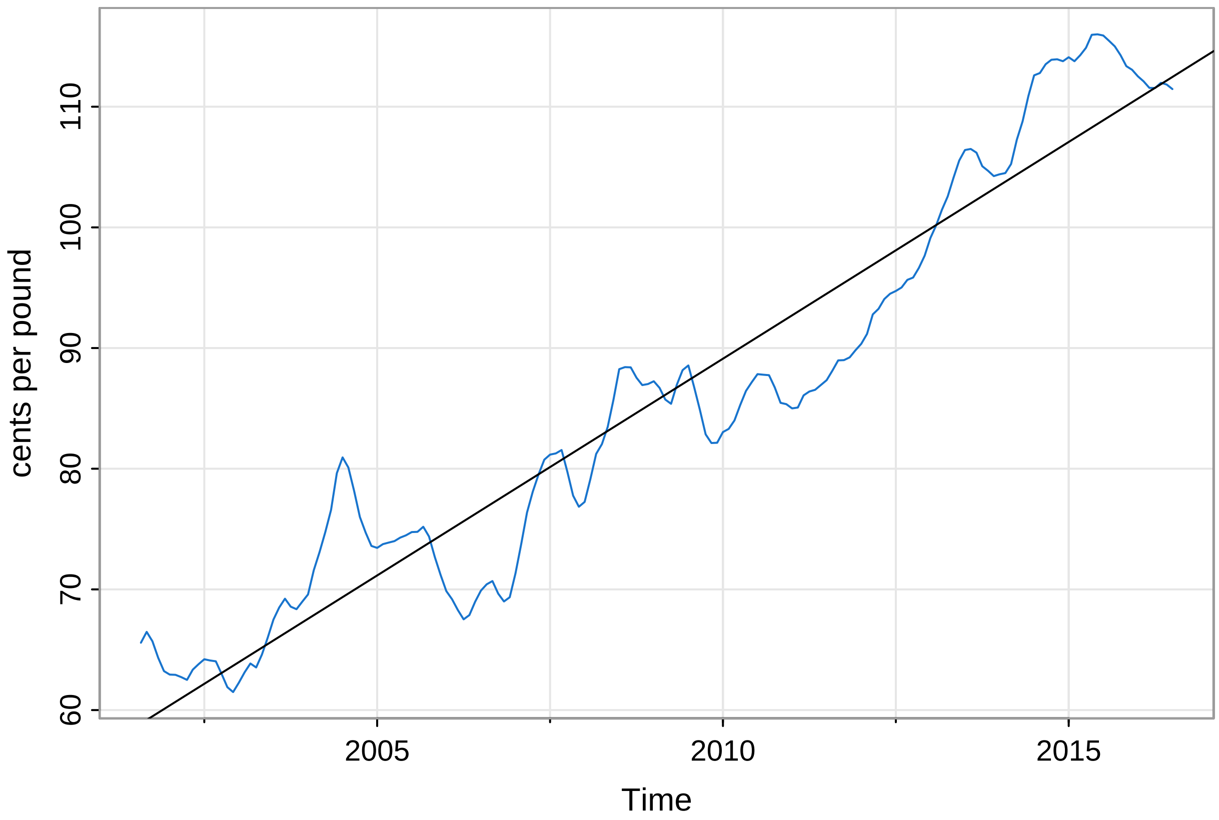
2024-09-15

3.1 Ordinary Least Squares

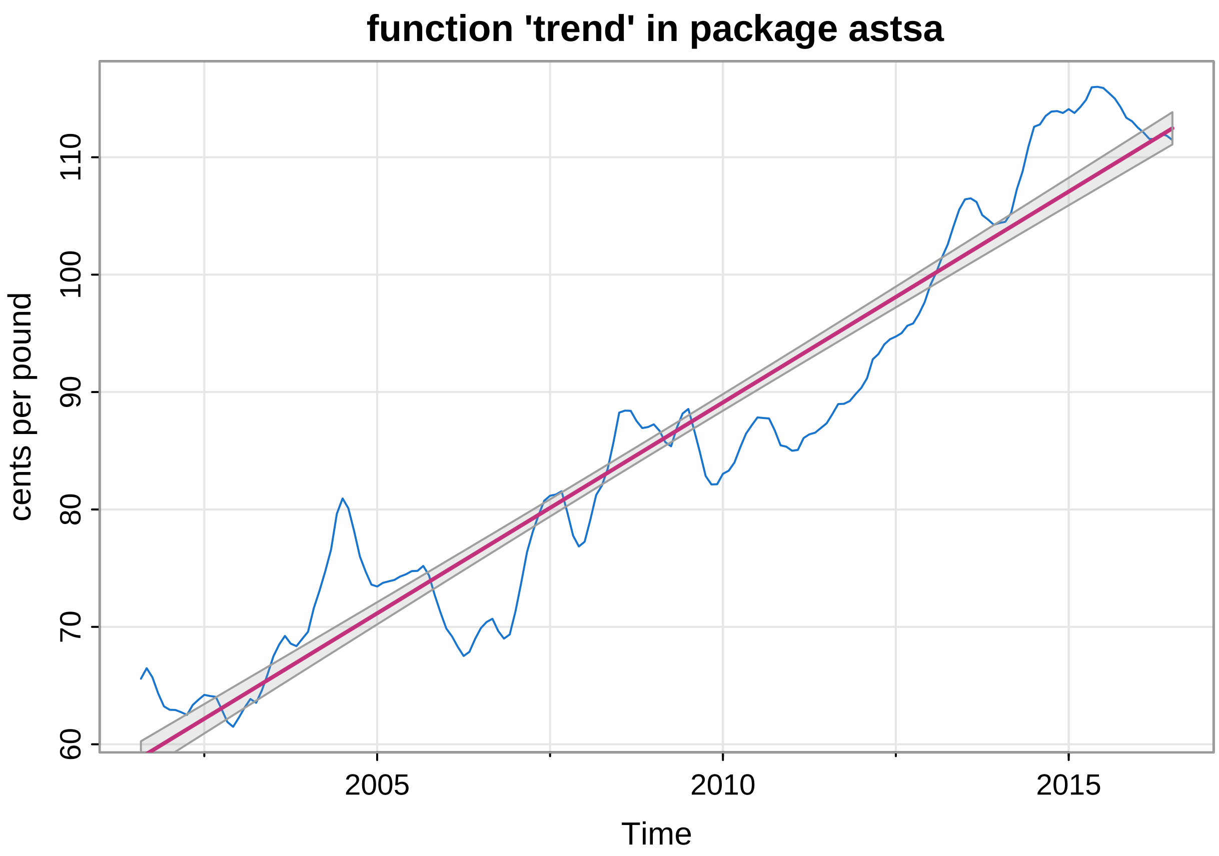
```
library(astsa)
data(chicken)
summary(fit <- lm(chicken~time(chicken),na.action=NULL))
```

```
##
## Call:
## lm(formula = chicken ~ time(chicken), na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7411 -3.4730  0.8251  2.7738 11.5804
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.131e+03  1.624e+02  -43.91   <2e-16 ***
## time(chicken)  3.592e+00  8.084e-02   44.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.696 on 178 degrees of freedom
## Multiple R-squared:  0.9173, Adjusted R-squared:  0.9168
## F-statistic:  1974 on 1 and 178 DF,  p-value: < 2.2e-16
```

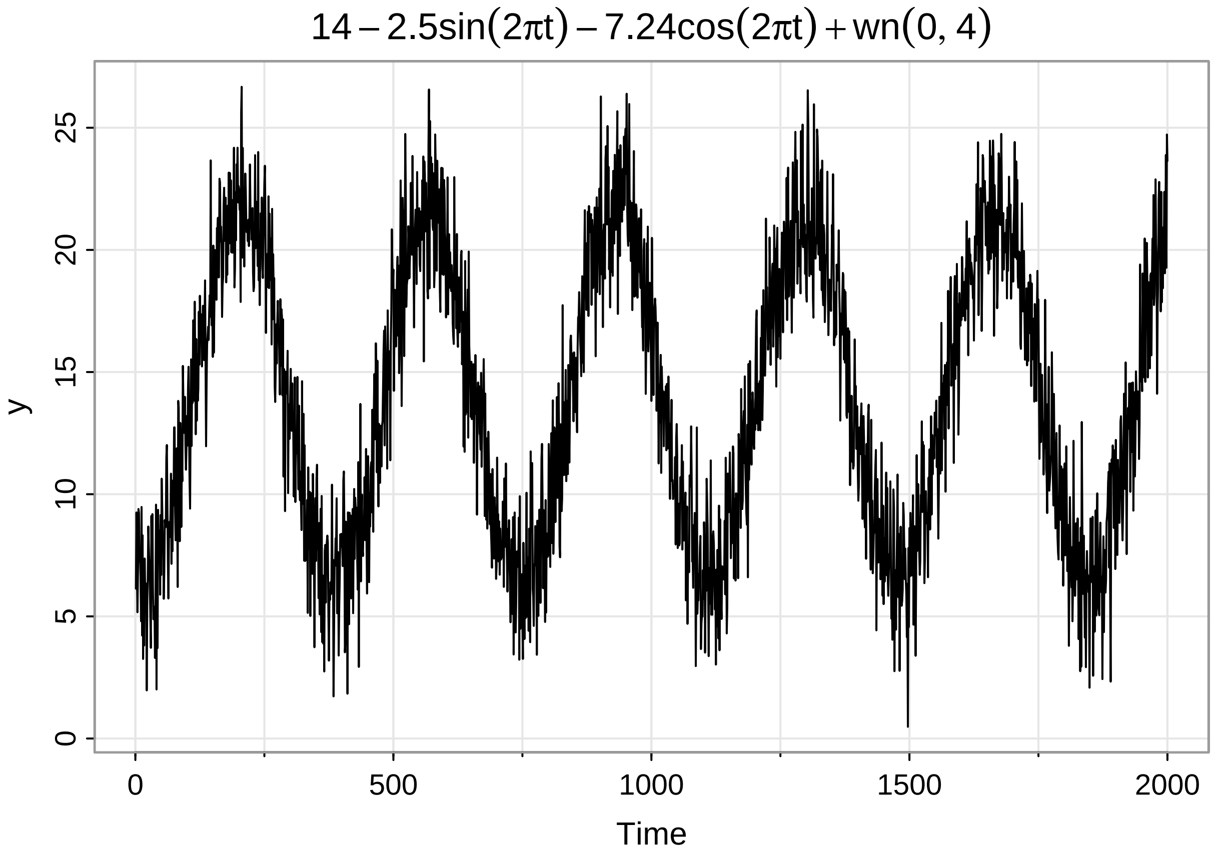
```
tsplot(chicken, ylab="cents per pound",col=4)
abline(fit)
```



```r
trend(chicken, ylab="cents per pound",main="function 'trend' in package astsa")
```



```
n<-2000
t <- (1:n)/365.25
w <- rnorm(n,sd=2)
b0 <- 14;b1<- -2.5;b2<- -7.24
y <- b0+b1*sin(2*pi*t) + b2*cos(2*pi*t) + w
tsplot(y,main=expression(14-2.5*sin(2*pi*t) -7.24*cos(2*pi*t) + wn(0,4)))
```



```
mdl1 <- lm(y~sin(2*pi*t) + cos(2*pi*t))
summary(mdl1)
```

```
##
## Call:
## lm(formula = y ~ sin(2 * pi * t) + cos(2 * pi * t))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3041 -1.3935 -0.0226  1.4093  6.3279
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     14.07675    0.04499  312.87   <2e-16 ***
## sin(2 * pi * t) -2.49032    0.06349  -39.22   <2e-16 ***
## cos(2 * pi * t) -7.18114    0.06356 -112.99   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.005 on 1997 degrees of freedom
## Multiple R-squared:  0.8775, Adjusted R-squared:  0.8774
## F-statistic:  7153 on 2 and 1997 DF,  p-value: < 2.2e-16
```

```
mdl2 <- lm(y~sin(2*pi*t))
summary(mdl2)
```

```
##
## Call:
## lm(formula = y ~ sin(2 * pi * t))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1229  -4.6521   0.0938   4.7211  12.7070
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      14.0486     0.1223  114.87   <2e-16 ***
## sin(2 * pi * t)  -2.4914     0.1726  -14.44   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 5.451 on 1998 degrees of freedom
## Multiple R-squared:  0.09445,    Adjusted R-squared:  0.094
## F-statistic: 208.4 on 1 and 1998 DF,  p-value: < 2.2e-16
```

```r
mdl3 <- lm(y~cos(2*pi*t))
summary(mdl3)
```

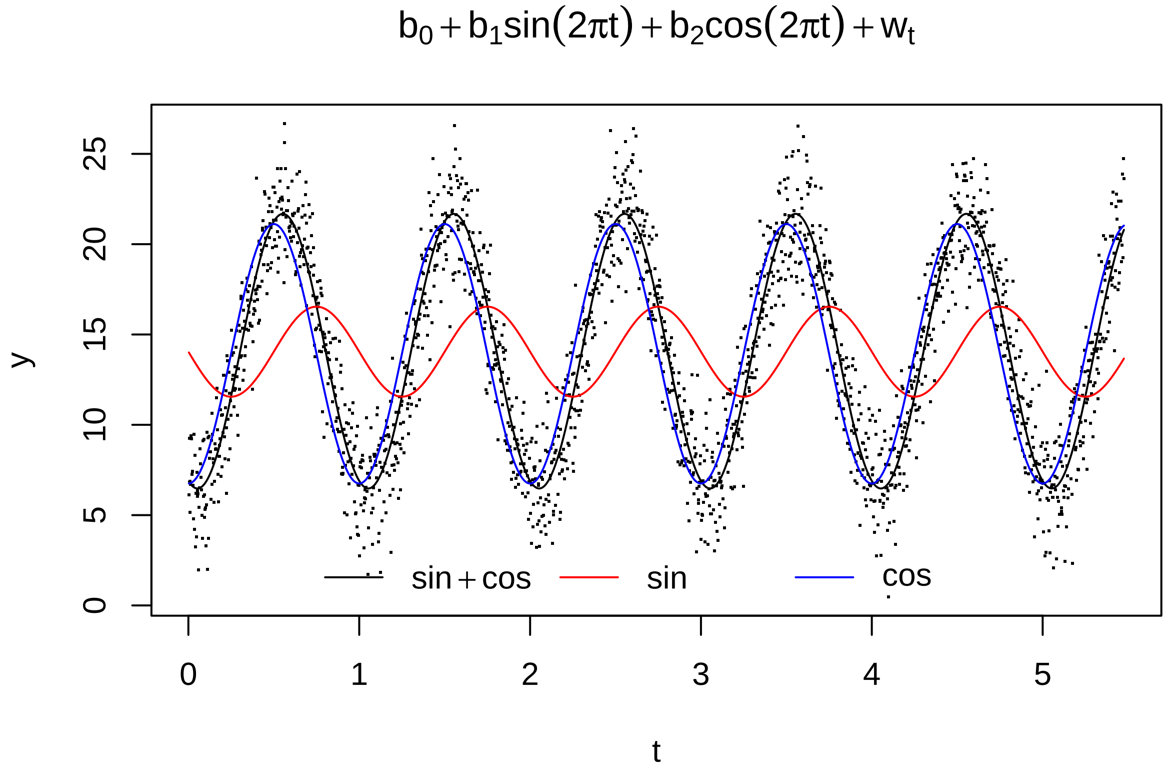
```
##
## Call:
## lm(formula = y ~ cos(2 * pi * t))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3246 -1.8576 -0.0078  1.8710  7.3577
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     13.93274    0.05965  233.57   <2e-16 ***
## cos(2 * pi * t) -7.18150    0.08455  -84.94   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.668 on 1998 degrees of freedom
## Multiple R-squared:  0.7831, Adjusted R-squared:  0.783
## F-statistic:  7215 on 1 and 1998 DF,  p-value: < 2.2e-16
```

```r
merge(AIC(mdl1,mdl2,mdl3),BIC(mdl1,mdl2,mdl3),by='row.names',all=TRUE)
```

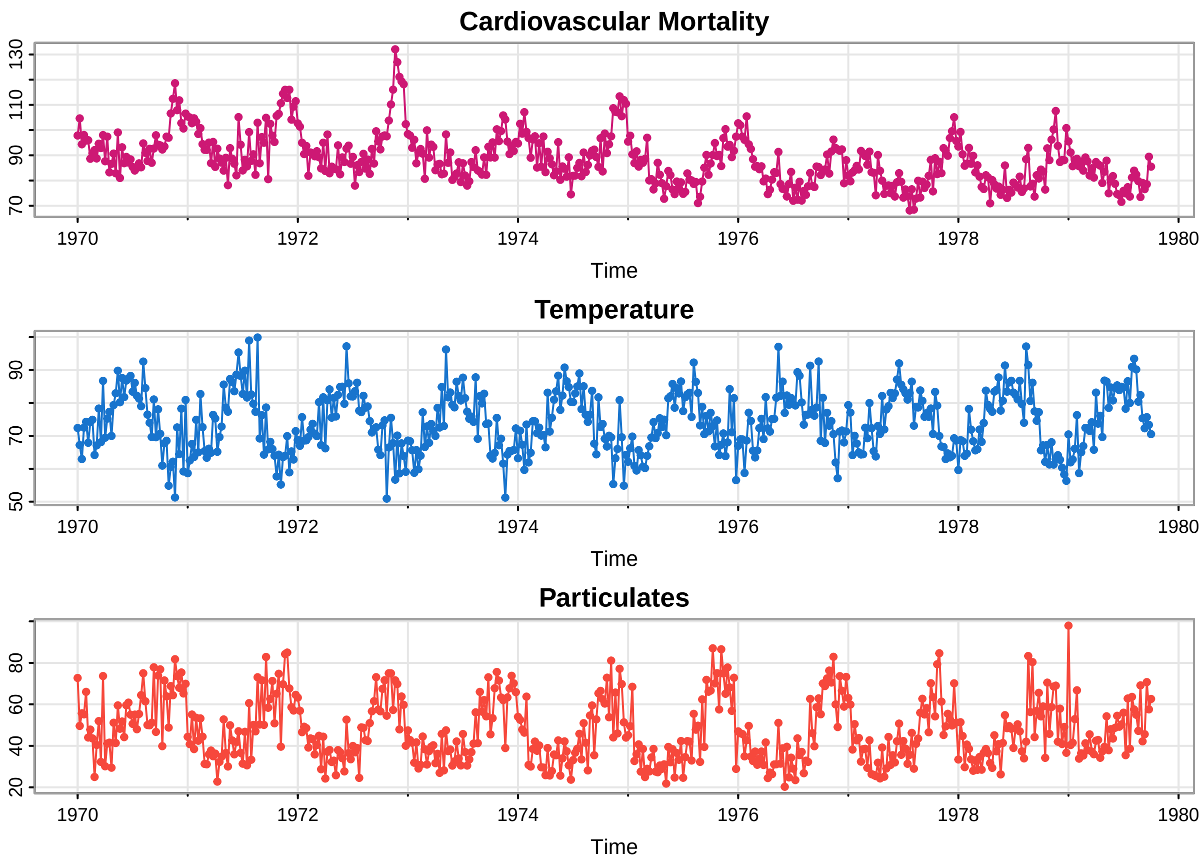
```
##   Row.names df.x       AIC df.y       BIC
## 1      mdl1    4  8464.133    4  8486.536
## 2      mdl2    3 12463.123    3 12479.926
## 3      mdl3    3  9604.563    3  9621.366
```

```r
plot(t,y,pch='.',main=expression(b[0]+b[1]*sin(2*pi*t) + b[2]*cos(2*pi*t) + w[t]))
lines(t,predict(mdl1))
lines(t,predict(mdl2),col='red')
lines(t,predict(mdl3),col='blue')
```

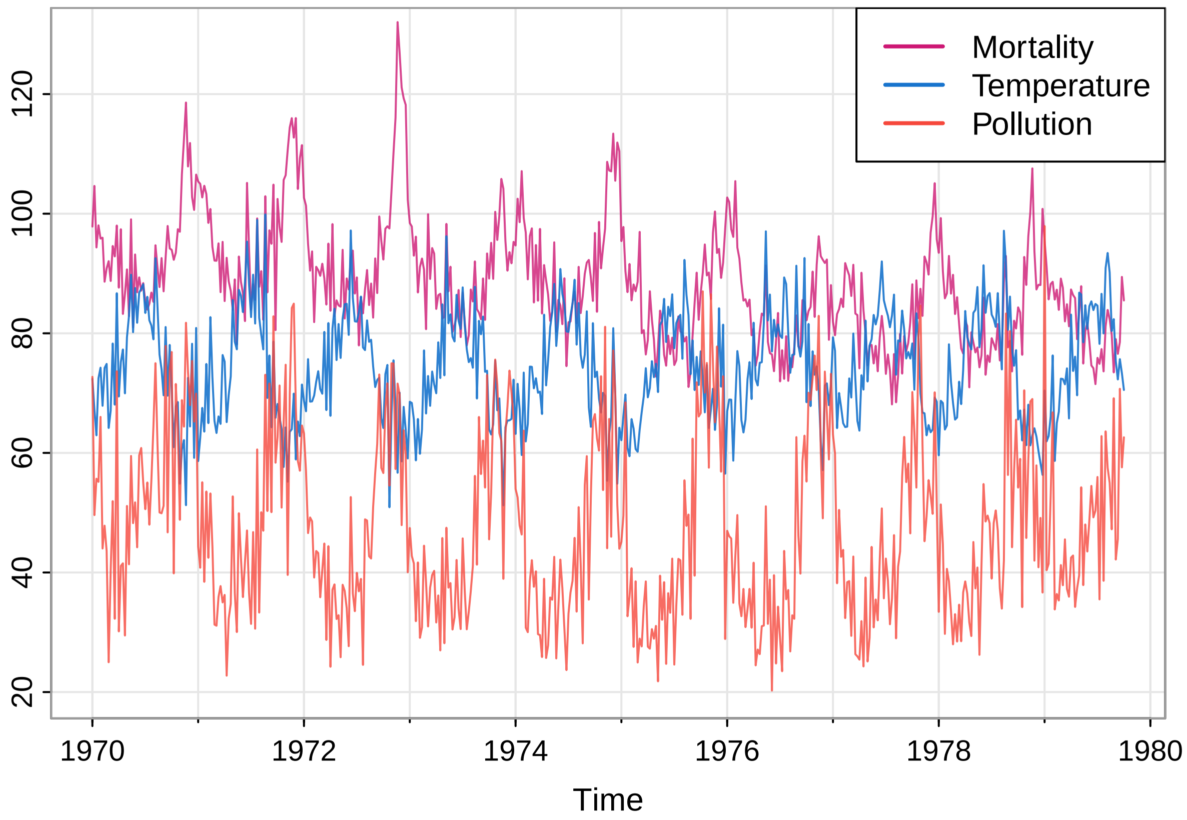
```
legend("bottom",c(expression(sin + cos),expression(sin),expression(cos)),
bty='n',ncol=3,lty=1,col=c("black","red","blue"))
```



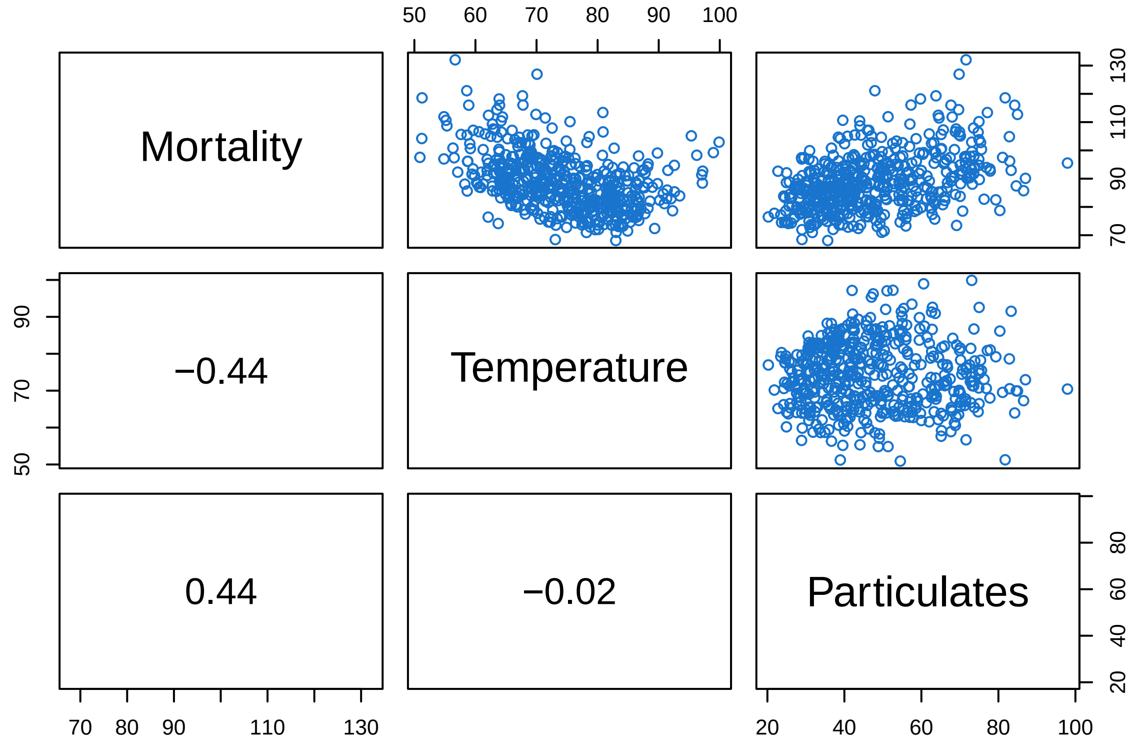
```
op<-par(mfrow=c(3,1))
tsplot(cmort, main="Cardiovascular Mortality", col=6, type="o", pch=19, ylab="")
tsplot(tempr, main="Temperature", col=4, type="o", pch=19, ylab="")
tsplot(part, main="Particulates", col=2, type="o", pch=19, ylab="")
```



```
tsplot(cmort, ylab="", ylim=c(20,130), col=astsa.col(6,.8))
lines(tempr, col=astsa.col(4,.9))
lines(part, col=astsa.col(2,.8))
legend("topright", legend=c("Mortality", "Temperature", "Pollution"), lty=1, lwd=2, col=c(6,4,2), bg="w
```



```
panel.cor <- function(x, y, ...){
par(usr = c(0, 1, 0, 1))
r <- round(cor(x, y), 2)
text(0.5, 0.5, r, cex = 1.75)
}
pairs(cbind(Mortality=cmort, Temperature=tempr, Particulates=part), col=4, lower.panel=panel.cor)
```



```
temp <- tempr-mean(tempr)
temp2 <- temp^2
trend <- time(cmort)
fit <- lm(cmort~ trend + temp + temp2 + part, na.action=NULL)
summary(fit)
```

```
##
## Call:
## lm(formula = cmort ~ trend + temp + temp2 + part, na.action = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.0760  -4.2153  -0.4878   3.7435  29.2448
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.831e+03  1.996e+02   14.19  < 2e-16 ***
## trend       -1.396e+00  1.010e-01  -13.82  < 2e-16 ***
## temp        -4.725e-01  3.162e-02  -14.94  < 2e-16 ***
## temp2        2.259e-02  2.827e-03    7.99 9.26e-15 ***
## part         2.554e-01  1.886e-02   13.54  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.385 on 503 degrees of freedom
## Multiple R-squared:  0.5954, Adjusted R-squared:  0.5922
## F-statistic:   185 on 4 and 503 DF,  p-value: < 2.2e-16
```

```
summary(aov(fit))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)    
## trend         1  10667   10667  261.62 <2e-16 ***
## temp          1   8607    8607  211.09 <2e-16 ***
## temp2         1   3429    3429   84.09 <2e-16 ***
## part          1   7476    7476  183.36 <2e-16 ***
## Residuals   503  20508      41                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(lm(cmort~cbind(trend, temp, temp2, part))))
```

```
##                                  Df Sum Sq Mean Sq F value Pr(>F)    
## cbind(trend, temp, temp2, part)   4  30178    7545     185 <2e-16 ***
## Residuals                       503  20508      41                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
num <- length(cmort)
AIC(fit)/num - log(2*pi)
```

```
## [1] 4.721732
```

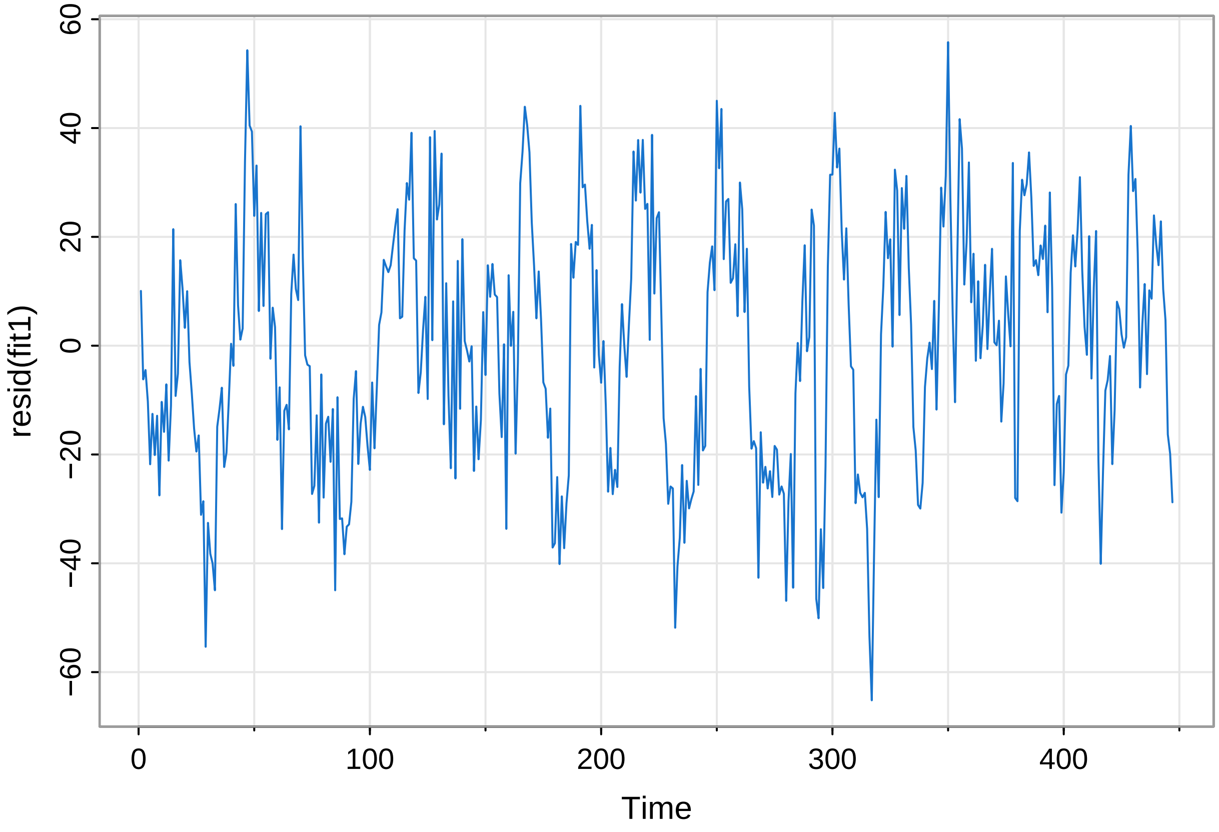
```
BIC(fit)/num - log(2*pi)
```

```
## [1] 4.771699
```

```
fish <- ts.intersect( rec, soiL6=lag(soi,-6) )
summary(fit1 <- lm(rec~ soiL6, data=fish, na.action=NULL))
```

```
## 
## Call:
## lm(formula = rec ~ soiL6, data = fish, na.action = NULL)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -65.187 -18.234   0.354  16.580  55.790 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   65.790      1.088   60.47   <2e-16 ***
## soiL6        -44.283      2.781  -15.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 22.5 on 445 degrees of freedom
## Multiple R-squared:  0.3629, Adjusted R-squared:  0.3615 
## F-statistic: 253.5 on 1 and 445 DF,  p-value: < 2.2e-16
```

```
tsplot(resid(fit1), col=4)
```



```
library(dynlm)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

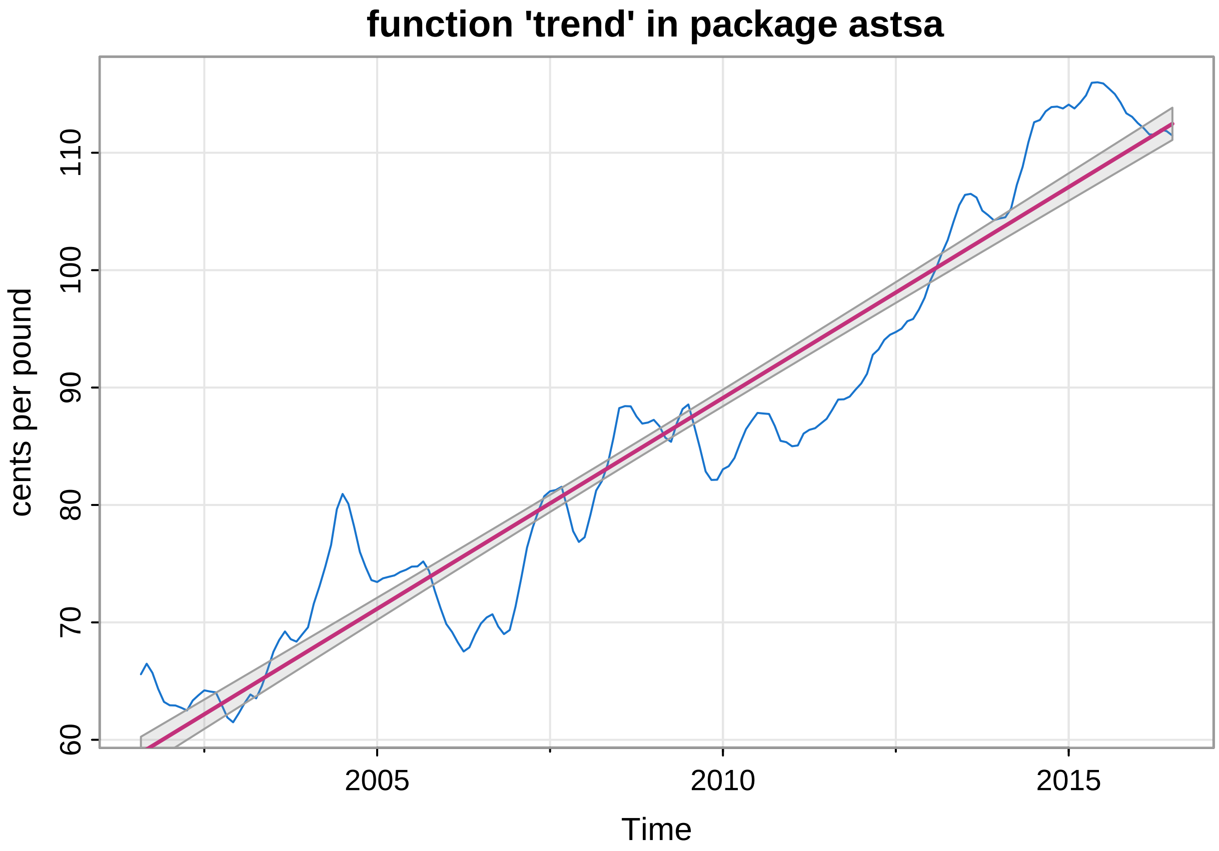
```
summary(fit2 <- dynlm(rec~ L(soi,6)))
```

```
##
## Time series regression with "ts" data:
## Start = 1950(7), End = 1987(9)
##
## Call:
## dynlm(formula = rec ~ L(soi, 6))
##
## Residuals:
```

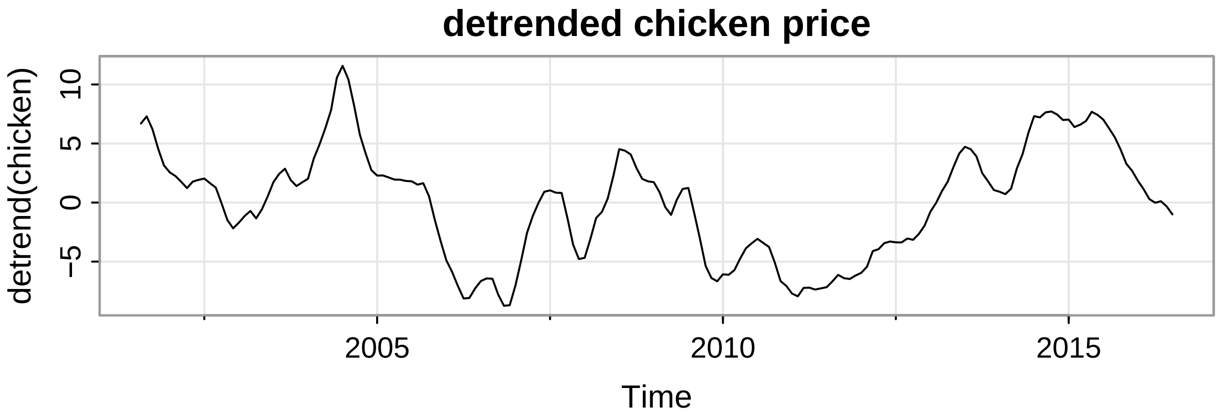
```
##      Min       1Q  Median      3Q     Max
## -65.187 -18.234   0.354  16.580  55.790
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   65.790      1.088   60.47   <2e-16 ***
## L(soi, 6)    -44.283      2.781  -15.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.5 on 445 degrees of freedom
## Multiple R-squared:  0.3629, Adjusted R-squared:  0.3615
## F-statistic: 253.5 on 1 and 445 DF,  p-value: < 2.2e-16
```

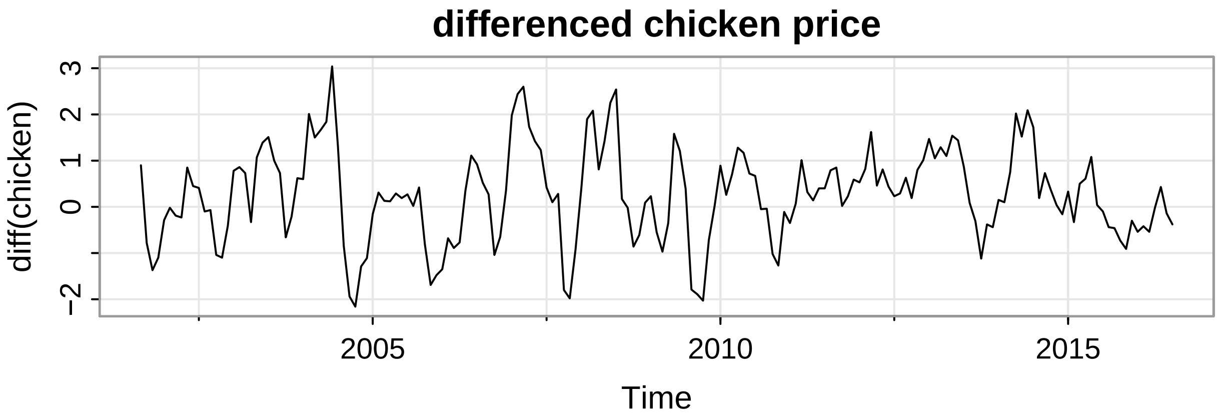
Explanatory Data Analysis

```
library(astsa)
data(chicken)
trend(chicken, ylab="cents per pound",main="function 'trend' in package astsa")
```

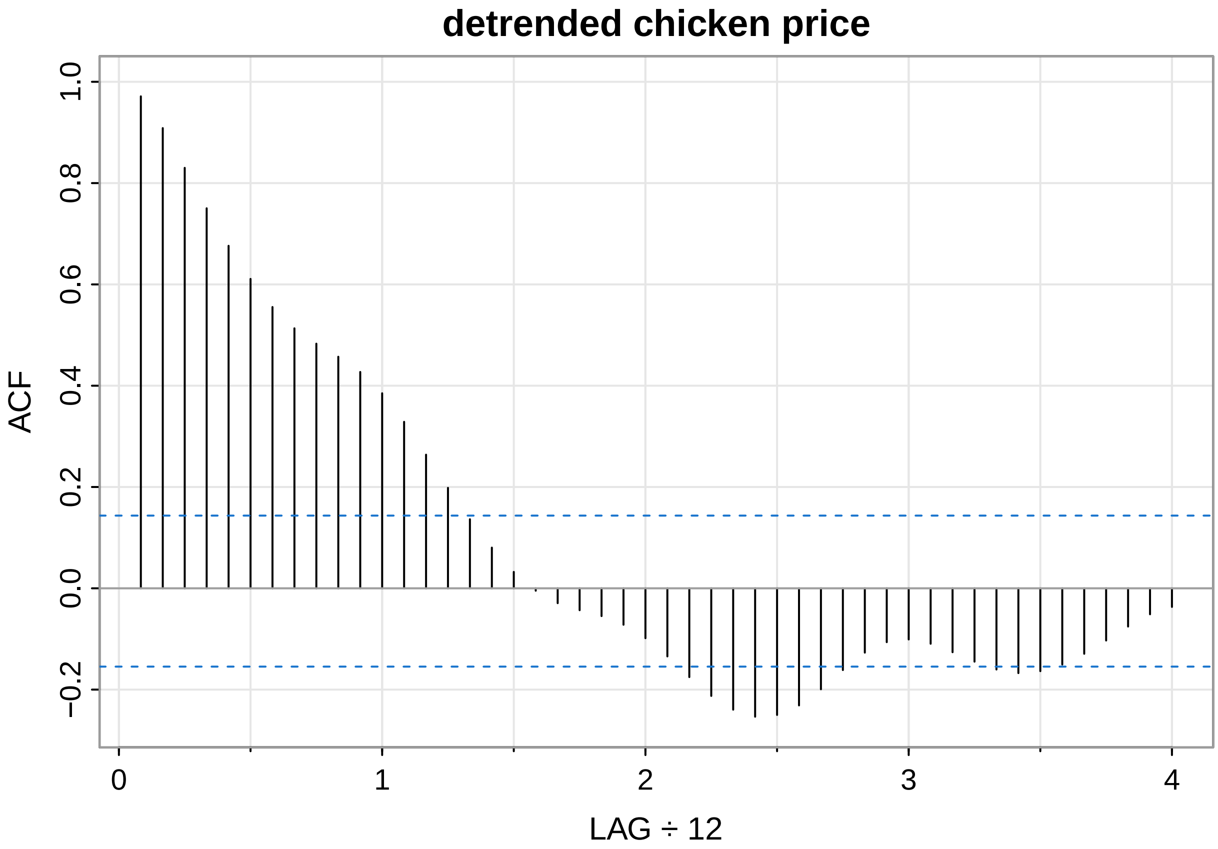


```
data(chicken)
par(mfrow=2:1) # plot transformed data
tsplot(detrend(chicken), main="detrended chicken price" )
tsplot(diff(chicken), main="differenced chicken price" )
```



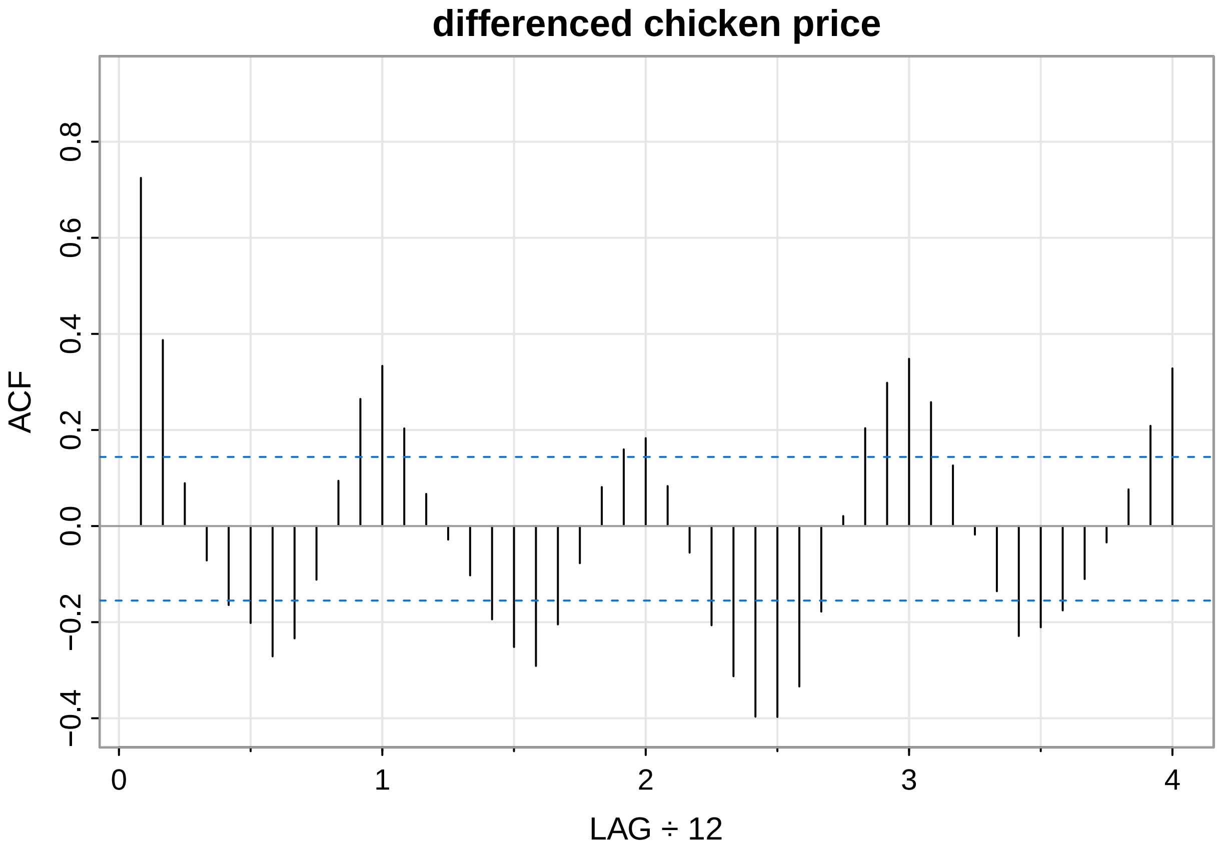


```
acf1(detrend(chicken), main="detrended chicken price")
```



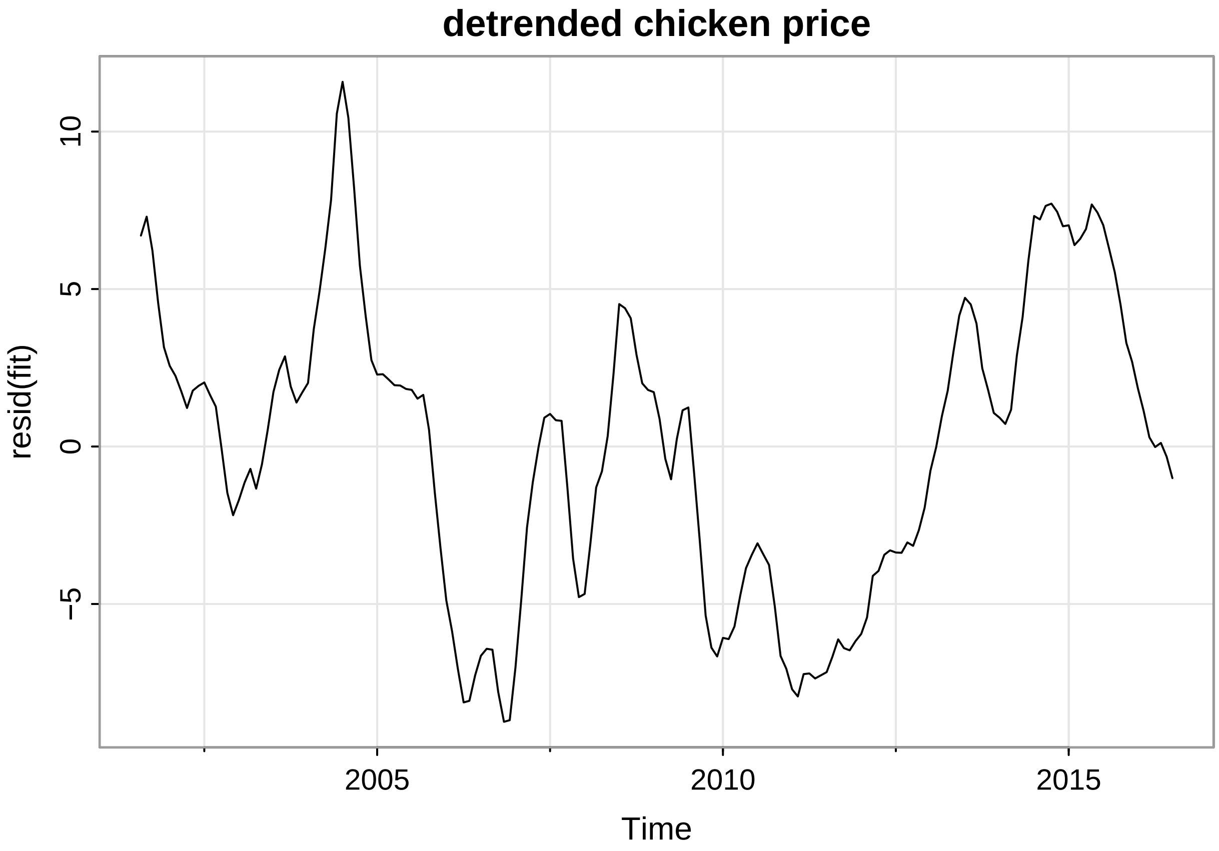
```
##  [1]  0.97  0.91  0.83  0.75  0.68  0.61  0.56  0.51  0.48  0.46  0.43  0.39
## [13]  0.33  0.26  0.20  0.14  0.08  0.03  0.00 -0.03 -0.04 -0.05 -0.07 -0.10
## [25] -0.13 -0.18 -0.21 -0.24 -0.25 -0.25 -0.23 -0.20 -0.16 -0.13 -0.11 -0.10
## [37] -0.11 -0.13 -0.14 -0.16 -0.17 -0.16 -0.15 -0.13 -0.10 -0.08 -0.05 -0.04
```

```
acf1(diff(chicken), main="differenced chicken price")
```

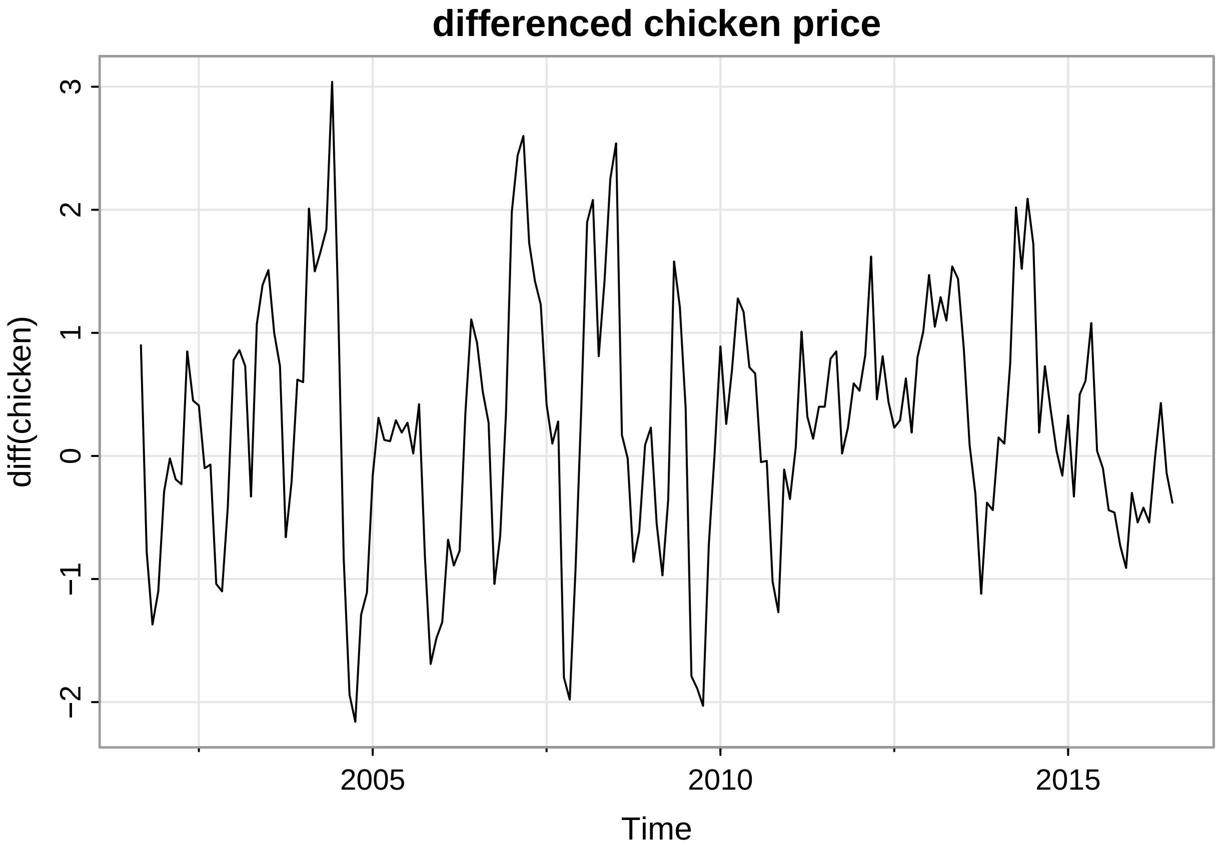


```
##  [1]  0.72  0.39  0.09 -0.07 -0.16 -0.20 -0.27 -0.23 -0.11  0.09  0.26  0.33
## [13]  0.20  0.07 -0.03 -0.10 -0.19 -0.25 -0.29 -0.20 -0.08  0.08  0.16  0.18
## [25]  0.08 -0.06 -0.21 -0.31 -0.40 -0.40 -0.33 -0.18  0.02  0.20  0.30  0.35
## [37]  0.26  0.13 -0.02 -0.14 -0.23 -0.21 -0.18 -0.11 -0.03  0.08  0.21  0.33
```

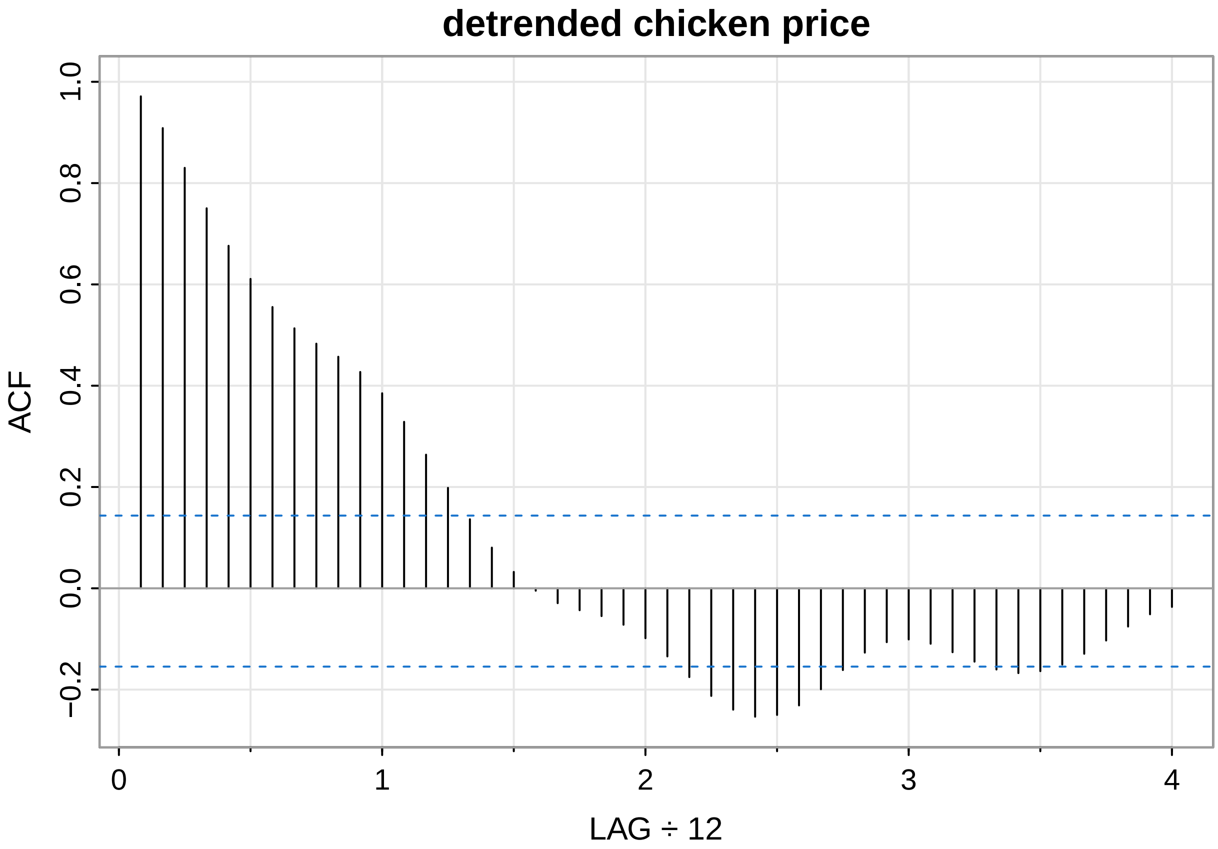
```
fit <- lm(chicken~time(chicken), na.action=NULL) # the regression
tsplot(resid(fit), main="detrended chicken price")
```



```
tsplot(diff(chicken), main="differenced chicken price")
```

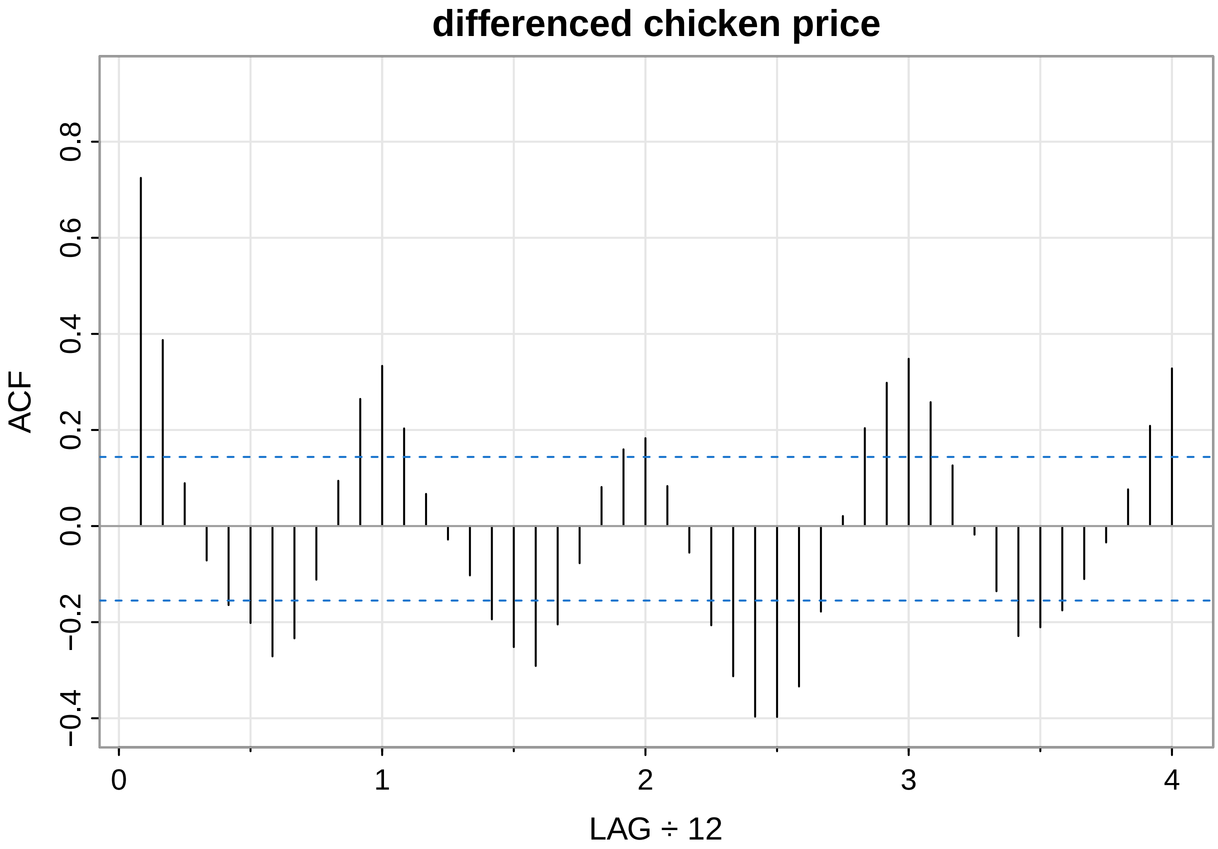


```
acf1(resid(fit), 48, main="detrended chicken price")
```



```
##  [1]  0.97  0.91  0.83  0.75  0.68  0.61  0.56  0.51  0.48  0.46  0.43  0.39
## [13]  0.33  0.26  0.20  0.14  0.08  0.03  0.00 -0.03 -0.04 -0.05 -0.07 -0.10
## [25] -0.13 -0.18 -0.21 -0.24 -0.25 -0.25 -0.23 -0.20 -0.16 -0.13 -0.11 -0.10
## [37] -0.11 -0.13 -0.14 -0.16 -0.17 -0.16 -0.15 -0.13 -0.10 -0.08 -0.05 -0.04
```

```
acf1(diff(chicken), 48, main="differenced chicken price")
```

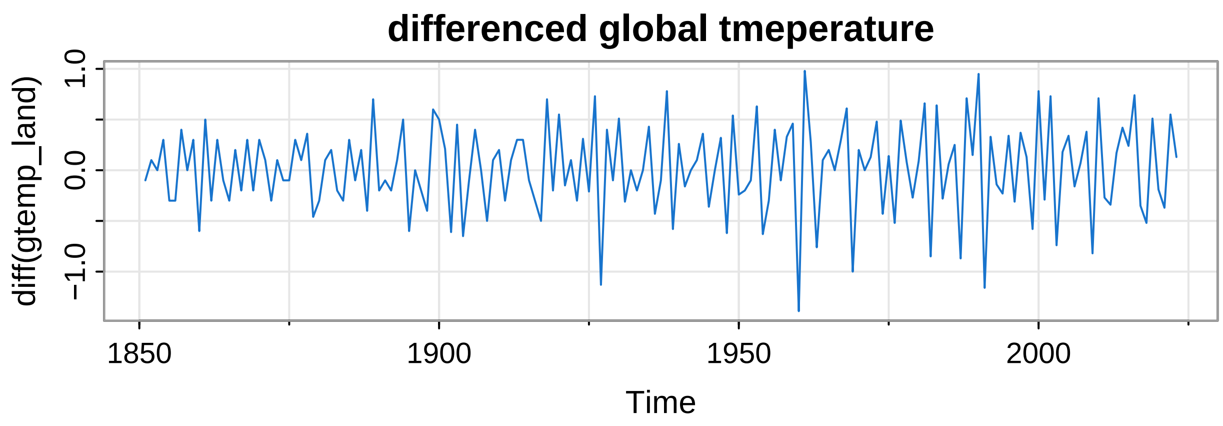


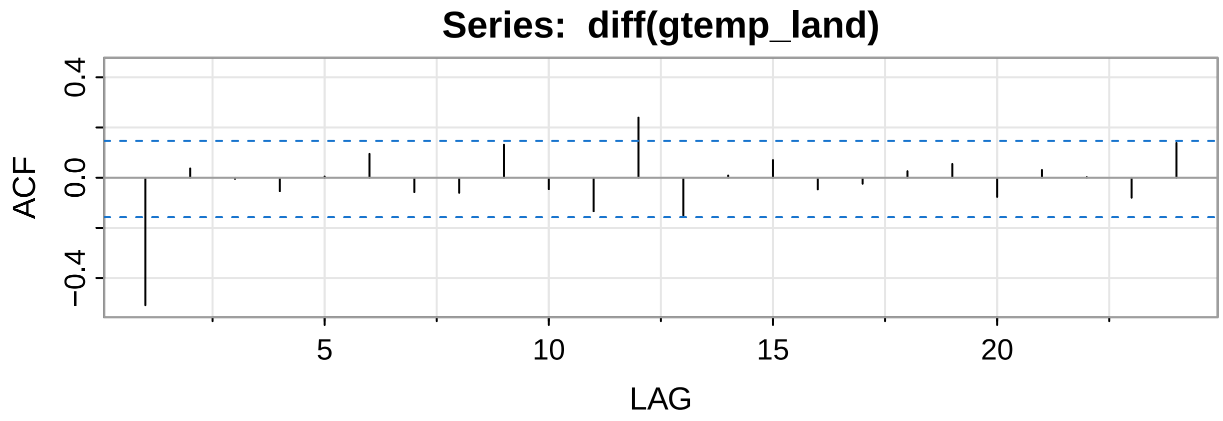
```
##  [1]  0.72  0.39  0.09 -0.07 -0.16 -0.20 -0.27 -0.23 -0.11  0.09  0.26  0.33
## [13]  0.20  0.07 -0.03 -0.10 -0.19 -0.25 -0.29 -0.20 -0.08  0.08  0.16  0.18
## [25]  0.08 -0.06 -0.21 -0.31 -0.40 -0.40 -0.33 -0.18  0.02  0.20  0.30  0.35
## [37]  0.26  0.13 -0.02 -0.14 -0.23 -0.21 -0.18 -0.11 -0.03  0.08  0.21  0.33
```

```
par(mfrow=c(2,1))
tsplot(diff(gtemp_land), col=4, main="differenced global tmeperature")
mean(diff(gtemp_land))
```

```
## [1] 0.01595376
```

```
acf1(diff(gtemp_land))
```





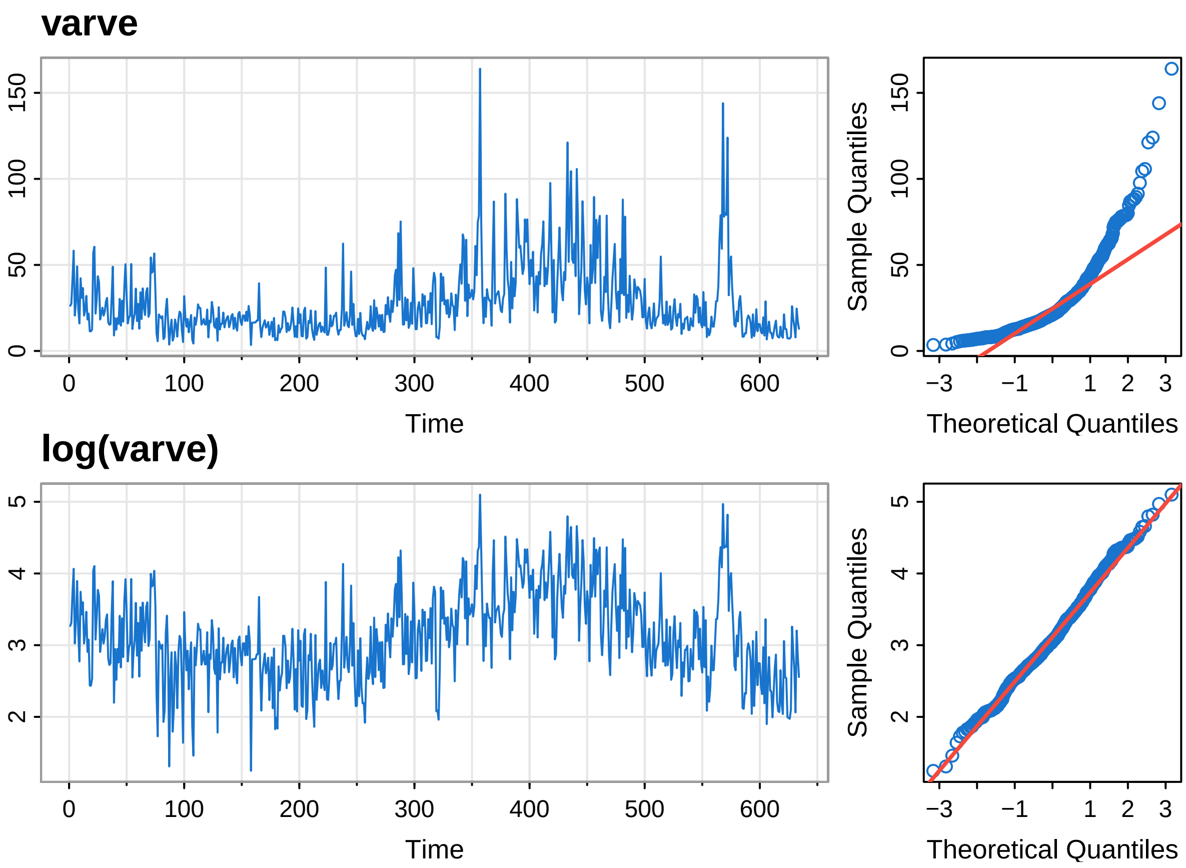
```
##  [1] -0.51  0.04  0.00 -0.05  0.00  0.09 -0.06 -0.06  0.13 -0.05 -0.13  0.24
## [13] -0.15  0.01  0.07 -0.05 -0.02  0.03  0.05 -0.08  0.03  0.00 -0.08  0.14
```

```
mean(window(diff(gtemp_land), start=1980))
```

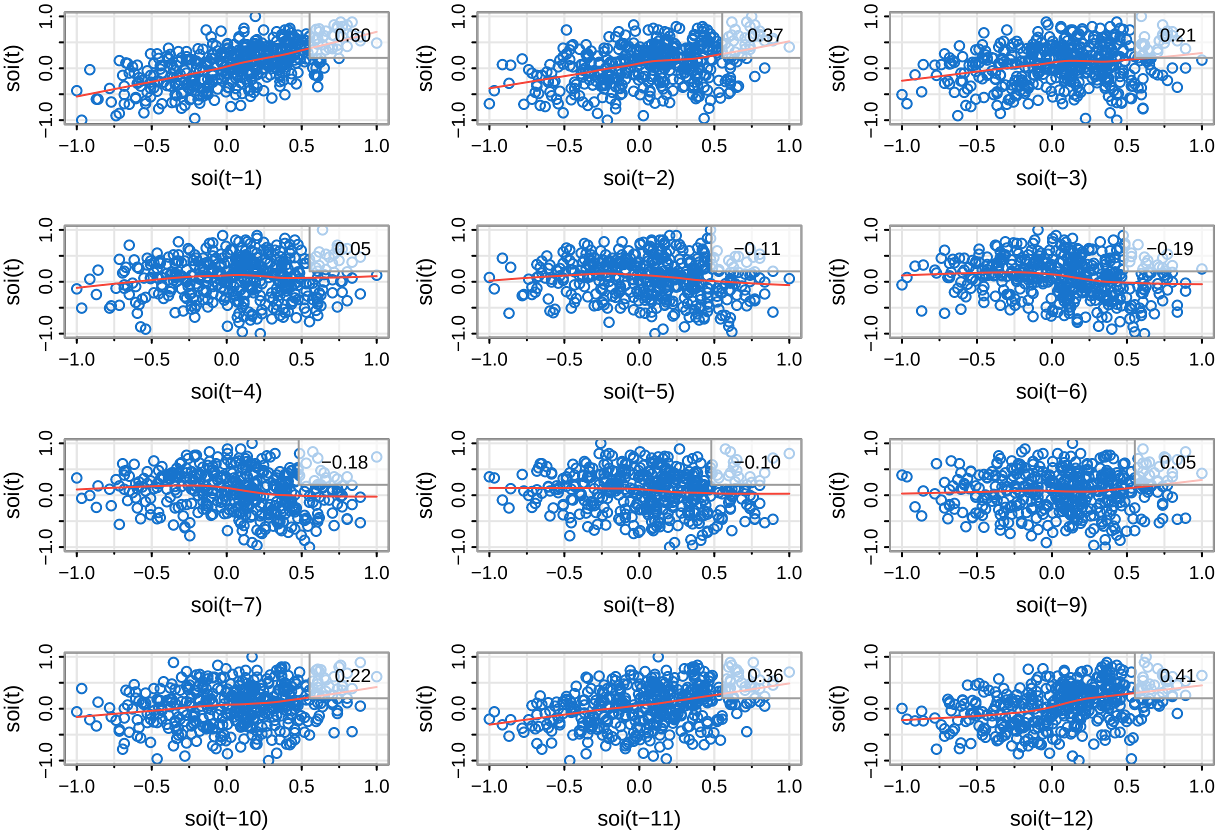
```
## [1] 0.04909091
```

```
par(mfrow=c(1,1))
```

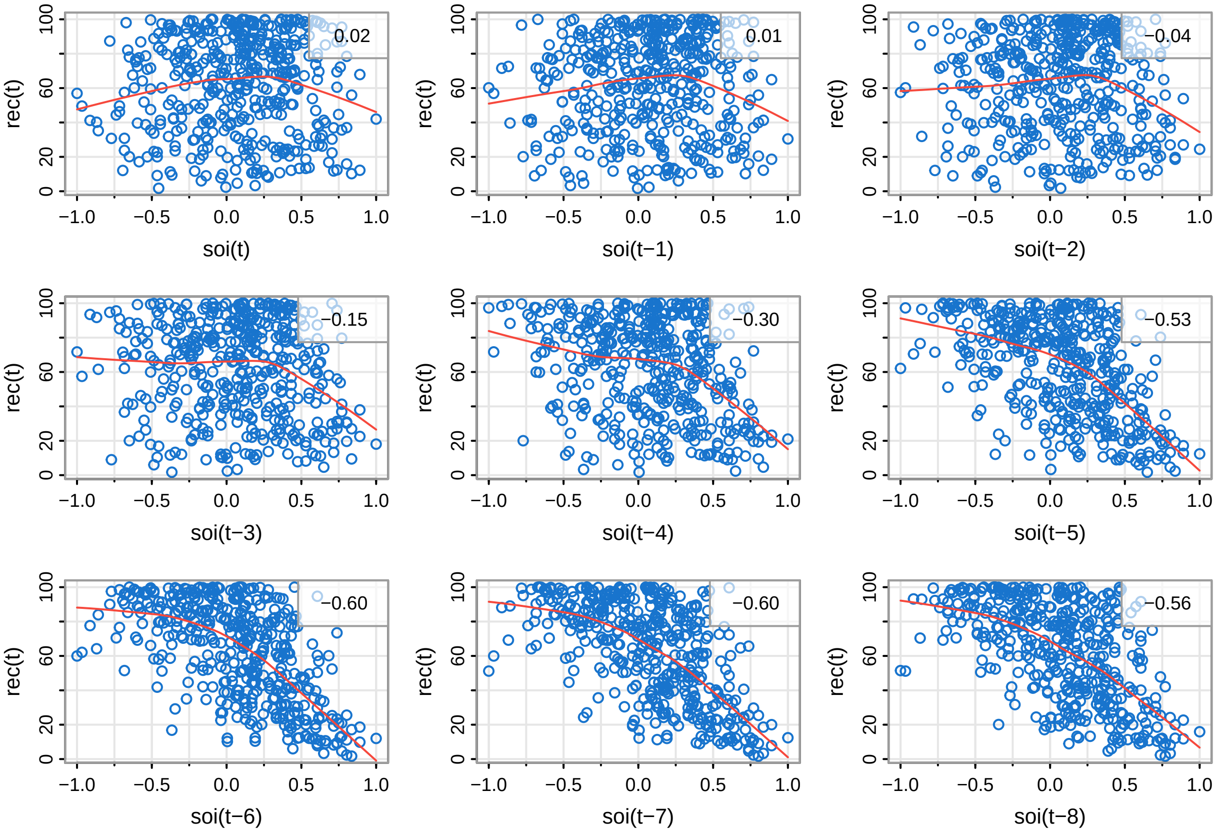
```
layout(matrix(1:4,2), widths=c(2.5,1))
par(oma=rep(.2, 4))
tsplot(varve, main="", ylab="", col=4, margin=0)
mtext("varve", side=3, line=.5, cex=1.2, font=2, adj=0)
tsplot(log(varve), main="", ylab="", col=4, margin=0)
mtext("log(varve)", side=3, line=.5, cex=1.2, font=2, adj=0)
qqnorm(varve, main="", col=4); qqline(varve, col=2, lwd=2)
qqnorm(log(varve), main="", col=4); qqline(log(varve), col=2, lwd=2)
```



```r
lag1.plot(soi, 12, col=4, cex=1)
```



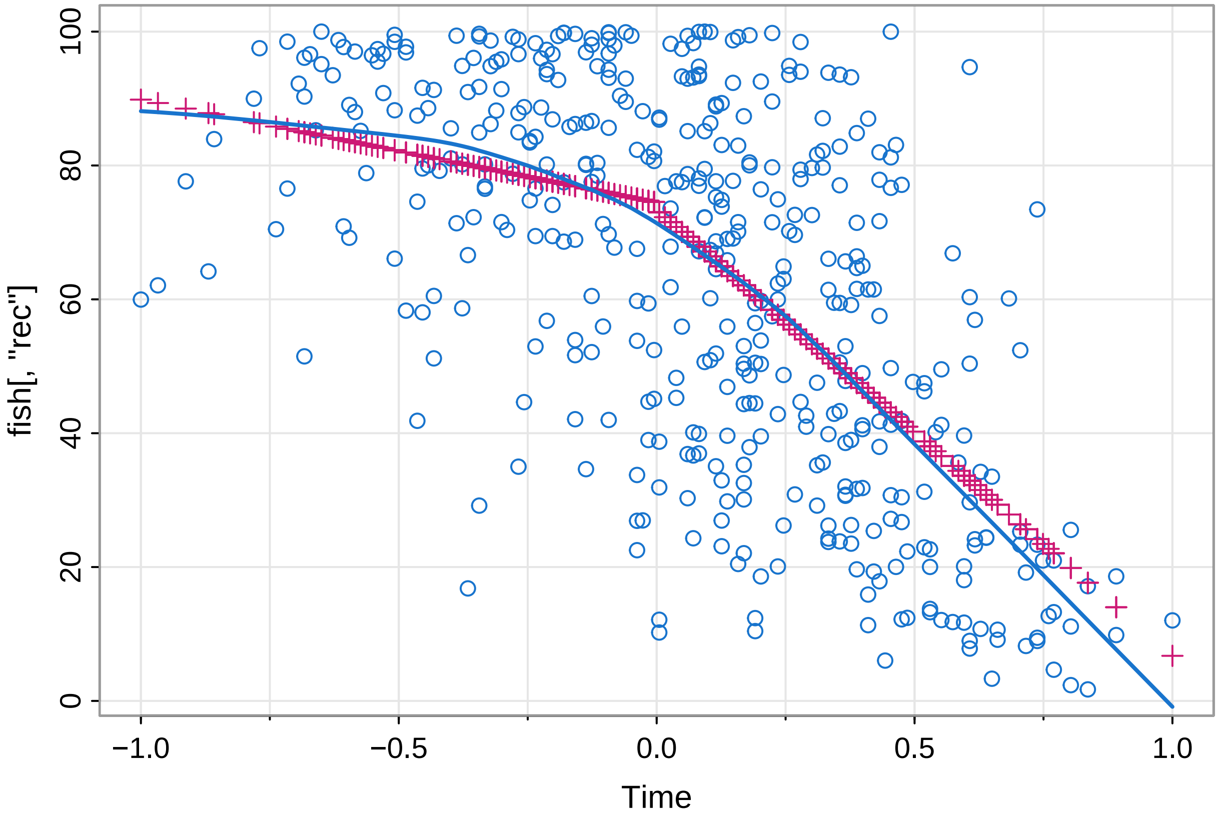
```
lag2.plot(soi, rec, 8, col=4, cex=1)
```



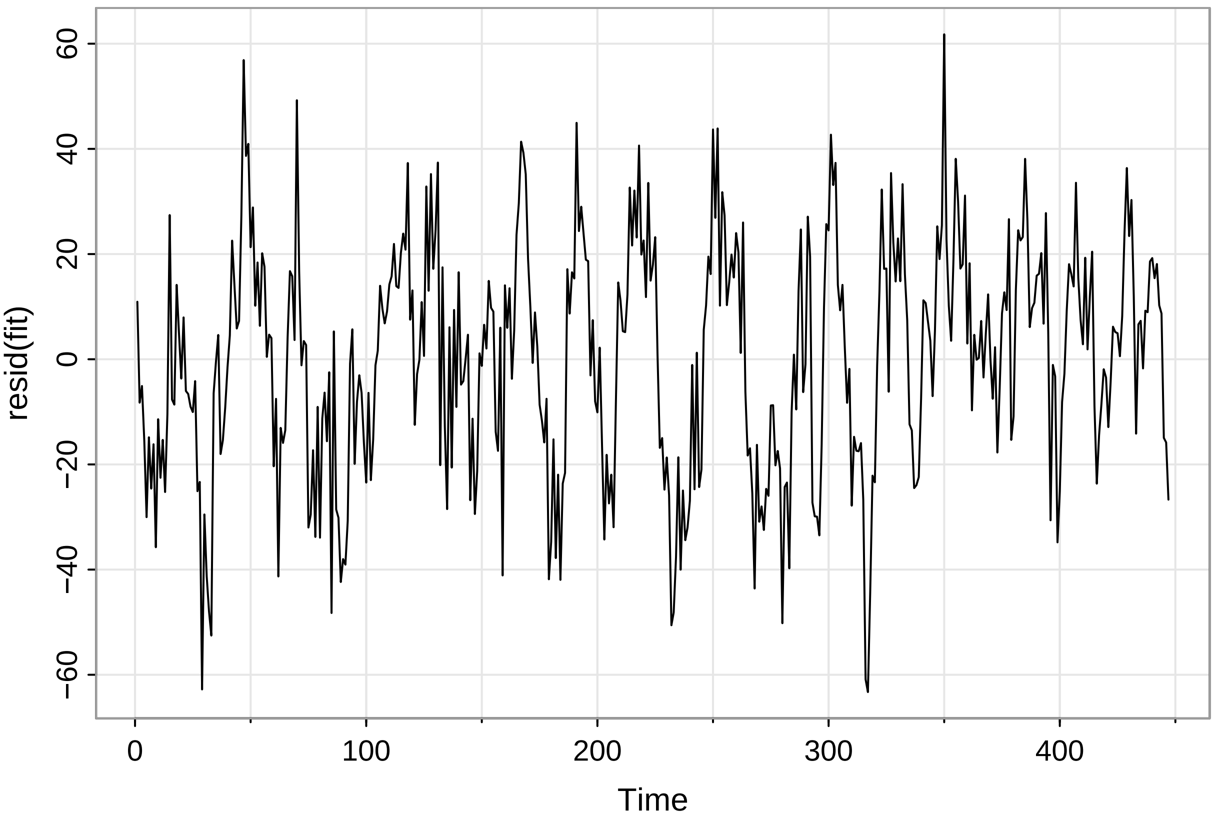
```
dummy <- ifelse(soi<0, 0, 1)
fish <- ts.intersect(rec, soiL6=lag(soi,-6), dL6=lag(dummy,-6))
summary(fit <- lm(rec~ soiL6*dL6, data=fish, na.action=NULL))
```

```
## 
## Call:
## lm(formula = rec ~ soiL6 * dL6, data = fish, na.action = NULL)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -63.291 -15.821   2.224  15.791  61.788 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   74.479      2.865  25.998  < 2e-16 ***
## soiL6        -15.358      7.401  -2.075   0.0386 *  
## dL6           -1.139      3.711  -0.307   0.7590    
## soiL6:dL6    -51.244      9.523  -5.381  1.2e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 21.84 on 443 degrees of freedom
## Multiple R-squared:  0.4024, Adjusted R-squared:  0.3984 
## F-statistic: 99.43 on 3 and 443 DF,  p-value: < 2.2e-16
```

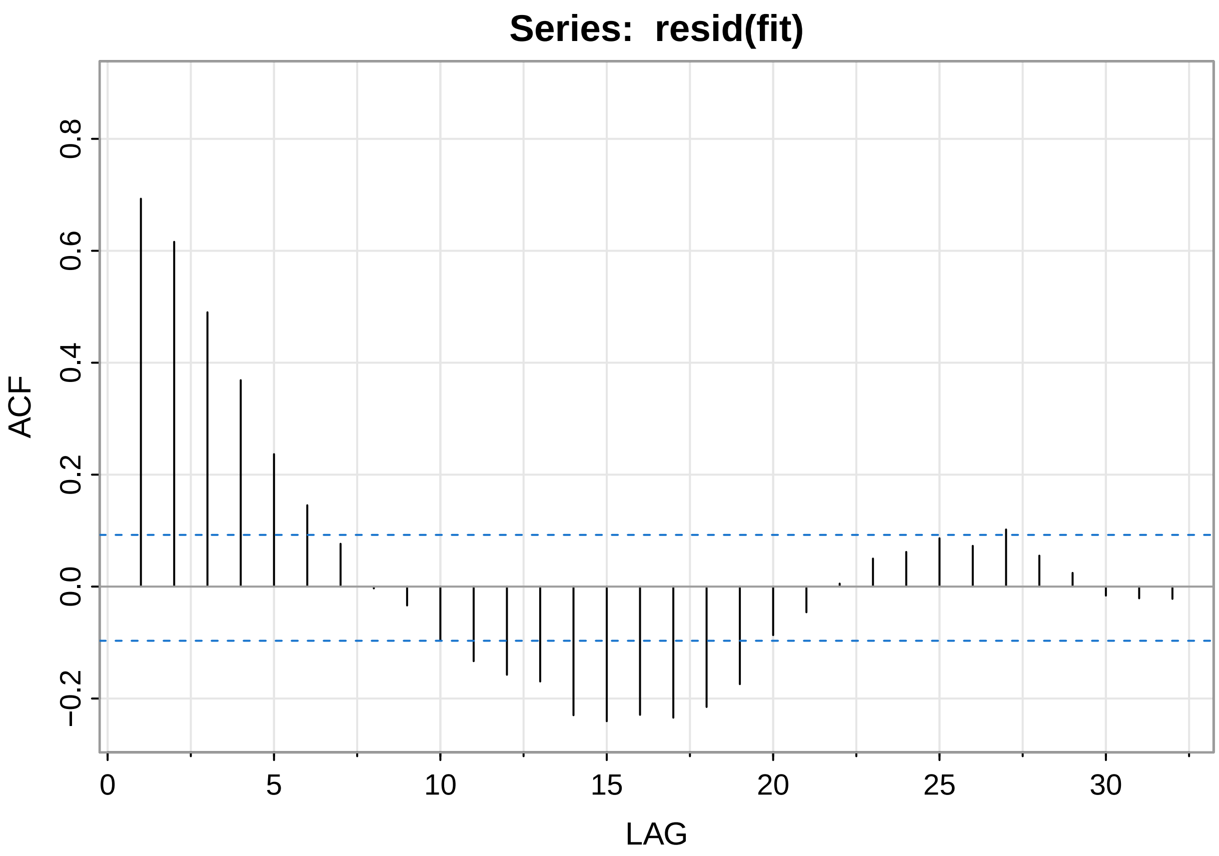
```
tsplot(fish[,'soiL6'], fish[,'rec'], type='p', col=4)
points(fish[,'soiL6'], fitted(fit), pch=3, col=6)
lines(lowess(fish[,'soiL6'], fish[,'rec']), col=4, lwd=2)
```



```
tsplot(resid(fit))
```



```
acf1(resid(fit))
```



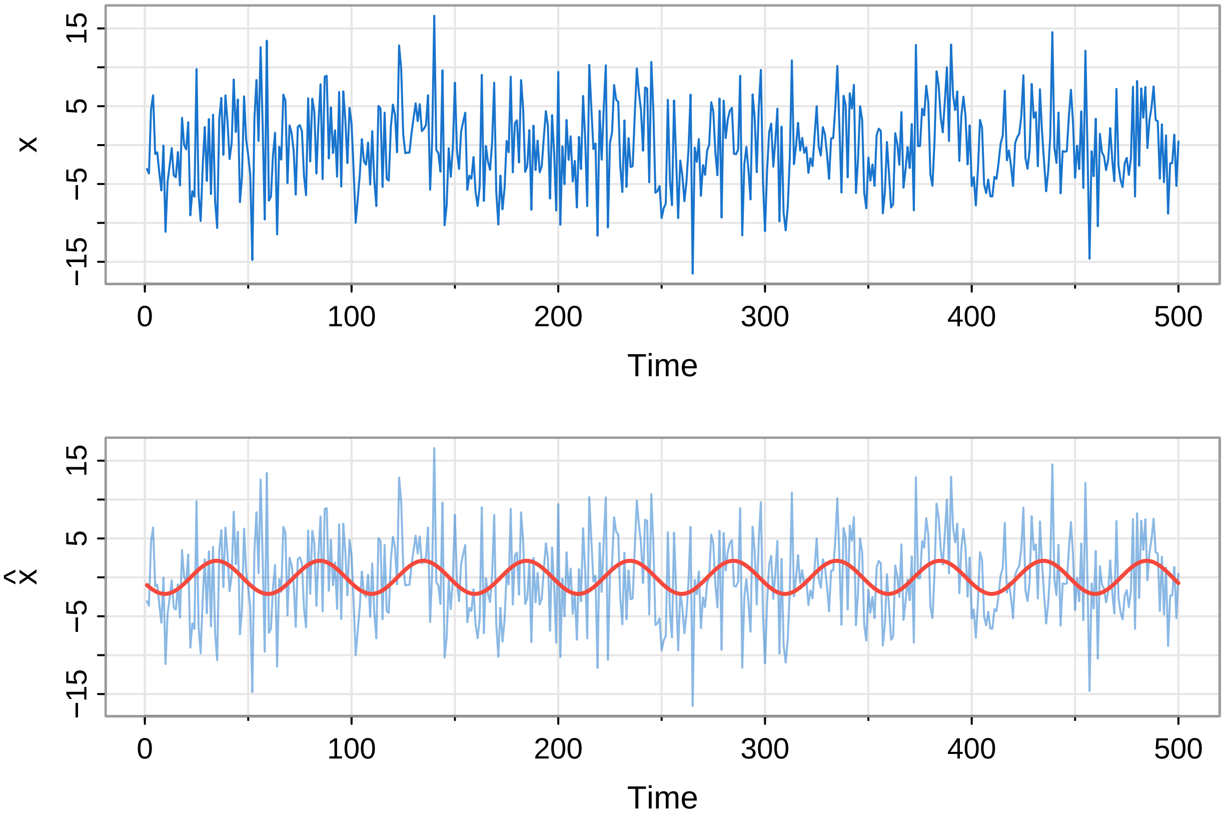
```
##  [1]  0.69  0.62  0.49  0.37  0.24  0.15  0.08  0.00 -0.03 -0.10 -0.13 -0.16
## [13] -0.17 -0.23 -0.24 -0.23 -0.23 -0.22 -0.17 -0.09 -0.05  0.01  0.05  0.06
## [25]  0.09  0.07  0.10  0.06  0.02 -0.02 -0.02 -0.02
```

```
set.seed(90210)
x <- 2*cos(2*pi*1:500/50 + .6*pi) + rnorm(500,0,5)
z1 <- cos(2*pi*1:500/50)
z2 <- sin(2*pi*1:500/50)
summary(fit <- lm(x~ 0 + z1 + z2))
```

```
##
## Call:
## lm(formula = x ~ 0 + z1 + z2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.8584  -3.8525  -0.3186   3.3487  15.5440
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## z1  -0.7442     0.3274  -2.273   0.0235 *
## z2  -1.9949     0.3274  -6.093 2.23e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 5.177 on 498 degrees of freedom
## Multiple R-squared:  0.07827,    Adjusted R-squared:  0.07456
## F-statistic: 21.14 on 2 and 498 DF,  p-value: 1.538e-09
```

```
par(mfrow=c(2,1))
tsplot(x, col=4)
tsplot(x, ylab=expression(hat(x)), col=astsa.col(4, .5))
lines(fitted(fit), col=2, lwd=2)
```


x
Time
$\hat{x}$
Time