STAT4870 Week3

Anton Yang

2024-09-06

2.2 Stationarity

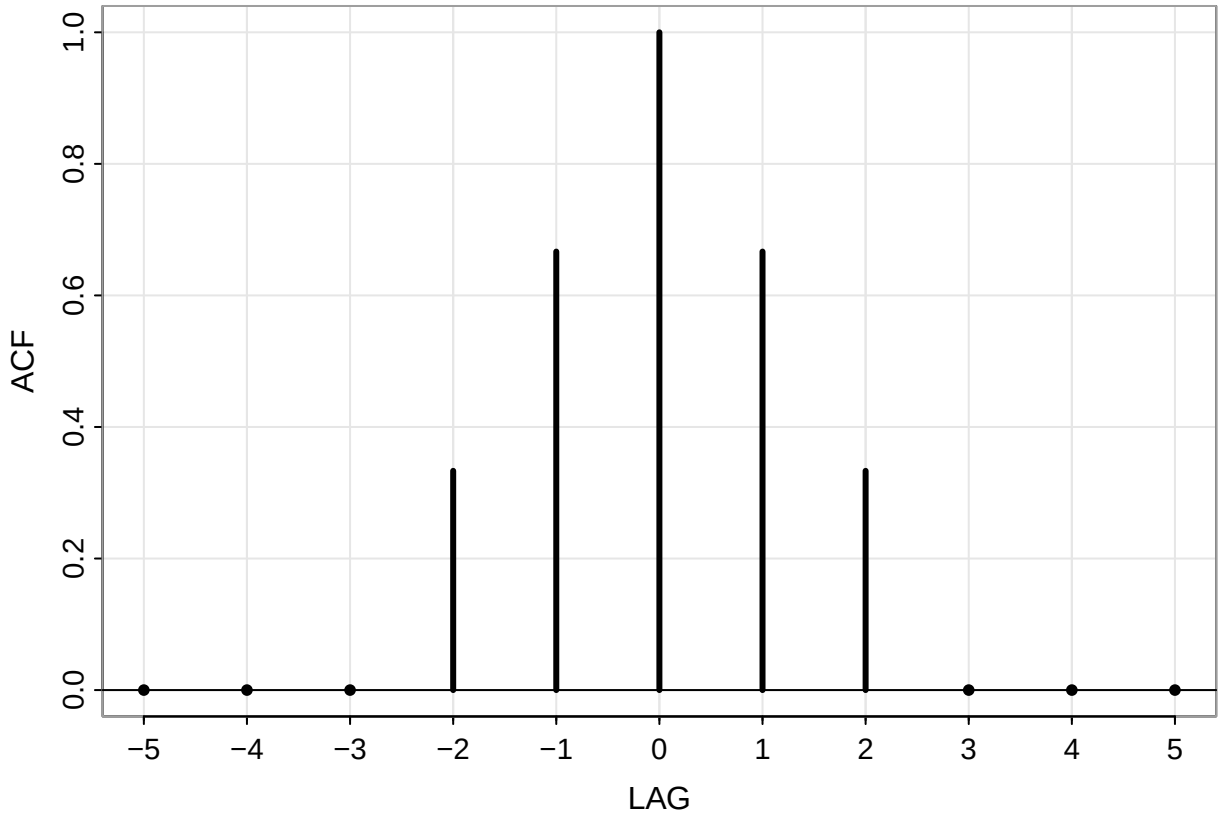
```
library("astsa")
library("xts")
```

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

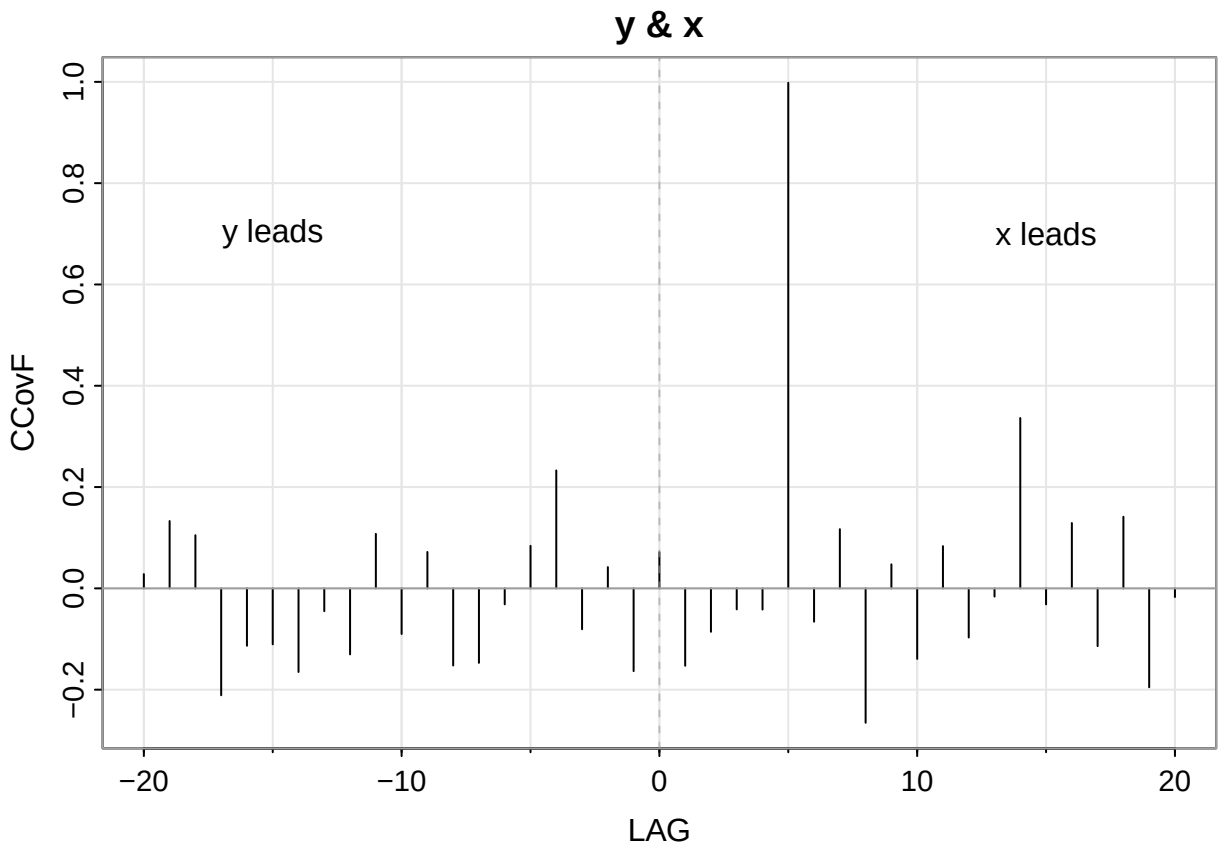
```
##
## Attaching package: 'zoo'
```

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

```
ACF <- c(0,0,0,1,2,3,2,1,0,0,0)/3
LAG <- -5:5
tsplot(LAG, ACF, type="h", lwd=3, xlab="LAG")
abline(h=0)
points(LAG[-(4:8)], ACF[-(4:8)], pch=20)
axis(1, at=seq(-5, 5, by=2))
```




```
x <- rnorm(100)
y <- lag(x,-5) + rnorm(100)
ccf2(y, x, ylab="CCovF", type="covariance")
text(c(-15,15),.7,c("y leads","x leads"))
```

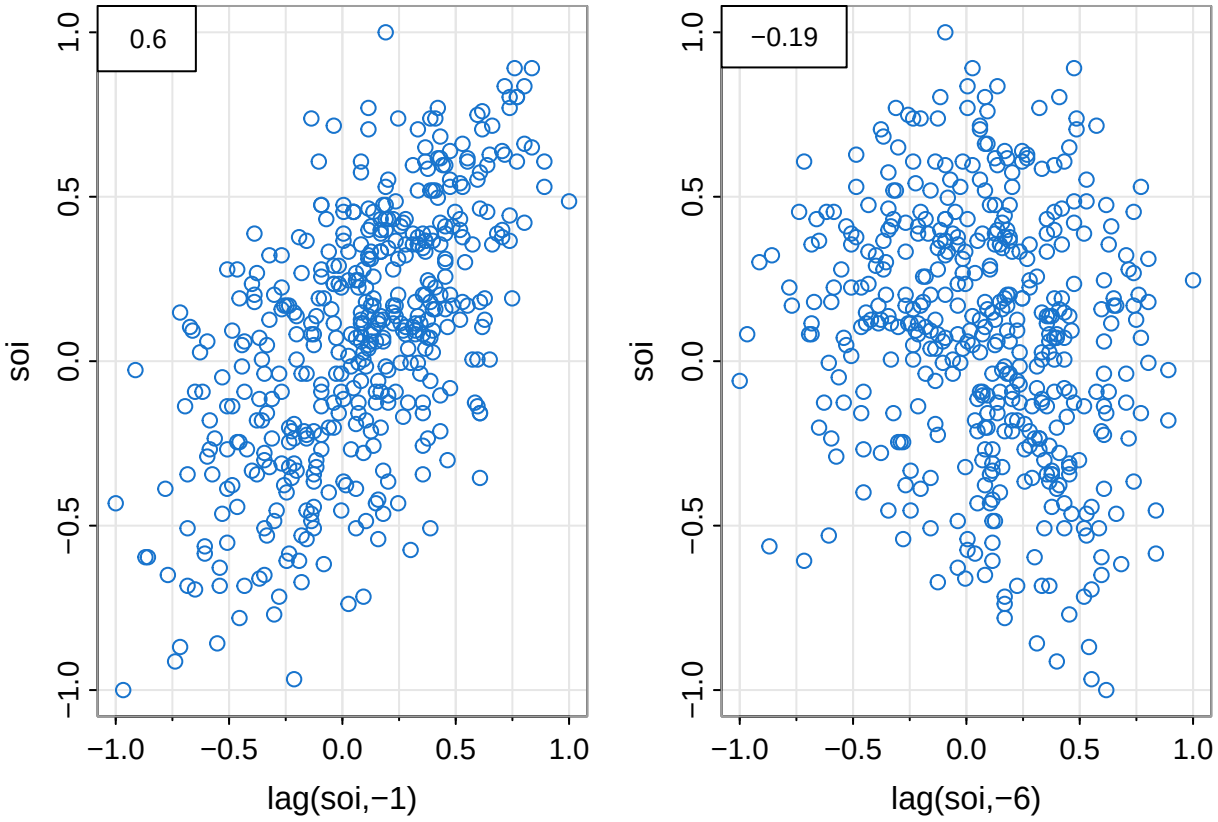


2.3 Estimation of Correlation

```
(r <- round( acf1(soi, 6, plot=FALSE), 2))
```

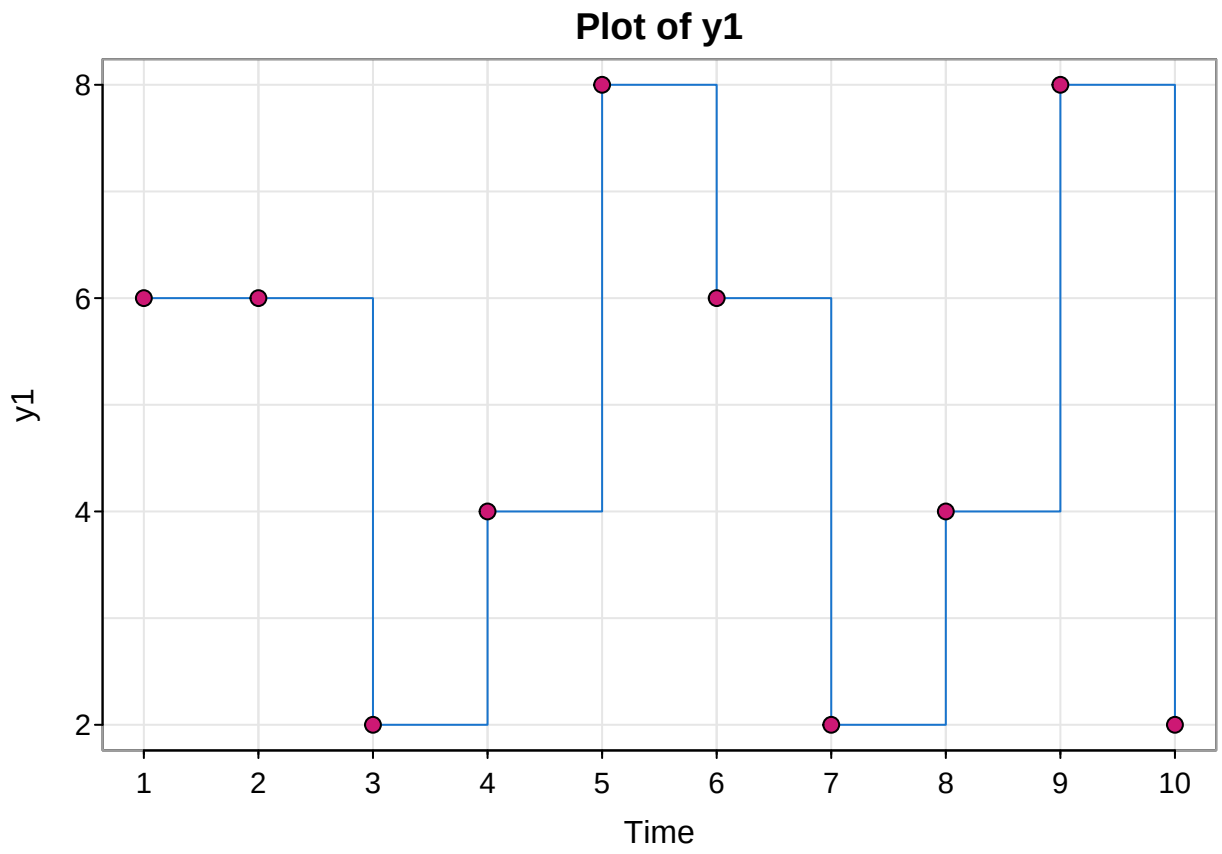
```
## [1]  0.60  0.37  0.21  0.05 -0.11 -0.19
```

```
par(mfrow=c(1,2))
tsplot(lag(soi,-1), soi, col=4, type='p', xlab='lag(soi,-1)')
legend("topleft", legend=r[1], bg="white", adj=.45, cex = 0.85)
tsplot(lag(soi,-6), soi, col=4, type='p', xlab='lag(soi,-6)')
legend("topleft", legend=r[6], bg="white", adj=.25, cex = 0.8)
```




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

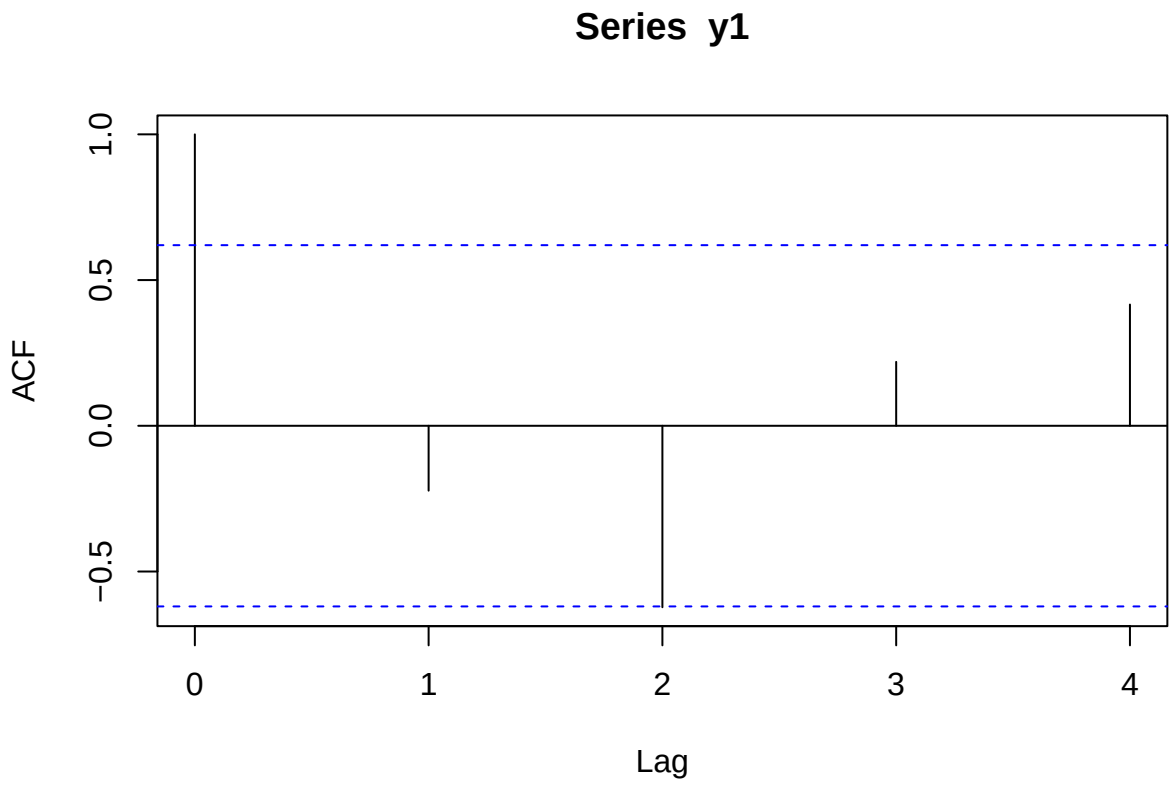
```
set.seed(101011)
x1 <- sample(c(-2,2), 11, replace=TRUE)
y1 <- 5 + filter(x1, sides=1, filter=c(1,-.5))[-1]
tsplot(y1, type="s", col=4, xaxt="n", yaxt="n",main="Plot of y1")
axis(1, 1:10); axis(2, seq(2,8,2), las=1)
points(y1, pch=21, cex=1.1, bg=6)
```



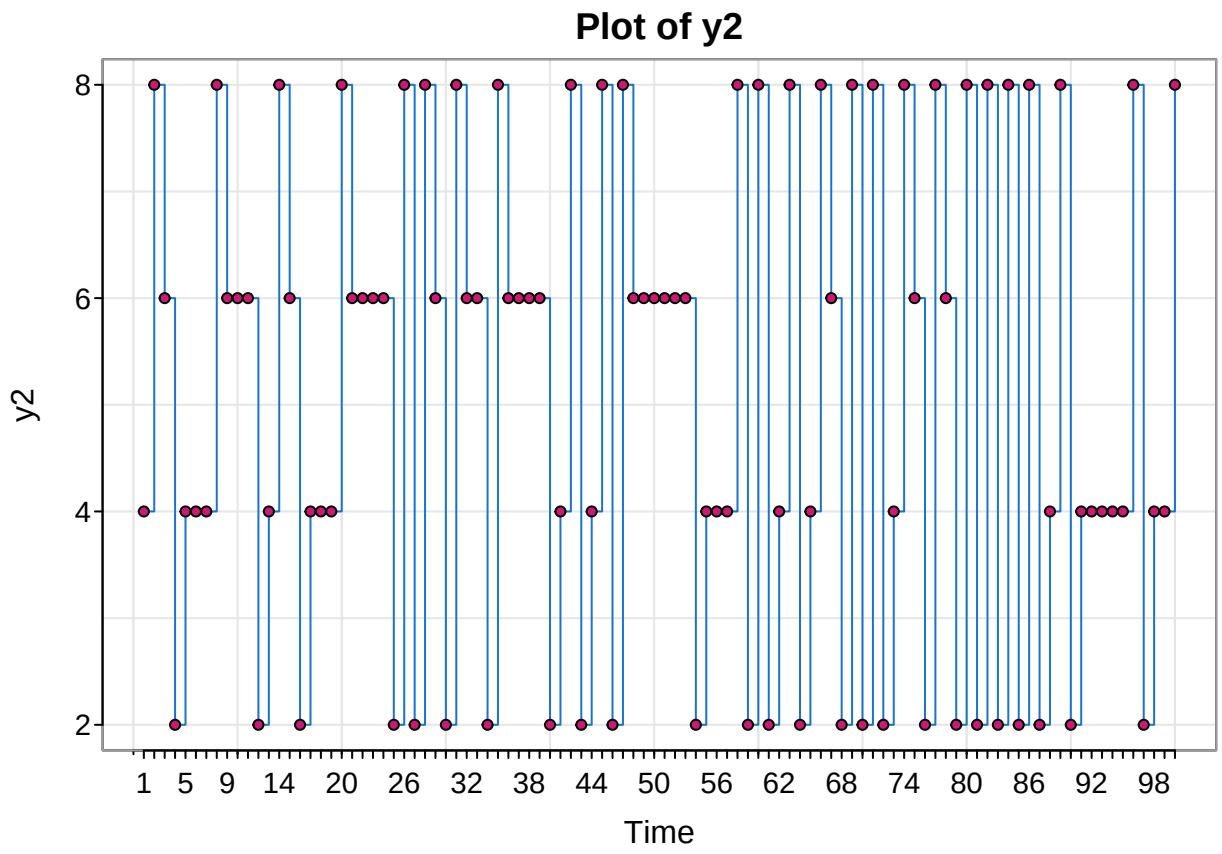

```
acf(y1, lag.max=4,plot=FALSE)
```

```
##
## Autocorrelations of series 'y1', by lag
##
##      0      1      2      3      4
##  1.000 -0.223 -0.623  0.219  0.416
```

```
acf(y1, lag.max=4)
```



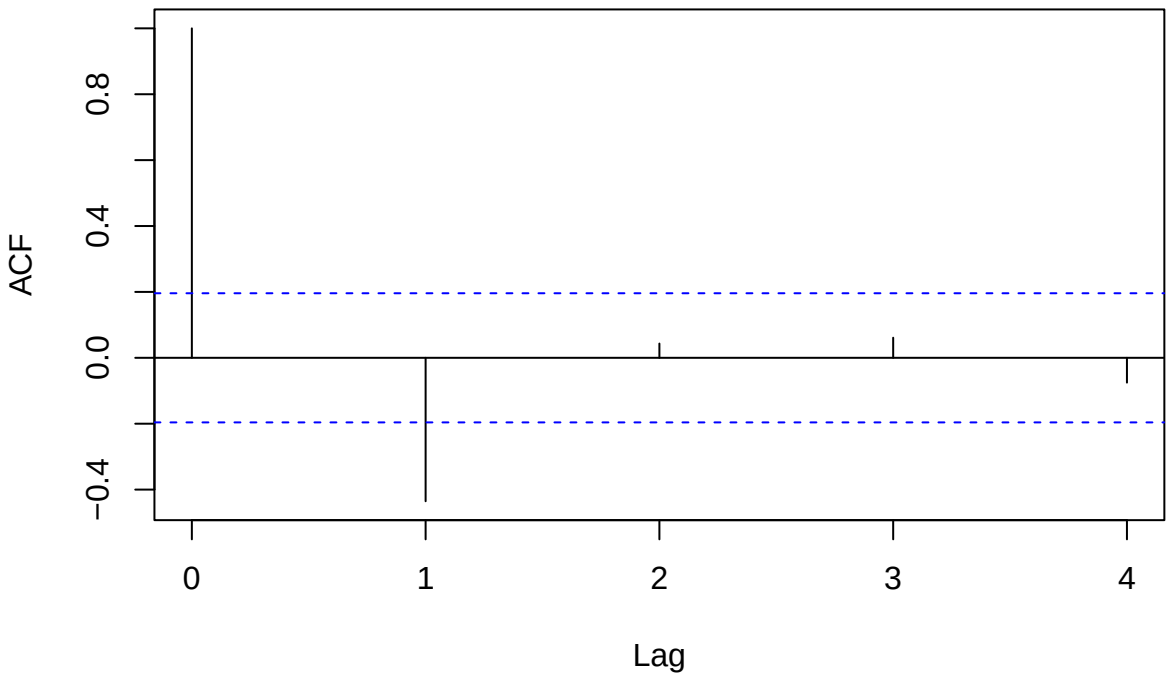

```
x2 <- sample(c(-2,2), 101, replace=TRUE)
y2 <- 5 + filter(x2, sides=1, filter=c(1,-.5))[-1]
tsplot(y2, type="s", col=4, xaxt="n", yaxt="n",main="Plot of y2")
axis(1, 1:100); axis(2, seq(2,8,2), las=1)
points(y2, pch=21, cex=0.7, bg=6)
```




```
acf(y2, lag.max=4, plot=FALSE)
```

```
##
## Autocorrelations of series 'y2', by lag
##
##      0      1      2      3      4
##  1.000 -0.435  0.043  0.061 -0.075
```

```
acf(y2, lag.max=4)
```



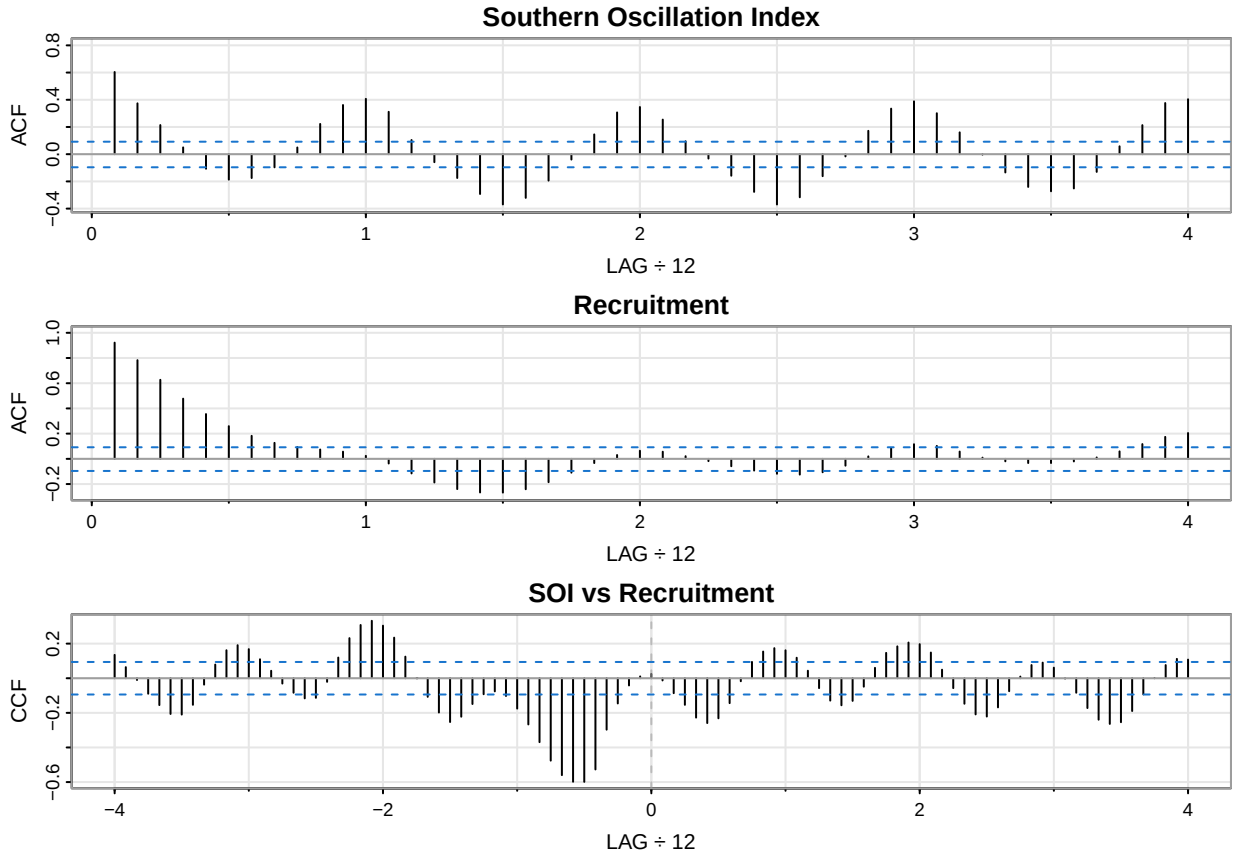

```
op<-par(mfrow=c(3,1))
acf1(soi, 48, main="Southern Oscillation Index")
```

```
##  [1]  0.60  0.37  0.21  0.05 -0.11 -0.19 -0.18 -0.10  0.05  0.22  0.36  0.41
## [13]  0.31  0.10 -0.06 -0.17 -0.29 -0.37 -0.32 -0.19 -0.04  0.15  0.31  0.35
## [25]  0.25  0.10 -0.03 -0.16 -0.28 -0.37 -0.32 -0.16 -0.02  0.17  0.33  0.39
## [37]  0.30  0.16  0.00 -0.13 -0.24 -0.27 -0.25 -0.13  0.06  0.21  0.38  0.40
```

```
acf1(rec, 48, main="Recruitment")
```

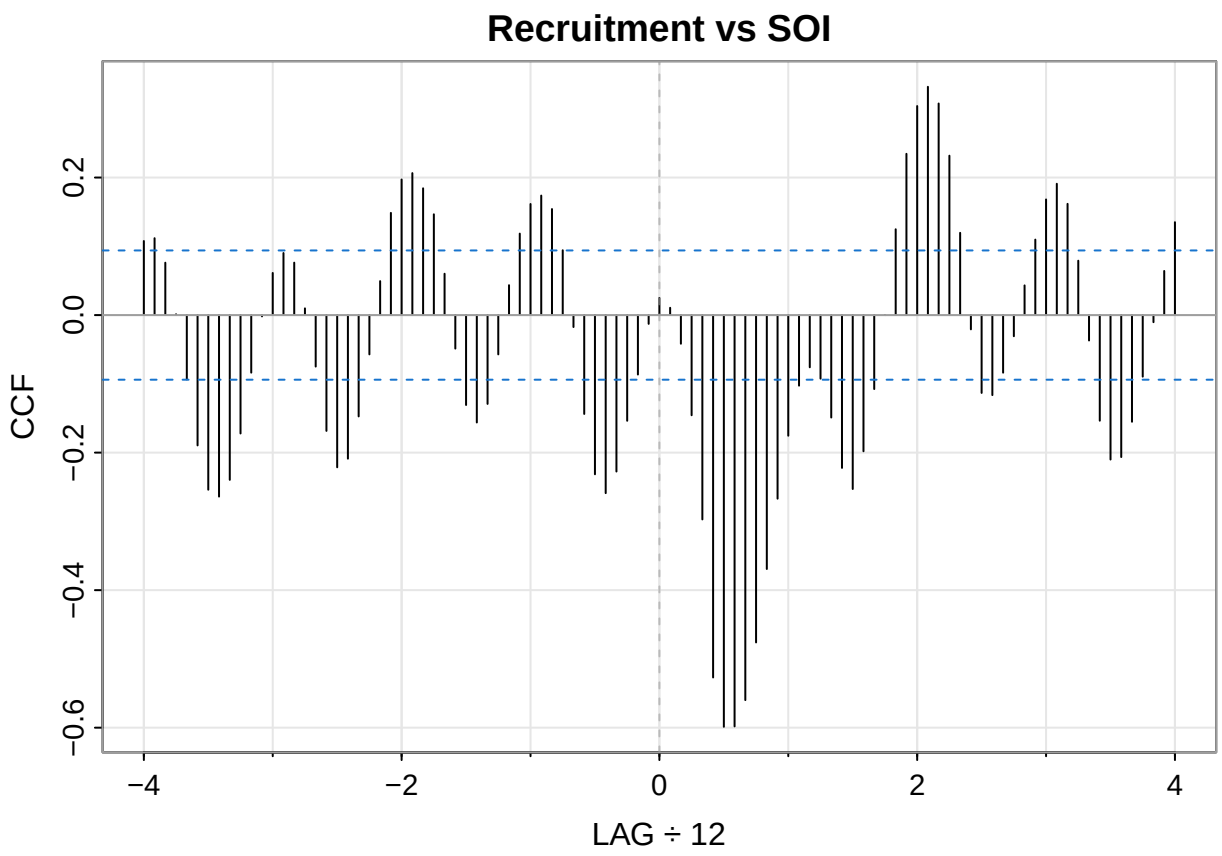
```
##  [1]  0.92  0.78  0.63  0.48  0.36  0.26  0.18  0.13  0.09  0.07  0.06  0.02
## [13] -0.04 -0.12 -0.19 -0.24 -0.27 -0.27 -0.24 -0.19 -0.11 -0.03  0.03  0.06
## [25]  0.06  0.02 -0.02 -0.06 -0.09 -0.12 -0.13 -0.11 -0.05  0.02  0.08  0.12
## [37]  0.10  0.06  0.01 -0.02 -0.03 -0.03 -0.02  0.01  0.06  0.12  0.17  0.20
```

```
ccf2(soi, rec, 48, main="SOI vs Recruitment")
```




```
par(op)
```

```
ccf2(rec,soi, 48, main="Recruitment vs SOI")
```



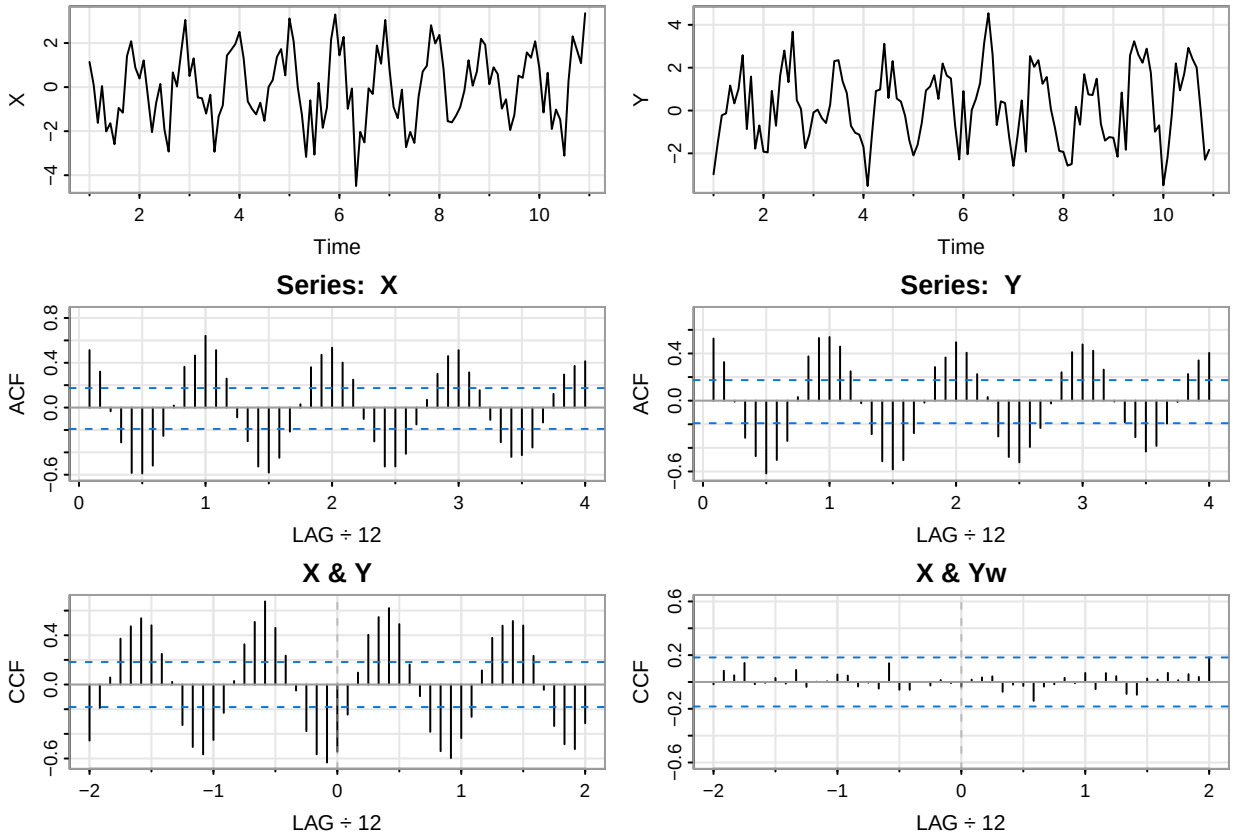

```
set.seed(1492)
num <- 120
t <- 1:num
X <- ts( 2*cos(2*pi*t/12) + rnorm(num), freq=12 )
Y <- ts( 2*cos(2*pi*(t+5)/12) + rnorm(num), freq=12 )
Yw <- resid(lm(Y~ cos(2*pi*t/12) + sin(2*pi*t/12), na.action=NULL))
op<-par(mfrow=c(3,2))
tsplot(X)
tsplot(Y)
acf1(X, 48)
```

```
##  [1]  0.51  0.32 -0.03 -0.31 -0.58 -0.59 -0.52 -0.25  0.02  0.36  0.46  0.64
## [13]  0.51  0.26 -0.09 -0.30 -0.53 -0.58 -0.45 -0.21  0.03  0.36  0.47  0.53
## [25]  0.40  0.25 -0.10 -0.30 -0.53 -0.53 -0.41 -0.15  0.07  0.30  0.46  0.51
## [37]  0.31  0.16 -0.11 -0.31 -0.44 -0.42 -0.36 -0.13  0.12  0.30  0.37  0.41
```

```
acf1(Y, 48)
```

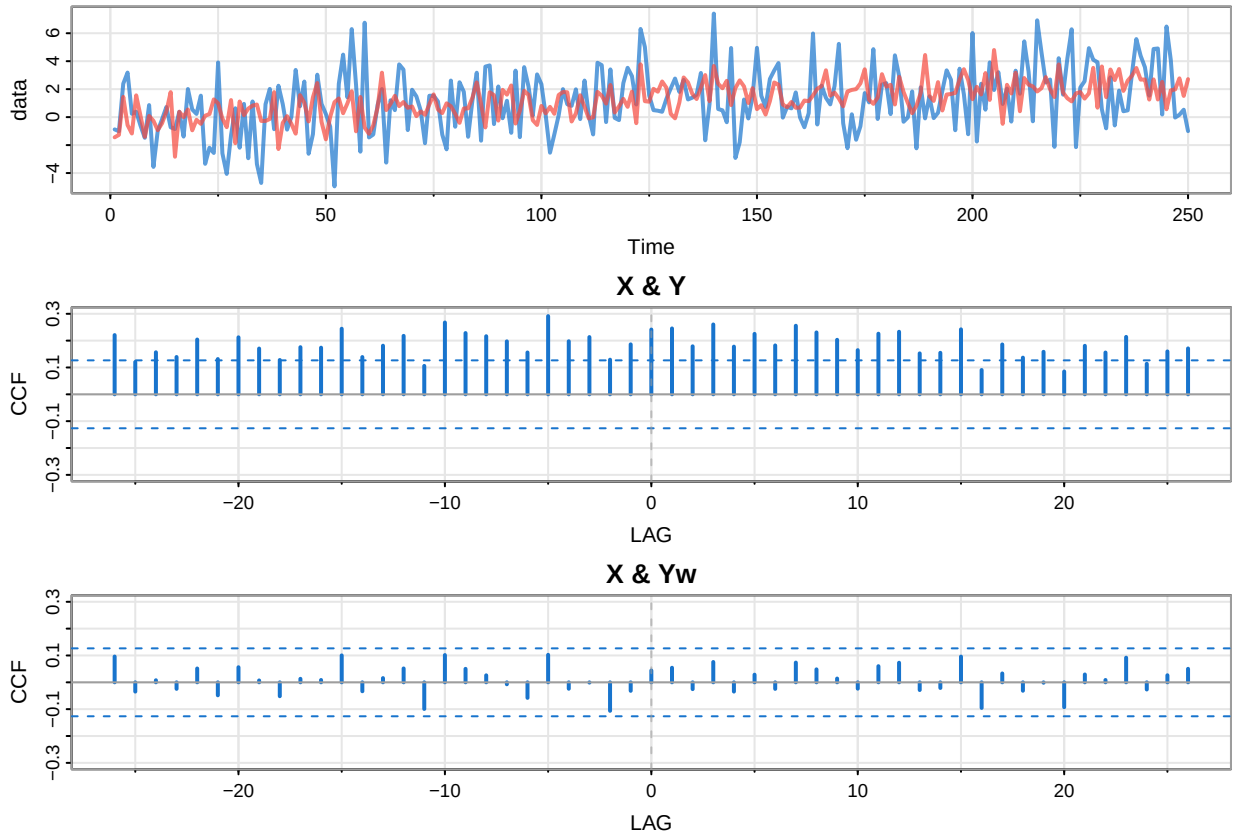
```
##  [1]  0.53  0.33 -0.01 -0.31 -0.47 -0.62 -0.50 -0.34  0.03  0.38  0.53  0.54
## [13]  0.46  0.25 -0.02 -0.28 -0.51 -0.58 -0.50 -0.27 -0.02  0.28  0.37  0.49
## [25]  0.41  0.22  0.03 -0.30 -0.48 -0.52 -0.39 -0.23 -0.02  0.24  0.41  0.48
## [37]  0.42  0.26 -0.01 -0.18 -0.31 -0.43 -0.38 -0.19 -0.01  0.23  0.34  0.40
```

```
ccf2(X, Y, 24)
ccf2(X, Yw, 24, ylim=c(-.6,.6))
```




```
par(op)
```

```
set.seed(90210)
num <- 250
t <- 1:num
X <- .01*t + rnorm(num,0,2)
Y <- .01*t + rnorm(num) # x and y are uncorrelated!
op<-par(mfrow=c(3,1))
tsplot(cbind(X,Y), spag=TRUE, col=astsa.col(c(4,2),.7), lwd=2, ylab='data')
ccf2(X, Y, ylim=c(-.3,.3), col=4, lwd=2)
Yw <- resid(lm(Y~t)) # whiten Y by removing trend
ccf2(X, Yw, ylim=c(-.3,.3), col=4, lwd=2)
```




```
par(op)
```

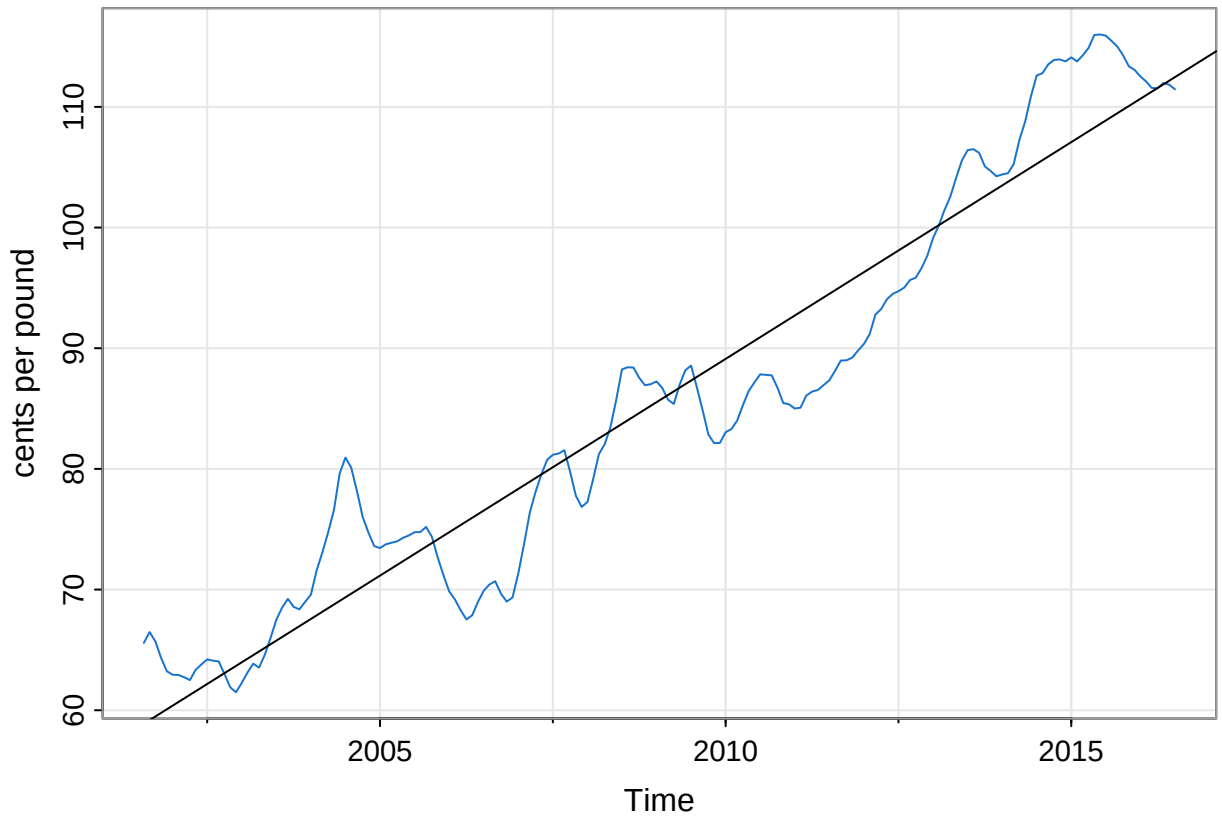
3.1 Ordinary Least Squares for Time Series

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

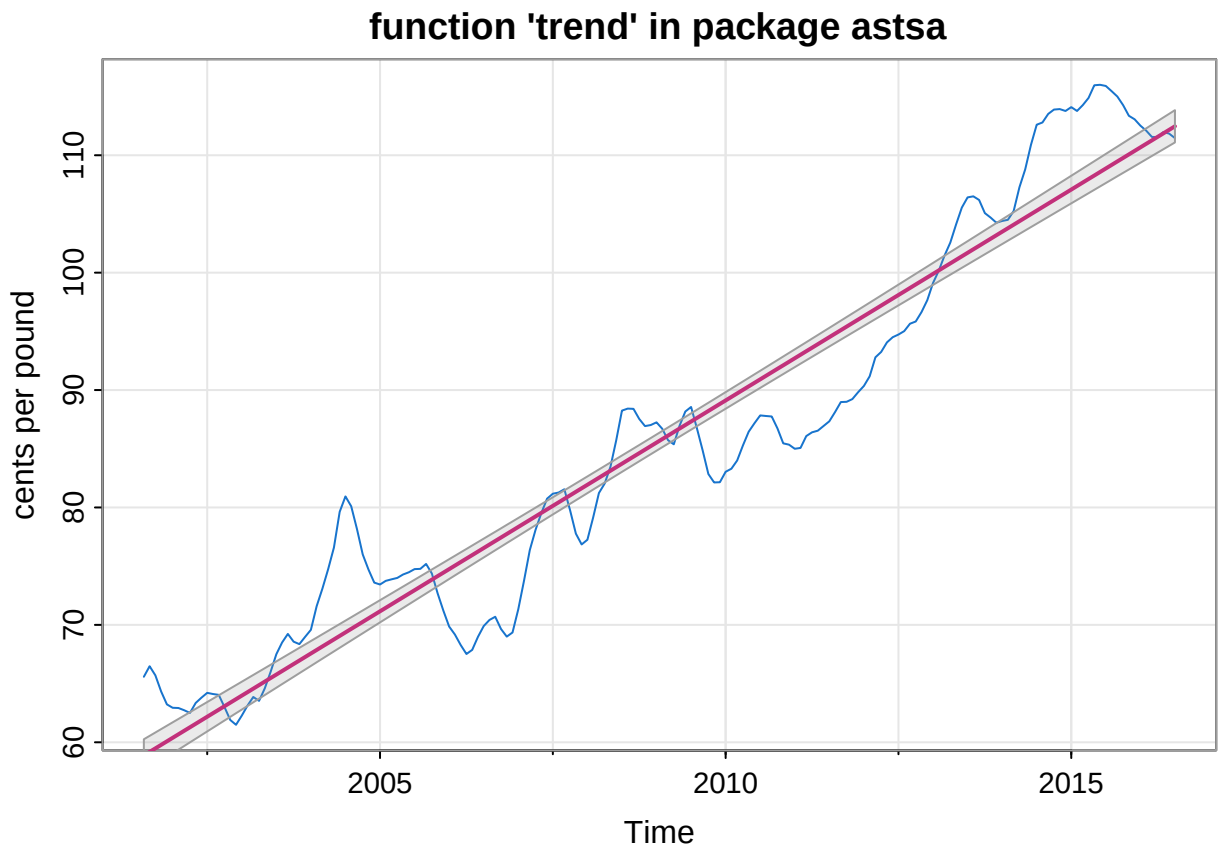
```
##
## Call:
## lm(formula = chicken ~ time(chicken), na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7411 -3.4730  0.8251  2.7738 11.5804
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.131e+03  1.624e+02  -43.91   <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
```

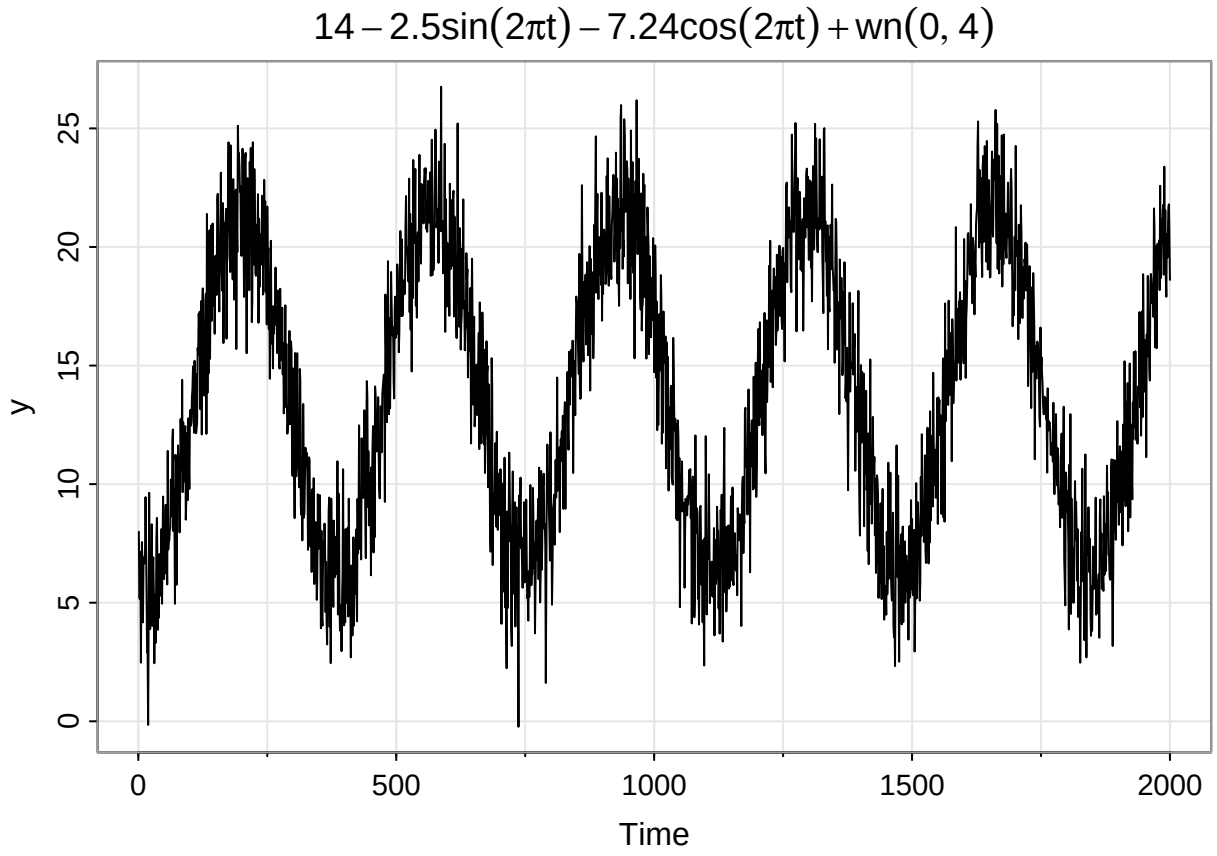
```
tsplot(chicken, ylab="cents per pound",col=4)
abline(fit)
```




```
# astsa version 1.5 has a new trend script that can produce Figure 3.1 with 95% CIs as follows:
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)))
```




```
# modeling
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
## -6.6341 -1.3651 -0.0181  1.3936  6.4789
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     13.92032    0.04504  309.08   <2e-16 ***
## sin(2 * pi * t) -2.47711    0.06355  -38.98   <2e-16 ***
## cos(2 * pi * t) -7.28698    0.06362 -114.54   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.007 on 1997 degrees of freedom
## Multiple R-squared:   0.88,  Adjusted R-squared:  0.8798
## F-statistic:  7320 on 2 and 1997 DF,  p-value: < 2.2e-16
```

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

```
##
## Call:
## lm(formula = y ~ sin(2 * pi * t))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8470  -4.8295   0.0408   4.8087  12.0599
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      13.8918     0.1239  112.14   <2e-16 ***
## sin(2 * pi * t)  -2.4782     0.1748  -14.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.522 on 1998 degrees of freedom
## Multiple R-squared:  0.09139,    Adjusted R-squared:  0.09094
## F-statistic:   201 on 1 and 1998 DF,  p-value: < 2.2e-16
```

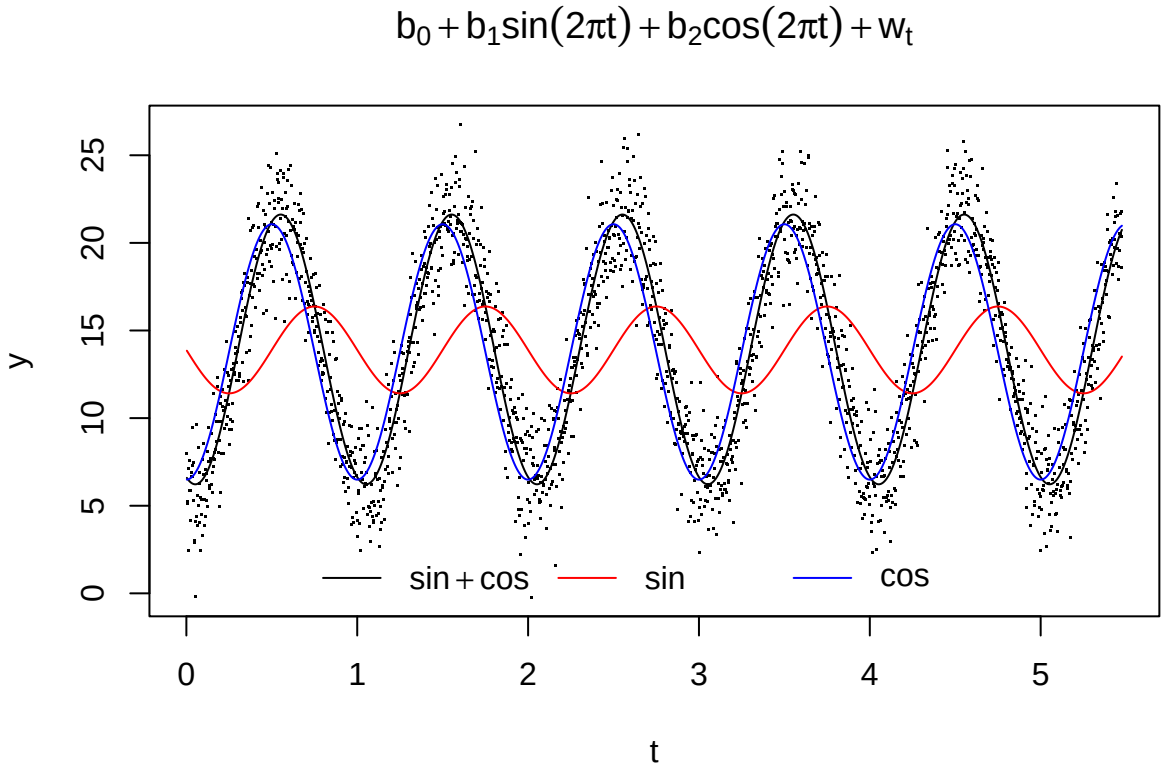
```r
mdl3 <- lm(y~cos(2*pi*t))
summary(mdl3)
```

```
##
## Call:
## lm(formula = y ~ cos(2 * pi * t))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3686 -1.9056 -0.0416  1.8589  8.9513
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     13.77707    0.05955  231.36   <2e-16 ***
## cos(2 * pi * t) -7.28734    0.08440  -86.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.663 on 1998 degrees of freedom
## Multiple R-squared:  0.7886, Adjusted R-squared:  0.7885
## F-statistic:  7455 on 1 and 1998 DF,  p-value: < 2.2e-16
```

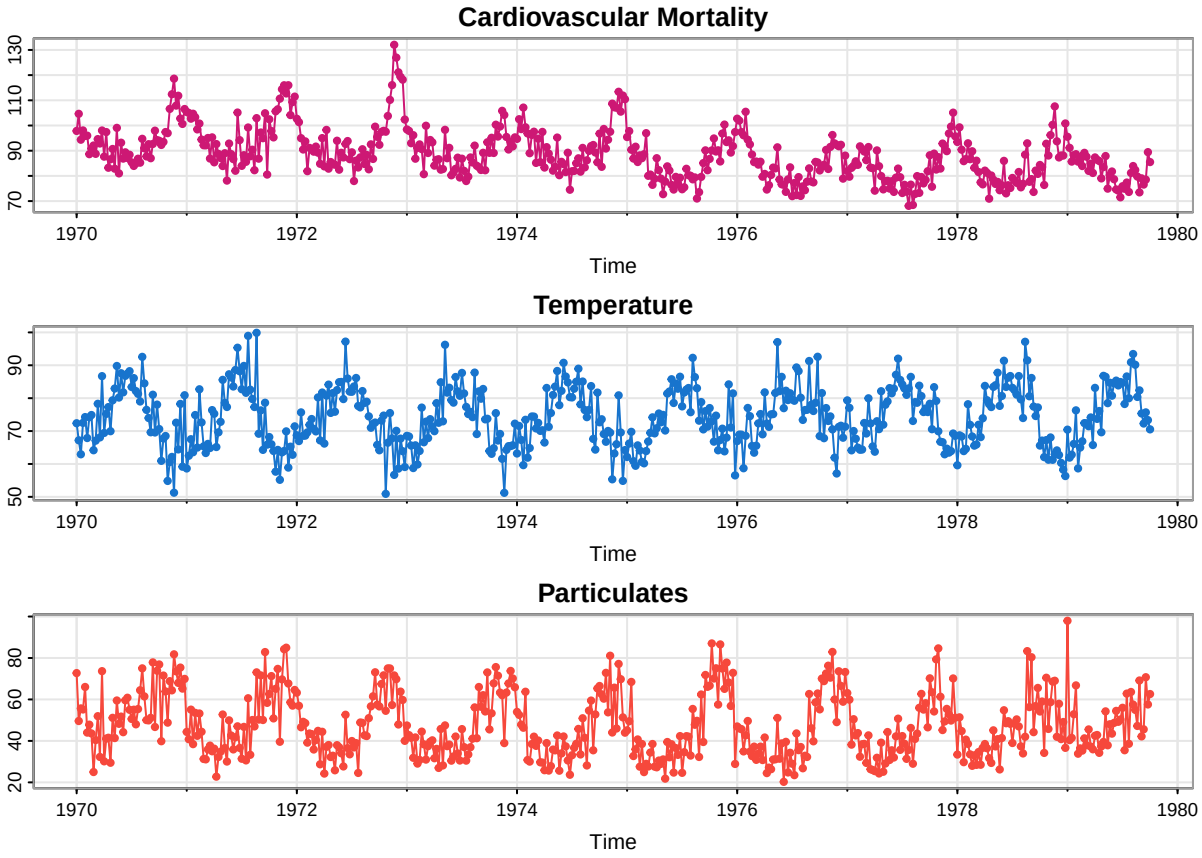
```r
merge(AIC(mdl1,mdl2,mdl3),BIC(mdl1,mdl2,mdl3),by='row.names',all=TRUE)
```

```
##   Row.names df.x       AIC df.y       BIC
## 1      mdl1    4  8468.152    4  8490.555
## 2      mdl2    3 12514.363    3 12531.166
## 3      mdl3    3  9597.611    3  9614.413
```

```
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"))
```

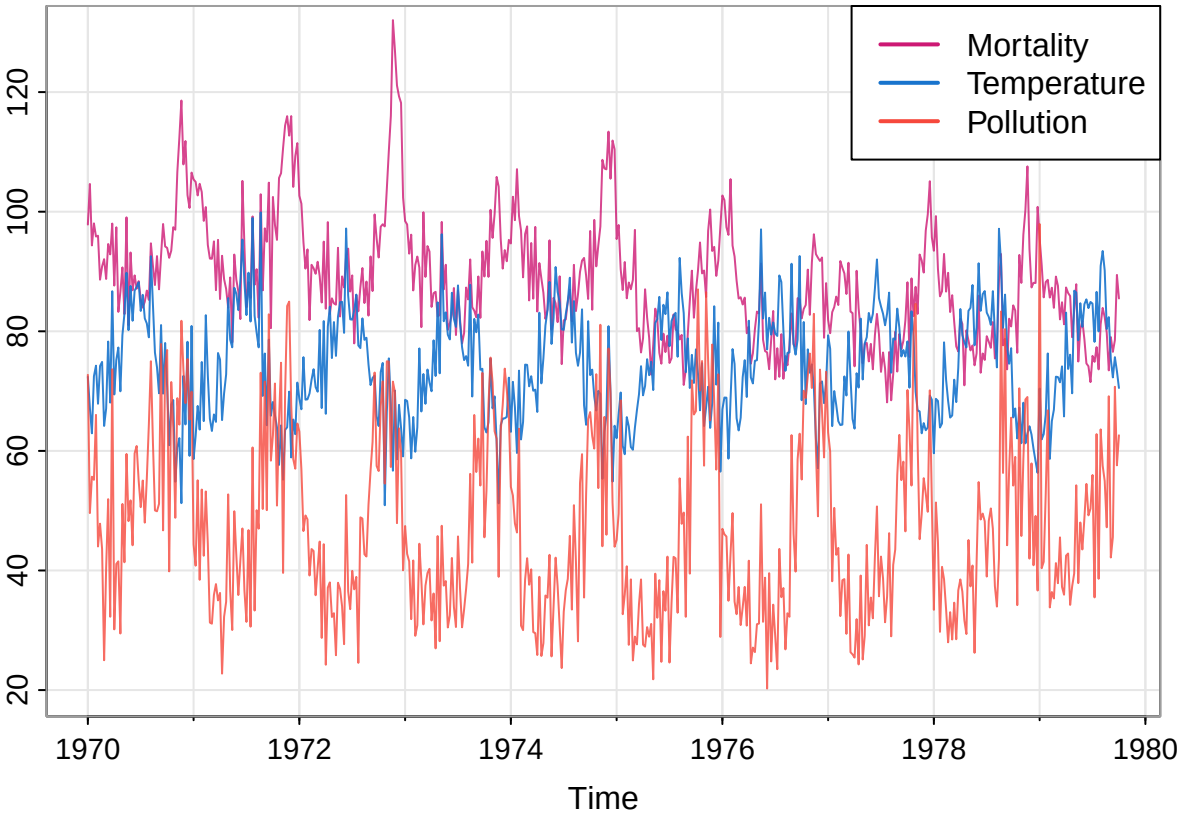



```
op<-par(mfrow=c(3,1))
tsplot(cmort, main="Cardiovascular Mortality", col=6, type="o", pch=19, ylab="")
tsplot(tempr, main="Temperature", col=4, type="o", pch=19, ylab="")
tsplot(part, main="Particulates", col=2, type="o", pch=19, ylab="")
```

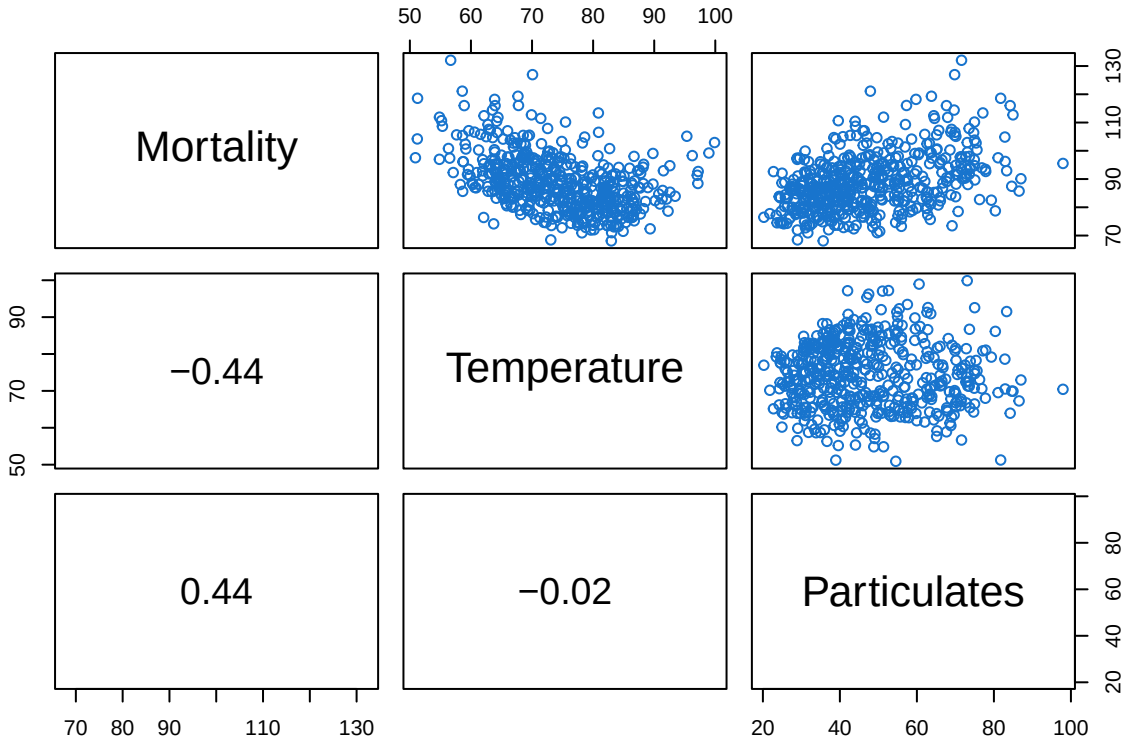



```
par(op)
```

```
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="wh
```




```
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
```

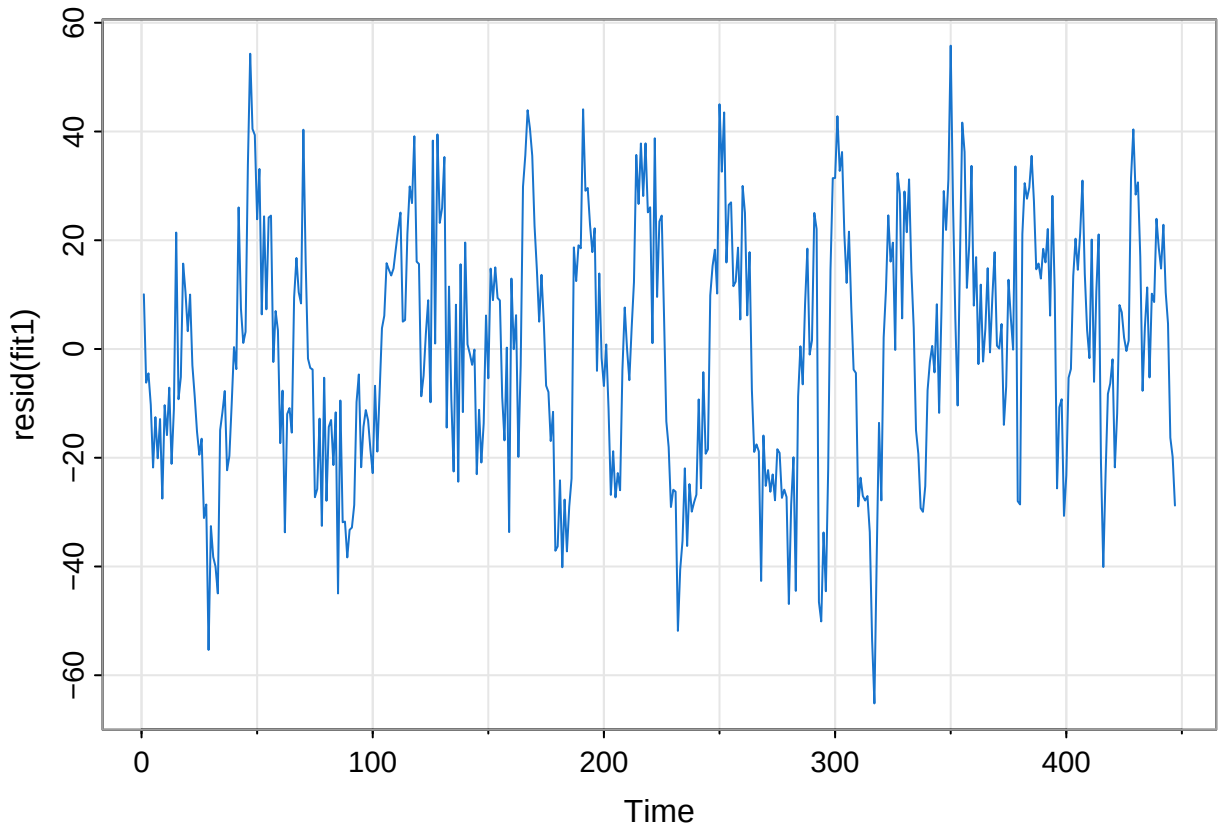
```
BIC(fit)/num - log(2*pi)
```

```
## [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)
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
```