

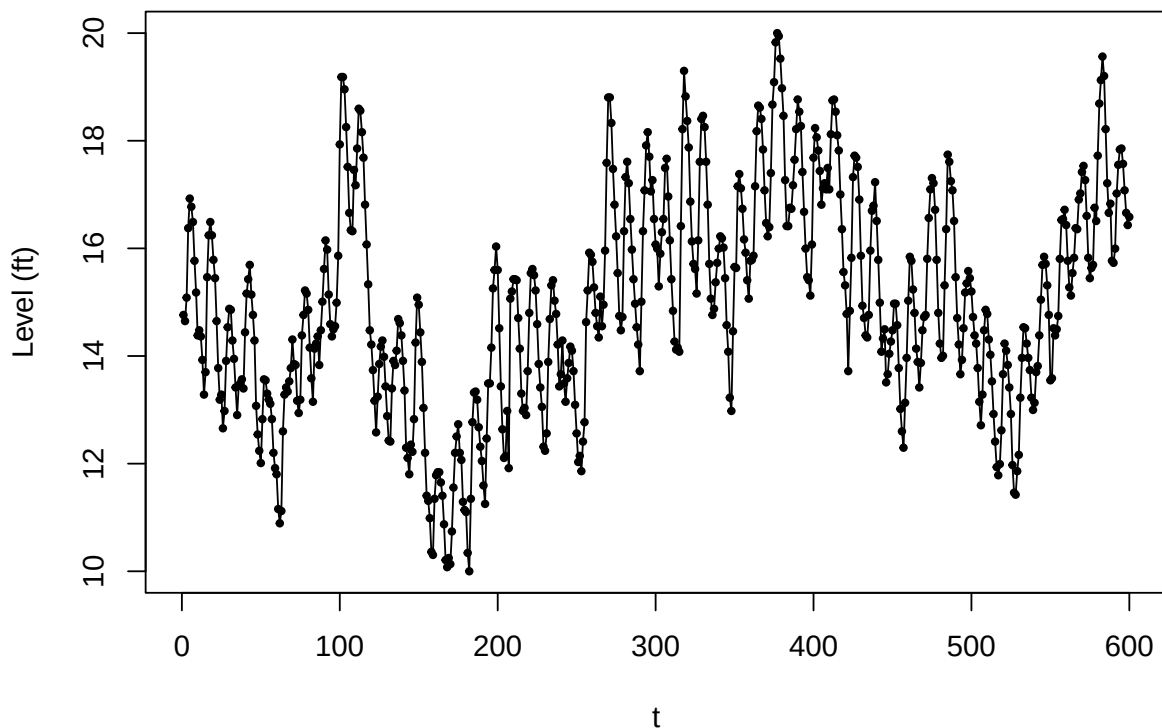
MATH 545: INTRODUCTION TO TIME SERIES ANALYSIS

WINTER 2025

EXERCISES 3: SOLUTIONS

1. The data set contained in the file *LakeErie.txt* is a collection of 600 monthly measurements of the level of Lake Erie, from January 1921 to December 1970.

```
erie.url<- 'https://www.math.mcgill.ca/dstephens/545/Data/LakeErie.txt'  
erie<-scan(erie.url)  
par(mar=c(4,4,2,2))  
plot(erie,type='l',xlab='t',ylab='Level (ft)')  
points(erie,pch=19,cex=0.5)
```



- (a) Carry out an analysis of these data in R by examining ACF and PACF plots of the original and any appropriately differenced series.
- (b) Attempt to characterize the stochastic (random) component of the data using a SARIMA model.
- (c) Using the code provided in *knitr* sheets, formulate and fit a seasonal state space model to these data.

Solutions:

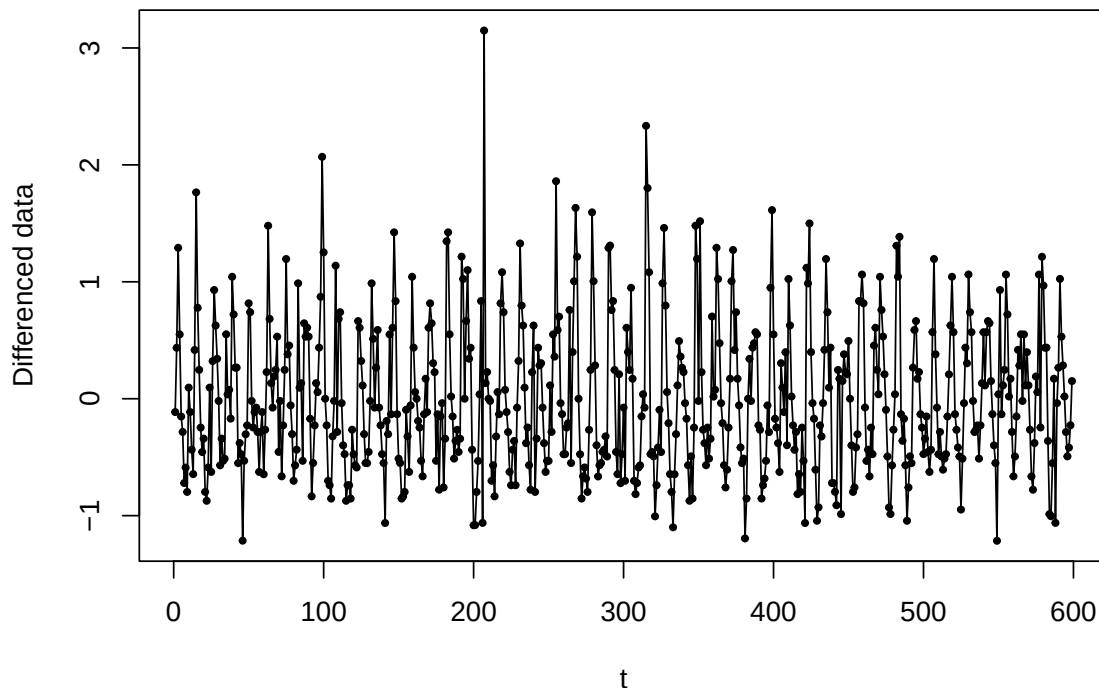
- (a) We can assess whether there is any global trend by using `lm` on the original data

```
x<-c(1:600)
y<-erie
summary(lm(y~x))$coef
```

	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	13.89846965	0.1563137695	88.913918	0.00000000000000000000
## x	0.00364253	0.0004506757	8.082376	0.000000000000003530613

It seems that there is a global increasing trend as the slope coefficient is significantly different from zero. This linear trend can be removed by first differencing.

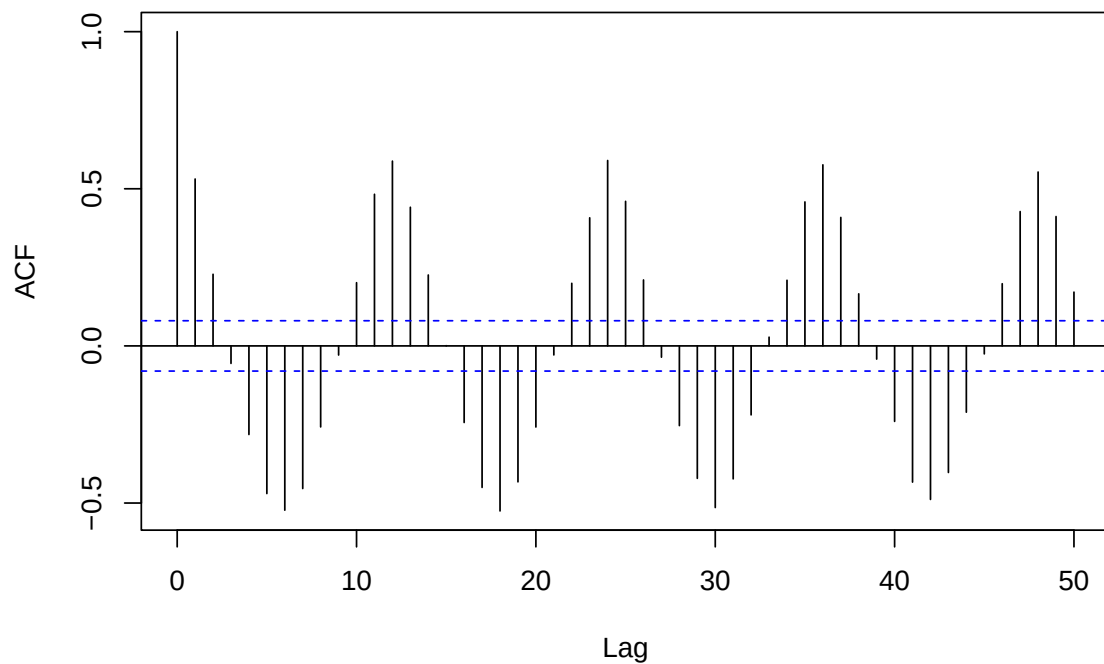
```
erie.diff<-diff(erie)
par(mar=c(4,4,2,2))
plot(erie.diff,type='l',xlab='t',ylab='Differenced data')
points(erie.diff,pch=19,cex=0.5)
```



This series is more plausibly stationary, so we examine the ACF

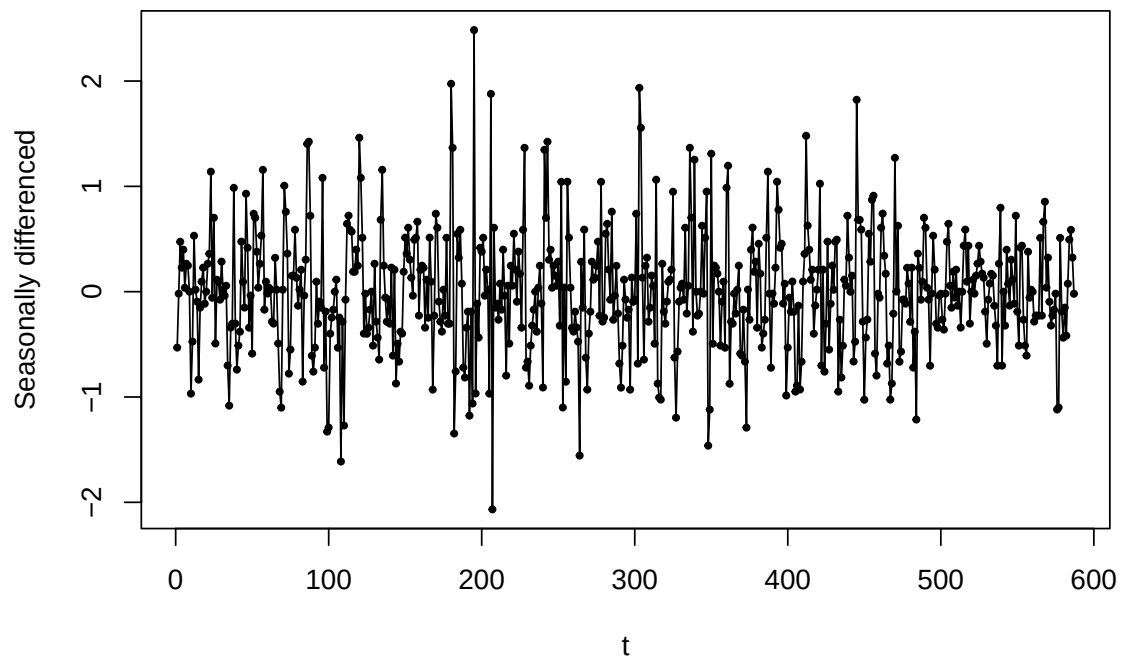
```
par(mar=c(4,4,3,2))
acf(erie.diff,lag.max=50,main='ACF of Erie data')
```

ACF of Erie data

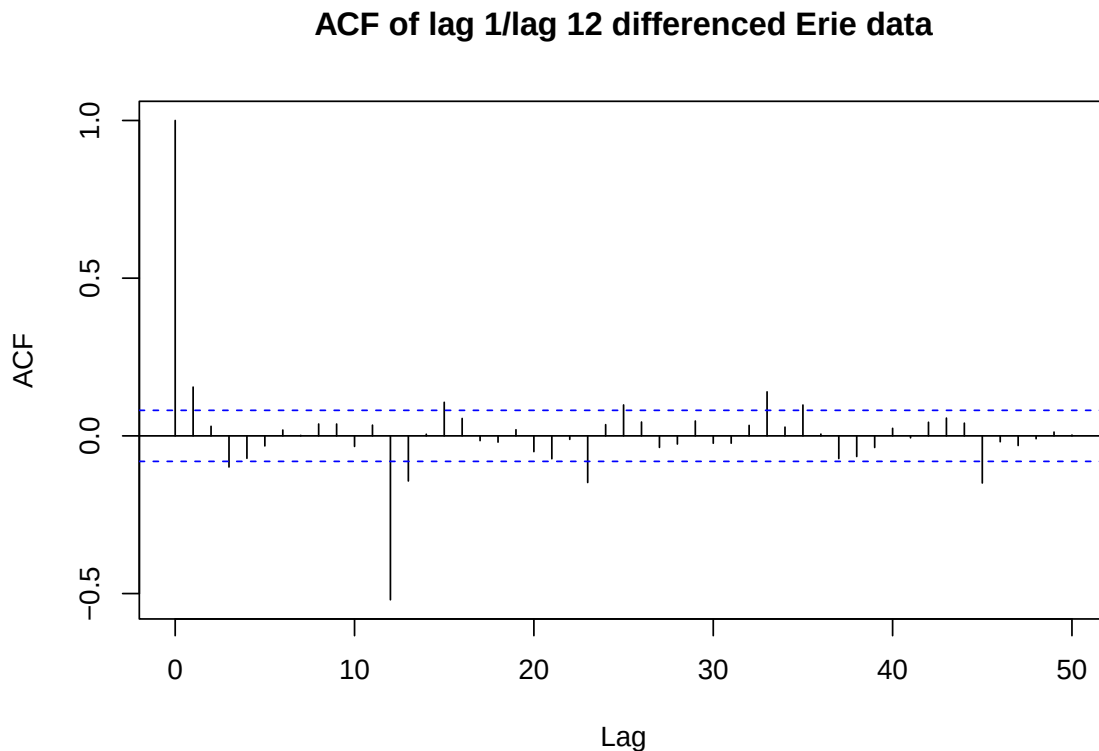


This suggests an analysis of the lag-12 differenced data may be more appropriate.

```
erie.diff<-diff(erie.diff, 12)
par(mar=c(4,4,3,2))
plot(erie.diff,type='l',xlab='t',ylab='Seasonally differenced')
points(erie.diff,pch=19,cex=0.5)
```



```
acf(erie.diff,lag.max=50,main='ACF of lag 1/lag 12 differenced Erie data')
```



We can conclude that if $\{X_t\}$ denotes the original data, then

$$Y_t = (1 - B^{12})(1 - B)X_t$$

is a plausibly stationary series, albeit one that still exhibits some dependence at lag 1 and lag 12.

(b) *Attempt to characterize the stochastic (random) component of the data using a SARIMA model.*

Given the final ACF plot from (a), it seems that the $\text{SARIMA}(1, 1, 1) \times (1, 1, 1)_{12}$ model, or some simpler related model may provide a good fit. We try fitting several SARIMA models, and compare using the AIC and BIC model selection criteria.

```
fit.1111 <- arima(erie,c(1,1,1), seasonal=list(order=c(1,1,1),period=12))
fit.1101 <- arima(erie,c(1,1,1), seasonal=list(order=c(0,1,1),period=12))
fit.1001 <- arima(erie,c(1,1,0), seasonal=list(order=c(0,1,1),period=12))
fit.1010 <- arima(erie,c(1,1,0), seasonal=list(order=c(1,1,0),period=12))
fit.0101 <- arima(erie,c(0,1,1), seasonal=list(order=c(0,1,1),period=12))
AIC(fit.1111,fit.1101,fit.1001,fit.1010,fit.0101)
```

```
##          df          AIC
## fit.1111   5 651.2898
## fit.1101   4 649.5151
## fit.1001   3 647.7138
## fit.1010   3 819.0623
## fit.0101   3 650.1119
```

```
AIC(fit.1111,fit.1101,fit.1001,fit.1010,fit.0101,k=log(length(erie)))
```

```
##          df          AIC
## fit.1111    5 673.2745
## fit.1101    4 667.1029
## fit.1001    3 660.9046
## fit.1010    3 832.2530
## fit.0101    3 663.3026
```

The results are as follows:

Model	AIC	BIC
SARIMA(1, 1, 1) $\times$ (1, 1, 1) ₁₂	651.2898363	673.2744845
SARIMA(1, 1, 1) $\times$ (0, 1, 1) ₁₂	649.5151454	667.1028641
SARIMA(1, 1, 0) $\times$ (0, 1, 1) ₁₂	647.7138238	660.9046127
SARIMA(1, 1, 0) $\times$ (1, 1, 0) ₁₂	819.062258	832.253047
SARIMA(0, 1, 1) $\times$ (0, 1, 1) ₁₂	650.1118575	663.3026465

and hence the selected model seems to be SARIMA(1, 1, 0) $\times$ (0, 1, 1)₁₂ as this has the lowest AIC and BIC.

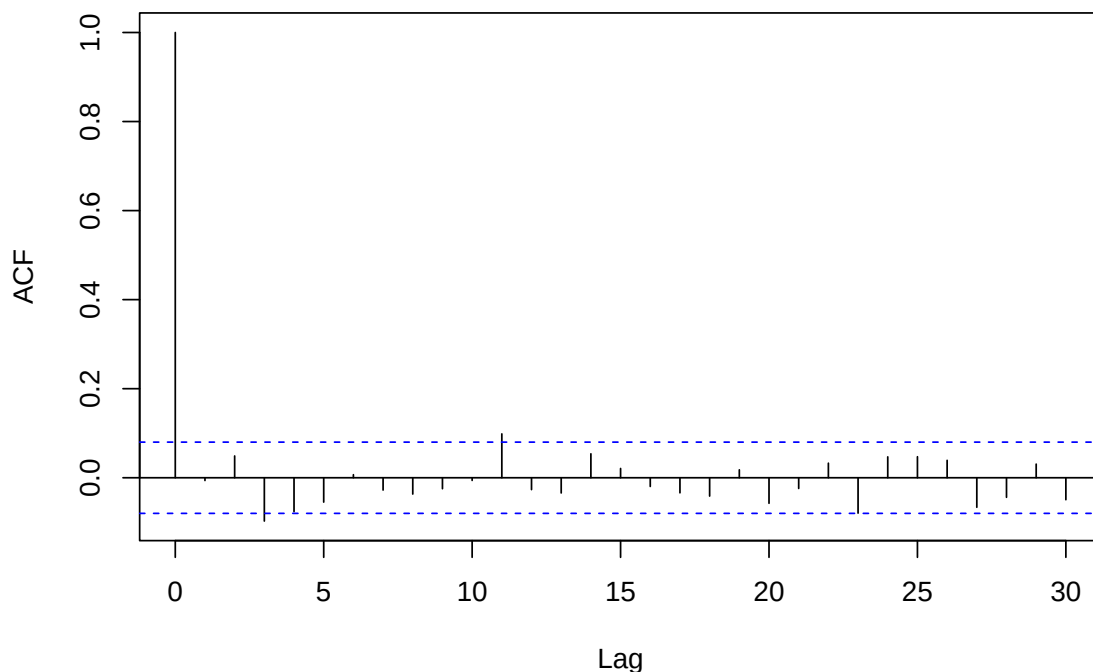
```
coef(fit.1001)
```

```
##          ar1          sma1
## 0.1903099 -0.9621257
```

If we take the residuals from this model, we see that the white noise requirement is approximately met

```
acf(residuals(fit.1001), lag.max=30, main='ACF of residuals')
```

ACF of residuals



- (c) Using the code provided in *knitr* sheets, formulate and fit a seasonal state space model to these data. It seems here that a local-level model plus a stochastic seasonal component with period 12 should be adequate. That is,

$$Y_t = M_t + S_t + V_t$$

with

$$M_t = M_{t-1} + W_{1,t-1}$$

yielding state vector $\mathbf{X}_t = [M_t \ S_t \ S_{t-1} \ \cdots \ S_{t-10}]^\top$ and

$$Y_t = \mathbf{F}\mathbf{X}_t + V_t$$

$$\mathbf{X}_{t+1} = \mathbf{G}\mathbf{X}_t + \mathbf{W}_t$$

with $\mathbf{F} = [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^\top$ and

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

where $\{V_t\} \sim \text{WN}(0, \sigma_V^2)$ and $\{\mathbf{W}_t\} \sim \text{WN}(0, \Omega_t)$ with $\Omega_t = \text{diag}(\sigma_1^2, \sigma_2^2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$.

```
##Compute Estimates

xv<-c(0,0,0,y[1],rnorm(11,10,0.2))
Erie.ml<-optim(xv,ml.filter.new,Fm=F1,Gm=G1,nv=n,Kval=K,Lval=L,yv=y,method="BFGS",
               control=list(trace=10,fnscale=-500,maxit=500,abstol=1e-6))

+ initial value 6.367205
+ iter 10 value 2.585567
+ iter 20 value 0.694063
+ final value -0.017462
+ converged

xv<-Erie.ml$par
Erie.ml

+ $par
+ [1] -2.1090453 -5.9553383 -5.3766594 15.4686035 -1.6110112 -0.7952766
+ [7] -0.4971384 -0.3006482 0.6402625 1.0762677 1.4778839 1.4181224
+ [13] 1.0508278 0.3320717 0.5660853
+
+ $value
+ [1] 8.731248
+
+ $counts
+ function gradient
+ 58 26
```

```

+
+ $convergence
+ [1] 0
+
+ $message
+ NULL

X.t<-matrix(0,nrow=length(y),ncol=length(Erie.ml$par)-3)
X.t[1,<-Erie.ml$par[4:length(Erie.ml$par)]
sig.1<-exp(Erie.ml$par[1])
sig.2<-exp(Erie.ml$par[2])
sig.v<-exp(Erie.ml$par[3])

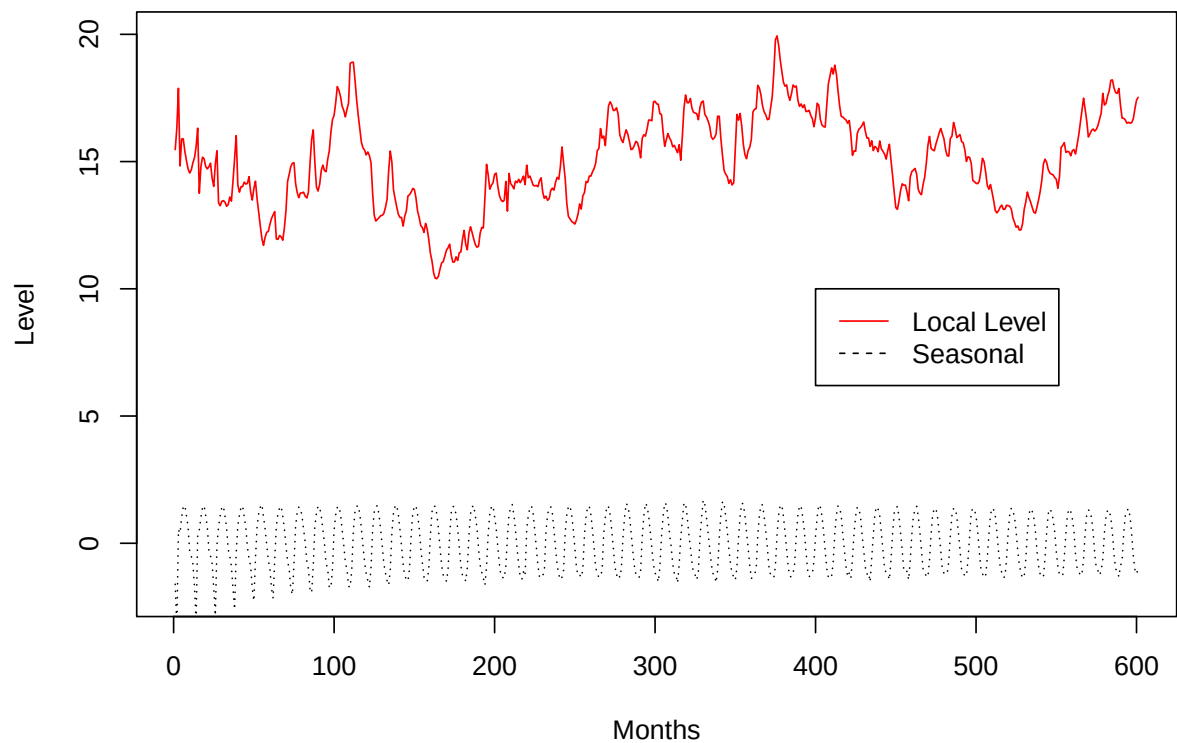
```

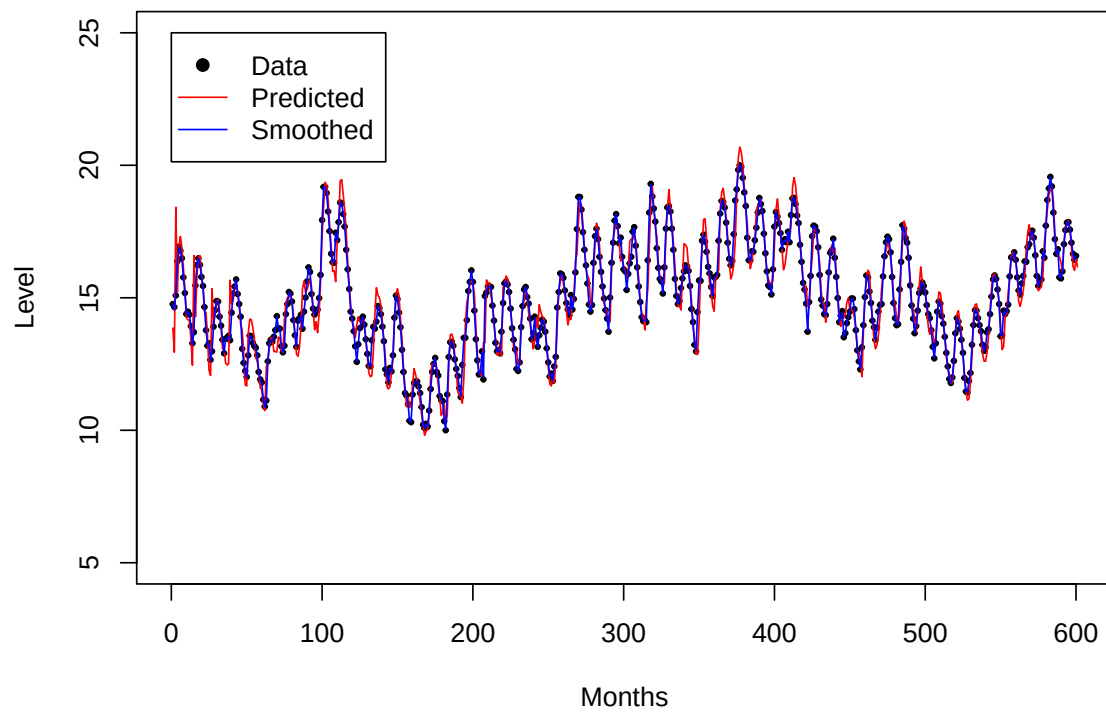
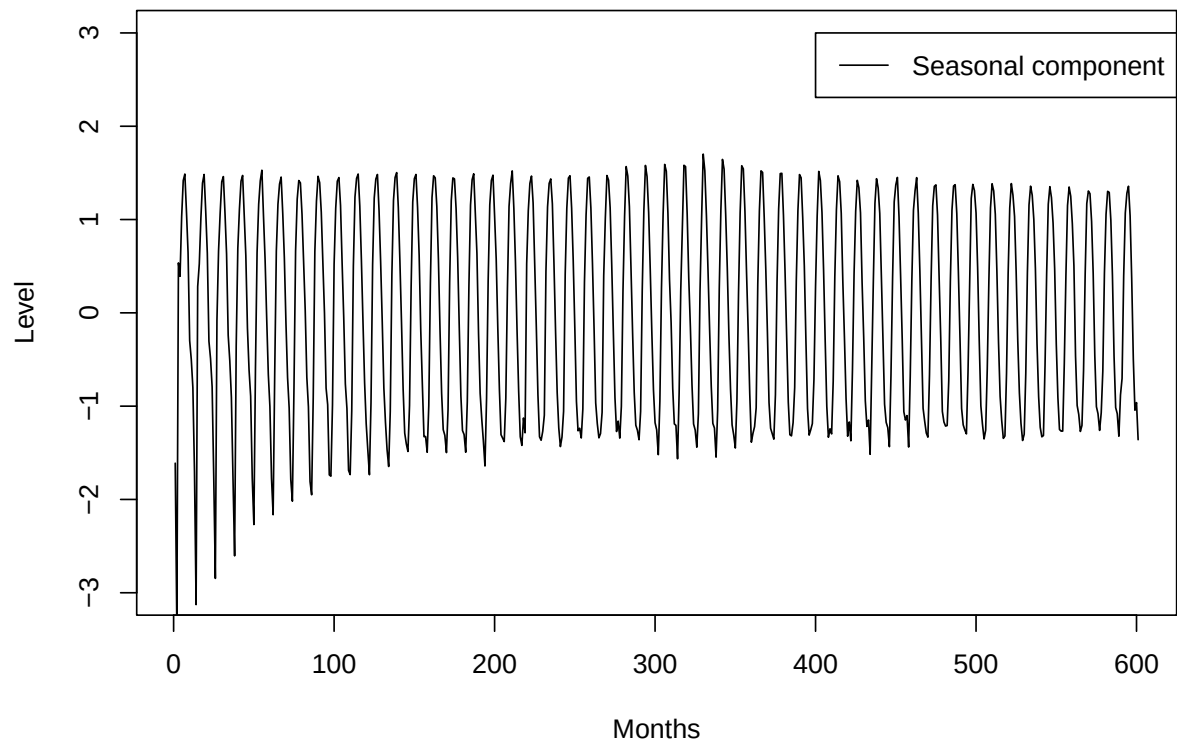
For this model the estimates are

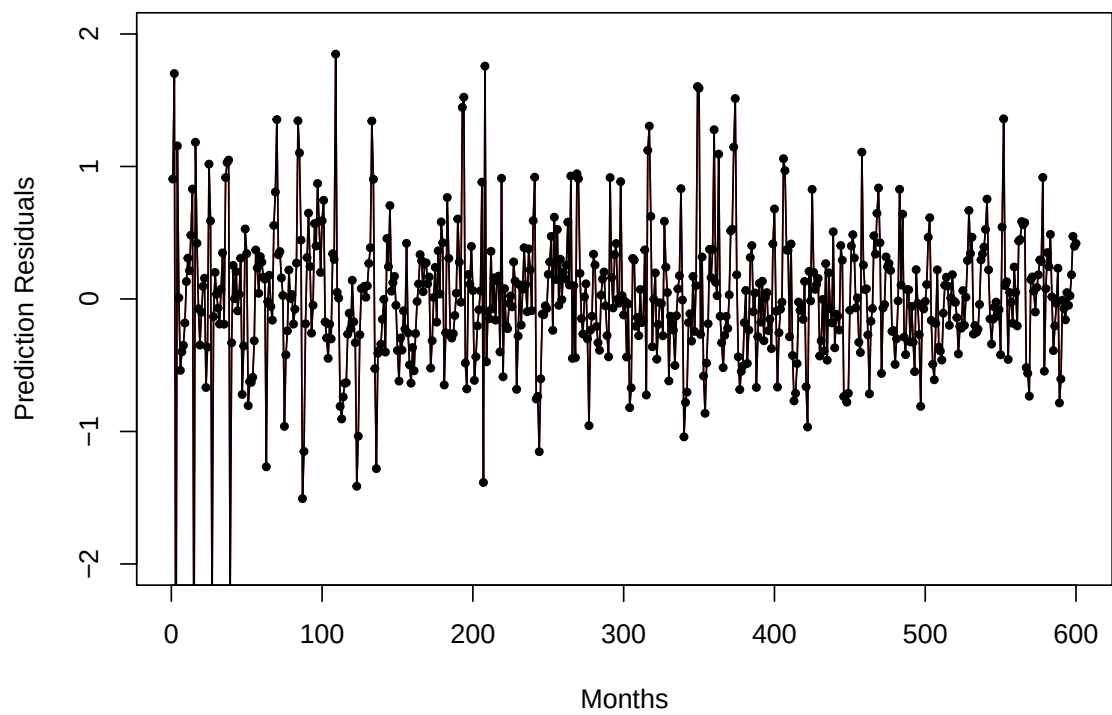
$$\sigma_1^2 = 0.1213538 \quad \sigma_2^2 = 0.002592 \quad \sigma_V^2 = 0.0046232$$

with seasonal component estimates

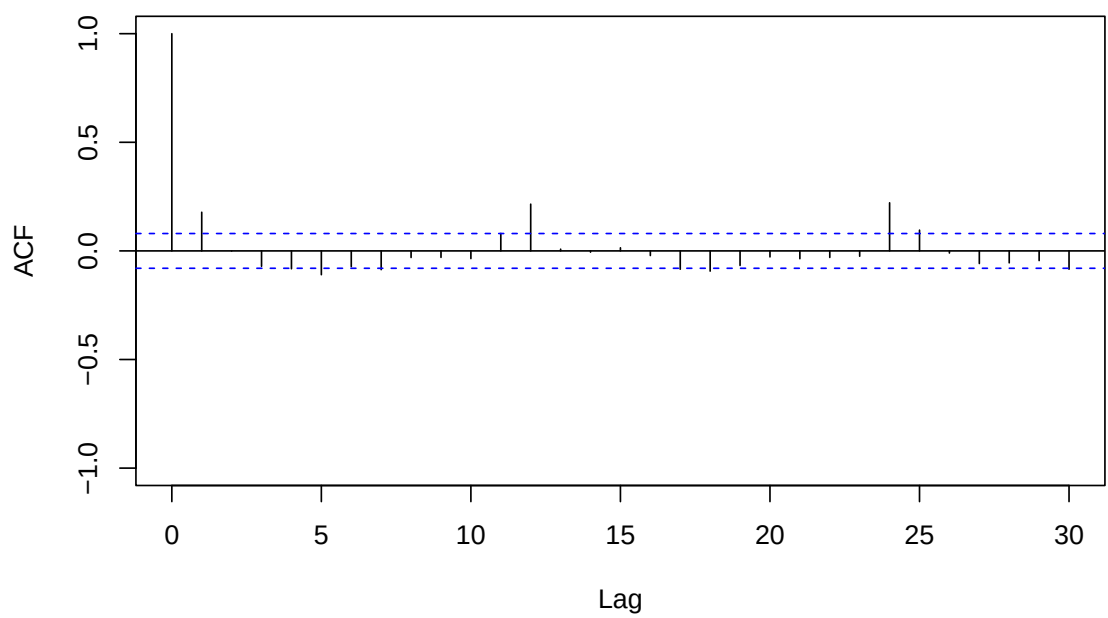
$$15.47, -1.61, -0.8, -0.5, -0.3, 0.64, 1.08, 1.48, 1.42, 1.05, 0.33, 0.57.$$

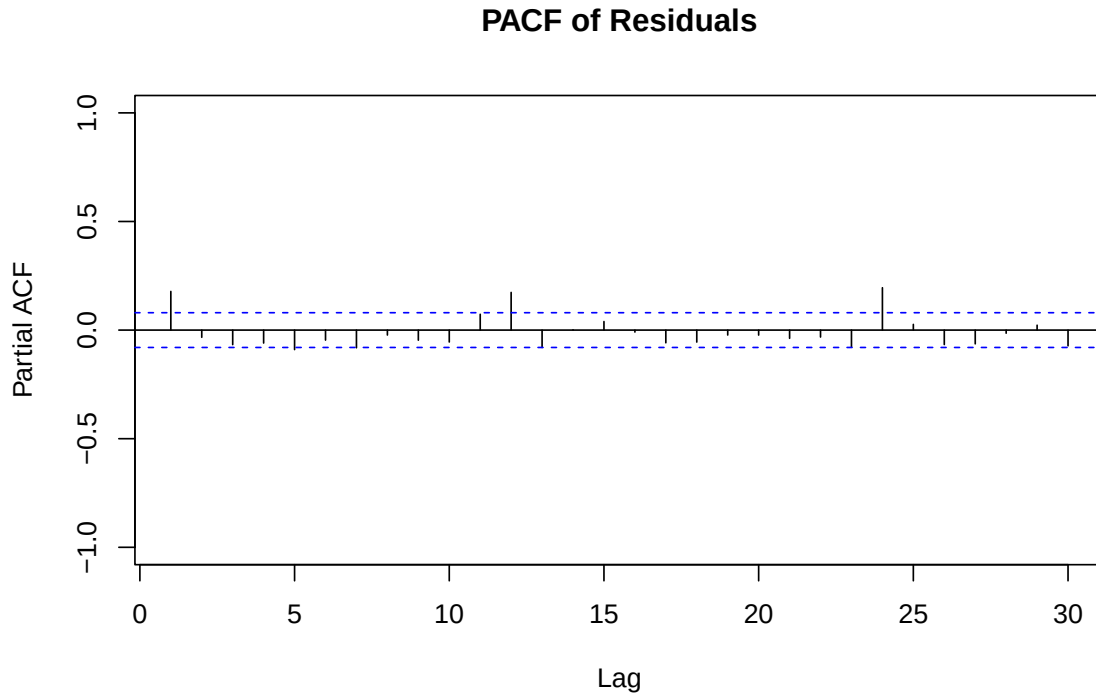






ACF of Residuals





2. Consider the following state space model for scalar observations $Y_{1:n}$ and states $X_{1:n}$

$$\begin{aligned} Y_t &= \beta X_t + V_t \\ X_{t+1} &= \phi X_t + W_t \end{aligned}$$

where $\{V_t\}$ and $\{W_t\}$ are independent Gaussian white noise series with variances σ_V^2 and σ_W^2 respectively. To initiate the model, suppose that $X_1 \sim \text{Normal}(0, \sigma_W^2/(1 - \phi^2))$.

(a) Under what conditions is $\{Y_t\}$ stationary ?

Solution: Provided $\phi \neq 1$, we have that $\mathbb{E}[Y_t] = \mathbb{E}[X_t] = 0$,

$$\text{Var}[Y_t] = \beta^2 \text{Var}[X_t] + \sigma_V^2 = \frac{\beta^2}{1 - \phi^2} \sigma_W^2 + \sigma_V^2$$

which does not depend on t , and that for $h \neq 0$

$$\begin{aligned} \text{Cov}[Y_t, Y_{t+h}] &= \text{Cov}[(\beta X_t + V_t), (\beta X_{t+h} + V_{t+h})] \\ &= \beta^2 \text{Cov}[X_t, X_{t+h}] \quad \text{other terms zero.} \end{aligned}$$

Thus $\{Y_t\}$ is stationary if $\{X_t\}$ is stationary, which holds provided $\phi \neq 1$.

(b) Assuming $\{Y_t\}$ is stationary, derive its ACVF, $\gamma_Y(h)$.

Solution: In this case, from properties of the stationary AR(1) process, we have from part (a) that

$$\gamma_Y(h) = \frac{\beta^2 \phi^h}{1 - \phi^2} \sigma_W^2$$

for $h = 0, \pm 1, \pm 2, \dots$

- (c) Derive the Kalman filtering recursions for this model, based on the filtering and prediction distributions $p(x_t|y_{1:t})$ and $p(x_{t+1}|y_{1:t})$. Compute also the prediction distribution $p(y_{t+1}|y_{1:t})$.

Solutions: We have two components of the recursion. Using the notation and results from lectures, we have for the scalar state process $\{X_t\}$ the filtering and prediction distributions

$$p(x_t|y_{1:t}) \equiv \text{Normal}(a_{t|t}, P_{t|t}) \quad p(x_{t+1}|y_{1:t}) \equiv \text{Normal}(a_{t+1|t}, P_{t+1|t}).$$

- **Filtering:** we have that

$$p(x_t|y_{1:t}) \propto p(y_t|x_t)p(x_t|y_{1:(t-1)})$$

where

$$p(y_t|x_t) \equiv \text{Normal}(\beta x_t, \sigma_V^2) \quad p(x_t|y_{1:(t-1)}) \equiv \text{Normal}(a_{t|t-1}, P_{t|t-1})$$

In the relevant quadratic component of the calculation, we have

$$\frac{1}{\sigma_V^2}(y_t - \beta x_t)^2 + \frac{1}{P_{t|t-1}}(x_t - a_{t|t-1})^2$$

and rearranging, this yields

$$\frac{\beta^2}{\sigma_V^2}(x_t - y_t/\beta)^2 + \frac{1}{P_{t|t-1}}(x_t - a_{t|t-1})^2.$$

Using the standard complete-the-square formula

$$A(x - a)^2 + B(x - b)^2 = (A + B) \left(x - \frac{Aa + Bb}{A + B} \right)^2 + \frac{AB}{A + B}(a - b)^2$$

may conclude that

$$a_{t|t} = \frac{(\beta y_t / \sigma_V^2) + (a_{t|t-1} / P_{t|t-1})}{(\beta^2 / \sigma_V^2) + (1 / P_{t|t-1})}$$

and

$$P_{t|t} = \left(\frac{\beta^2}{\sigma_V^2} + \frac{1}{P_{t|t-1}} \right)^{-1}$$

- **Prediction:** We have that

$$p(x_{t+1}|y_{1:t}) = \int p(x_{t+1}|x_t)p(x_t|y_{1:t}) dx_t.$$

Taking expectations in the state equation, we have that

$$a_{t+1|t} = \mathbb{E}_{X_t|Y_{1:t}}[\phi X_t + W_t|y_{1:t}] = \phi \mathbb{E}_{X_t|Y_{1:t}}[X_t|y_{1:t}] = \phi a_{t|t}$$

and

$$P_{t+1|t} = \text{Var}_{X_t|Y_{1:t}}[\phi X_t + W_t|y_{1:t}] = \phi^2 \text{Var}_{X_t|Y_{1:t}}[X_t|y_{1:t}] + \sigma_W^2 = \phi^2 P_{t|t} + \sigma_W^2.$$

Finally, from the observation equation, we have that

$$\mathbb{E}_{Y_{t+1}|Y_{1:t}}[Y_{t+1}|y_{1:t}] = \mathbb{E}_{Y_{t+1}|Y_{1:t}}[\beta X_{t+1} + V_t|y_{1:t}] = \beta a_{t+1|t}$$

and

$$\text{Var}_{Y_{t+1}|Y_{1:t}}[Y_{t+1}|y_{1:t}] = \text{Var}_{Y_{t+1}|Y_{1:t}}[\beta X_{t+1} + V_t|y_{1:t}] = \beta^2 \text{Var}_{Y_{t+1}|Y_{1:t}}[X_{t+1}|y_{1:t}] + \sigma_V^2 = \beta^2 P_{t+1|t} + \sigma_V^2$$

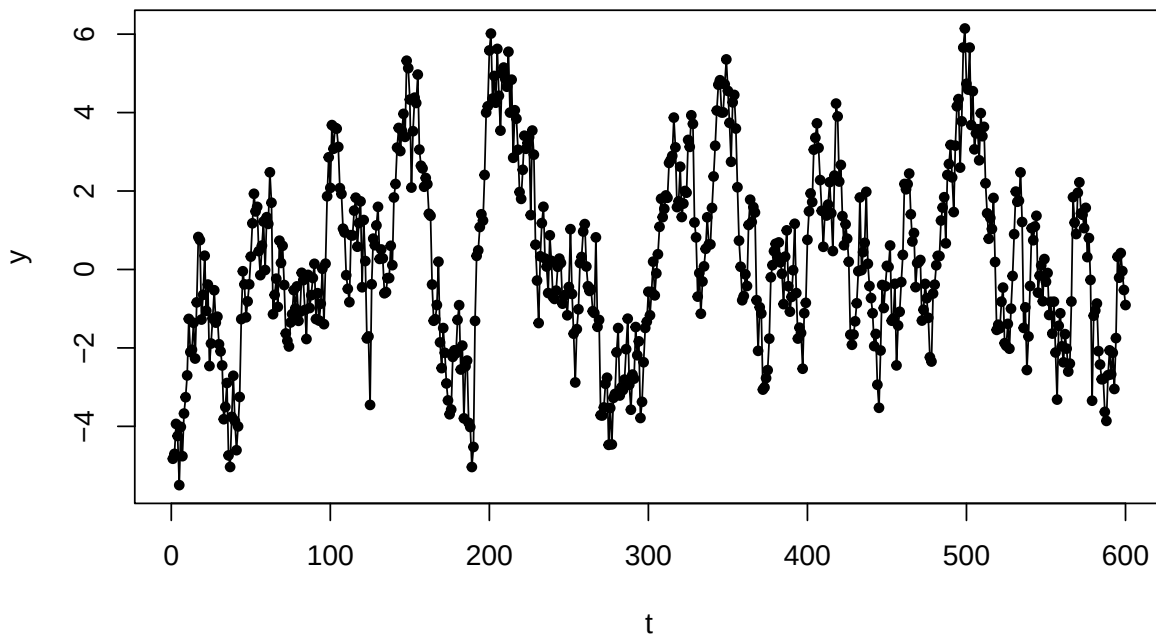
and so

$$p(y_{t+1}|y_{1:t}) \equiv \text{Normal}(\beta a_{t+1|t}, \beta^2 P_{t+1|t} + \sigma_V^2)$$

- (d) Using code similar to that provided in *knitr* sheets, or otherwise, perform maximum likelihood estimation of the system parameters for the data in the *Ex3-Q2data.txt*

<https://www.math.mcgill.ca/dstephens/545/Data/Ex3-Q2data.txt>

Solution:



```
#Perform the optimization
ml.filter.new<-function(x,nv,Lval,Kval,yv){

  Yt<-yv
  Sigmat<-exp(x[2])
  Omegat<-exp(x[1])

  att<-array(0,c(nv,Lval))
  apt<-array(0,c(nv+1,Lval))
  Ptt<-array(0,c(nv,Lval,Lval))
  Ppt<-array(0,c(nv+1,Lval,Lval))
  rt<-array(0,c(nv,Kval))
  Mt<-array(0,c(nv,Kval,Kval))

  apt[1,]<-x[5]
  Ppt[1,]<-Omegat

  Fm<-matrix(x[3],1,1)
  Gm<-matrix(x[4],1,1)

  #Filter Forward

  loglike<-0
  for(i in 1:nv){

    Fmat<-matrix(Fm,nrow=Kval,ncol=Lval); tFmat<-t(Fmat)
```

```

rt[i,]<-Yt[i] - Fmat %*% apt[i,]
Mt[i,,]<-Fmat %*% Ppt[i,,] %*% t(Fmat) + Sigmat

if(Sigmat == Inf || Mt[i,,] == 0 || is.na(Mt[i,,])){
  return(Inf)
}
Mtinv<-solve(Mt[i,,])

att[i,]<-apt[i,] + Ppt[i,,] %*% tFmat %*% Mtinv %*% rt[i,]
Ptt[i,,]<-Ppt[i,,]-Ppt[i,,] %*% tFmat %*% Mtinv %*% Fmat %*% Ppt[i,,]
apt[i+1,]<-Gm %*% att[i,]
Ppt[i+1,,]<-Gm %*% Ptt[i,,] %*% t(Gm) + Omegat

loglike<-loglike-0.5*log(det(matrix(Mt[i,,],Kval,Kval)))-
  0.5*t(rt[i,]) %*% Mtinv %*% rt[i,]
}

return(as.numeric(loglike))
}

```

```
##Compute Estimates
```

```

xv<-c(0.1,0.1,0,0.5,0.5)
Q2.ml<-optim(xv,ml.filter.new,nv=n,Kval=K,Lval=L,yv=y,
  method="BFGS",control=list(trace=10,fnscale=-500,maxit=500,abstol=1e-6))

+ initial value 2.984770
+ iter 10 value 1.550739
+ iter 20 value 0.569878
+ iter 30 value 0.558864
+ iter 40 value 0.558741
+ iter 50 value 0.558693
+ iter 60 value 0.558685
+ iter 70 value 0.558680
+ iter 80 value 0.558674
+ iter 90 value 0.558660
+ iter 100 value 0.558660
+ iter 110 value 0.558658
+ final value 0.558658
+ converged

xv<-Q2.ml$par
Q2.ml

+ $par
+ [1] -2.211527 -3.3472934 -2.8185828 0.9112386 1.7144779
+
+ $value
+ [1] -279.3288
+
+ $counts

```

```

+ function gradient
+   126   112
+
+ $convergence
+ [1] 0
+
+ $message
+ NULL

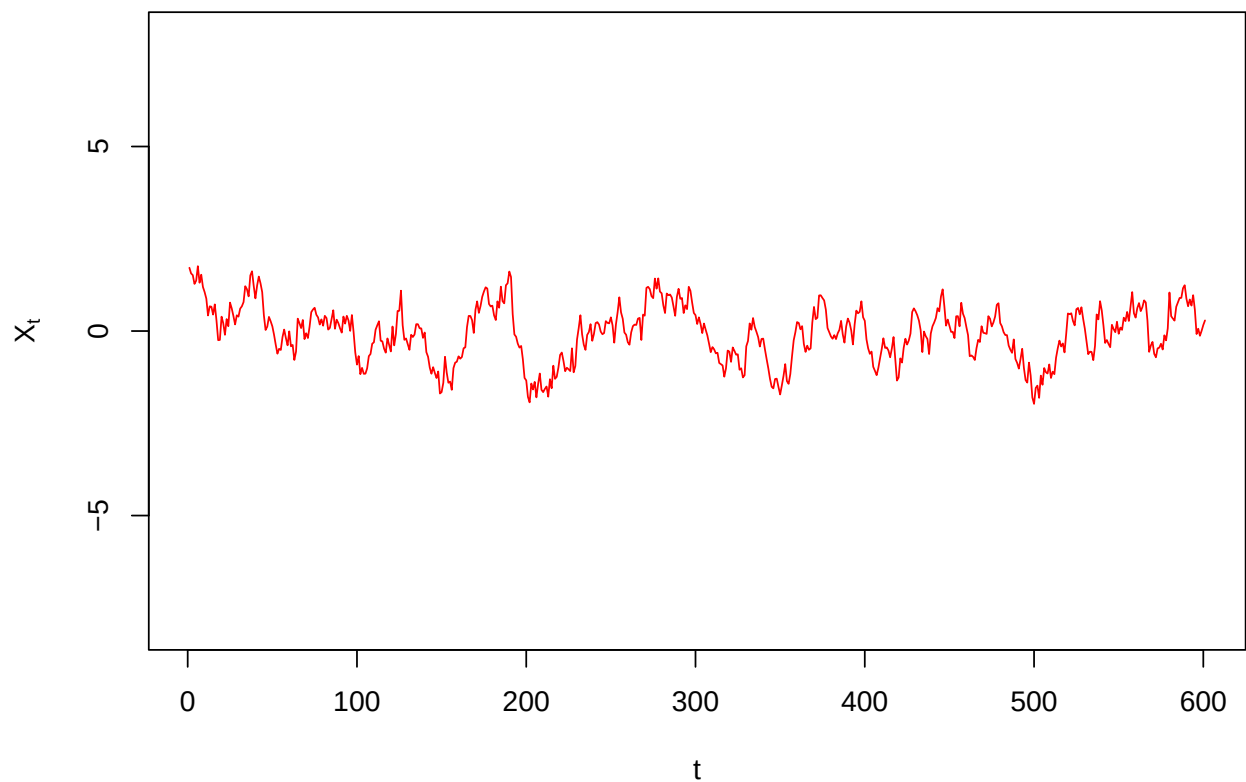
X.t<-matrix(0,nrow=length(y),ncol=1)
X.t[1,<-Q2.ml$par[5]
sig.W<-exp(Q2.ml$par[1])
sig.V<-exp(Q2.ml$par[2])
beta.hat<-Q2.ml$par[3]
phi.hat<-Q2.ml$par[4]

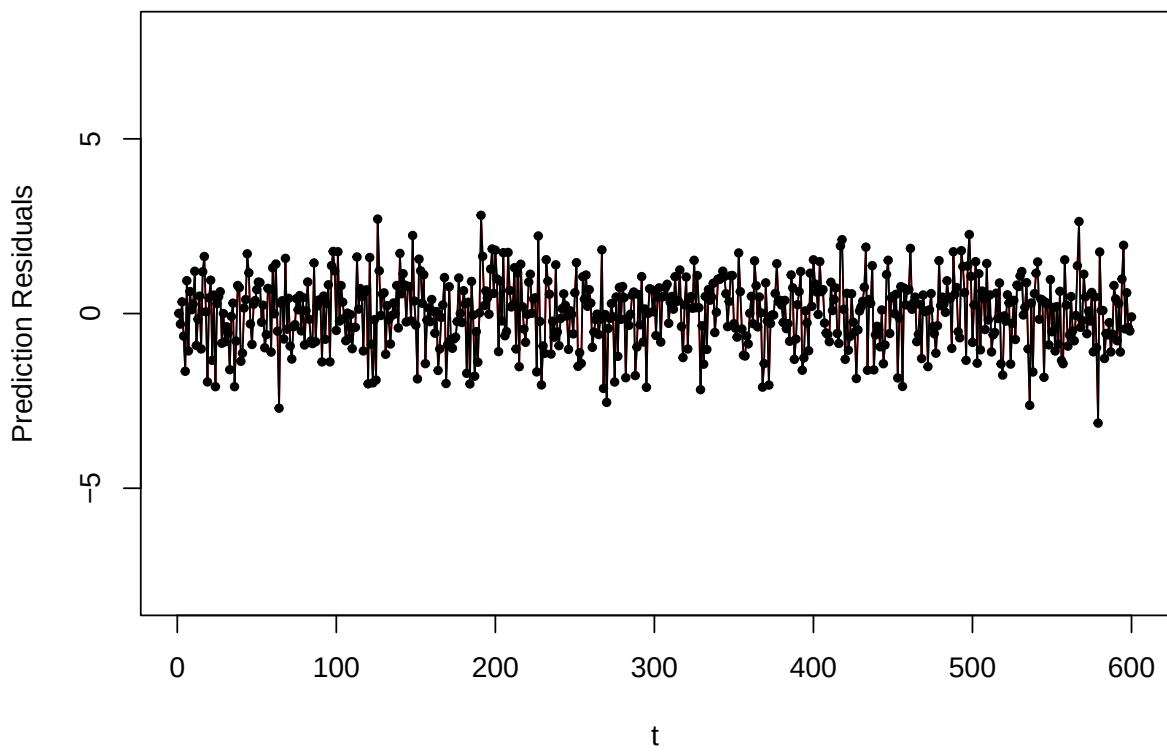
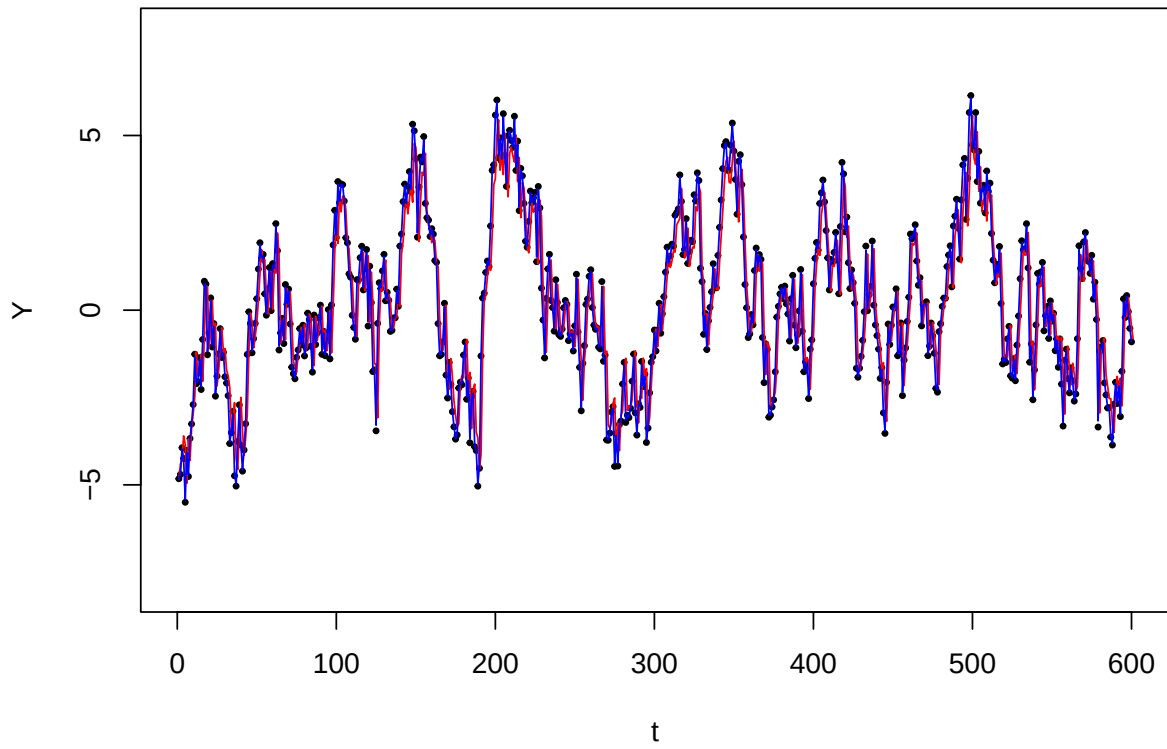
```

For this model the estimates are $\beta = -2.8185828$, $\phi = 0.9112386$

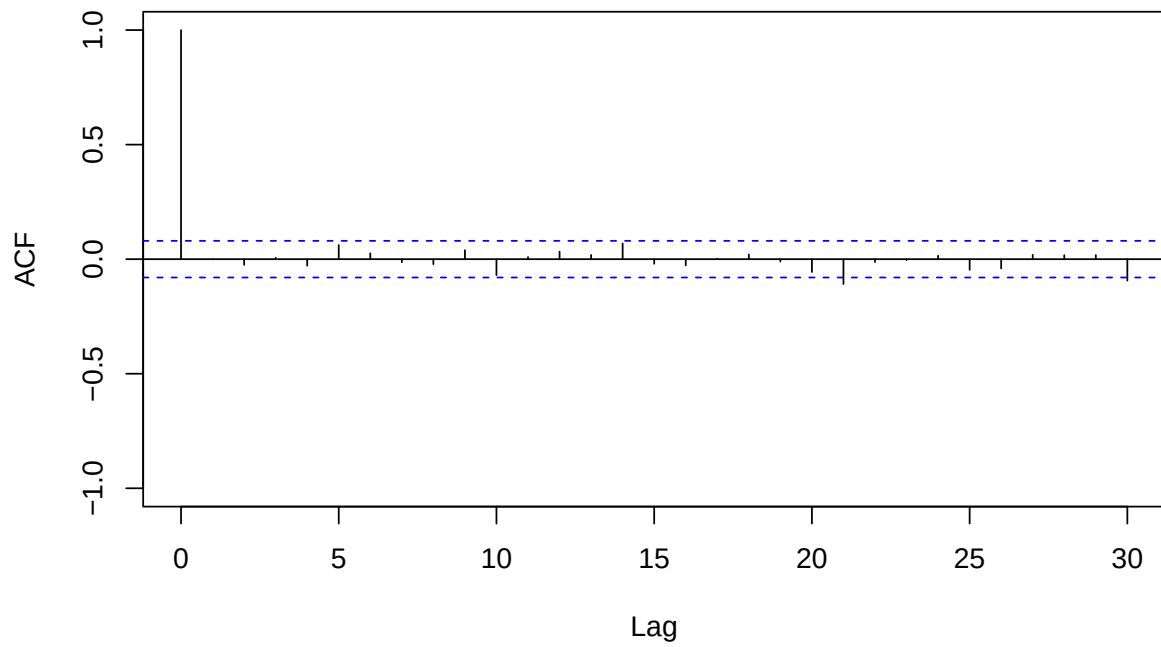
$$\sigma_W^2 = 0.1095304 \quad \sigma_2^2 = 0.0351794$$

with initial value $X_1 = 1.7144779$.

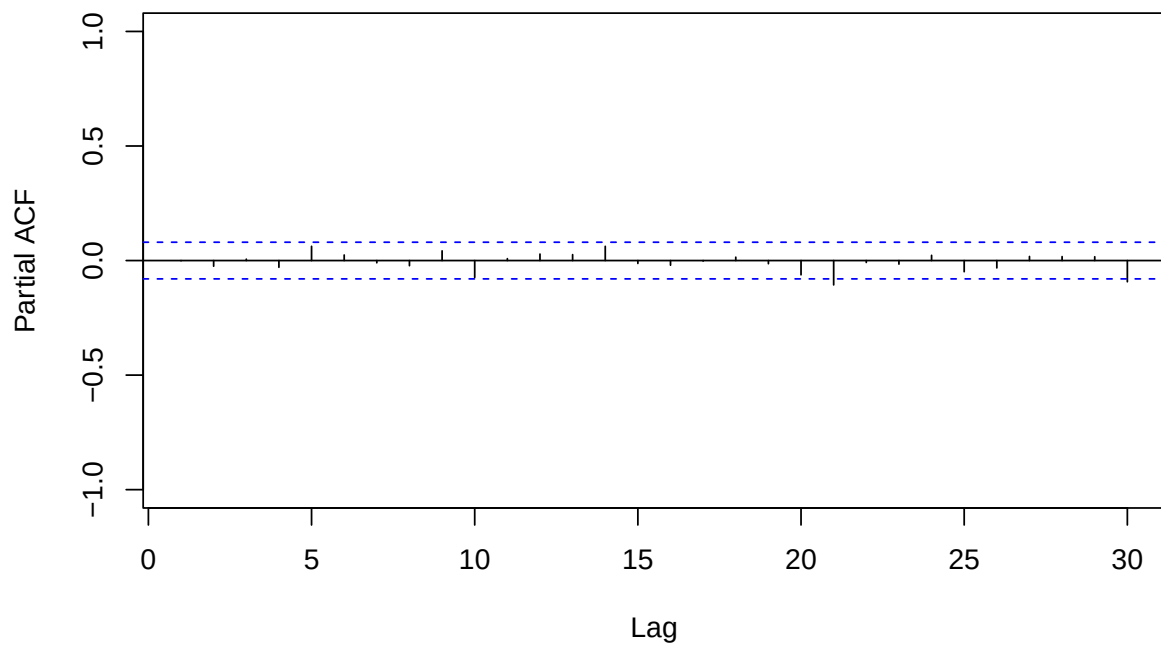




ACF of Residuals



PACF of Residuals



3. Formulate the stationary ARMA(2, 2) model

$$(1 - 0.5B)(1 - 0.9B)X_t = (1 + B)(1 + 2B)Z_t$$

as a state space model.

If it was required to add a **deterministic** seasonal component, $\{S_t\}$, with period 4 to this model, so that say

$$Y_t = X_t + S_t$$

how would the state space model be amended? Identify the corresponding G and F matrices, and the other system components.

Solution: We can write the model

$$X_t - 1.4X_{t-1} + 0.45X_{t-2} = Z_t + 3Z_{t-1} + 2Z_{t-2}$$

The formulation for ARMA(2, 2) is given as follows: let $m = \max\{p, q + 1\} = 3$, and

$$\begin{aligned} Y_t &= \mathbf{F}_t \mathbf{X}_t + V_t \\ \mathbf{X}_{t+1} &= \mathbf{G}_t \mathbf{X}_t + \mathbf{W}_t \end{aligned}$$

where we define the state $\mathbf{X}_t = (X_{1,t-2}, X_{1,t-1}, \dots, X_{1,t})^\top$, and $\mathbf{W}_t = (0, 0, 0, \dots, Z_{t+1})^\top$, so that

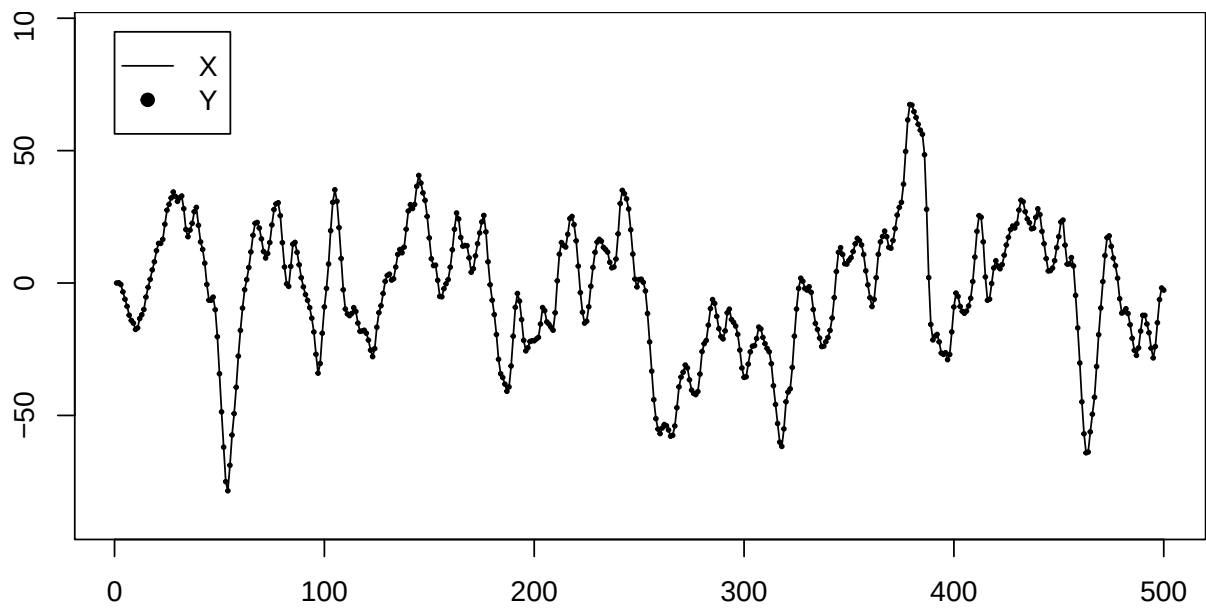
$$\mathbf{G}_t = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \phi_3 & \phi_2 & \phi_1 \end{bmatrix} \quad \mathbf{F}_t = [\theta_2 \quad \theta_1 \quad \theta_0].$$

with $V_t \equiv 0$ and $Z_t \sim \text{Normal}(0, \sigma_Z^2)$ and where

$$(\phi_1, \phi_2, \phi_3) = (1.4, -0.45, 0) \quad (\theta_0, \theta_1, \theta_2) = (1, 3, 2).$$

The following code verifies the simulation

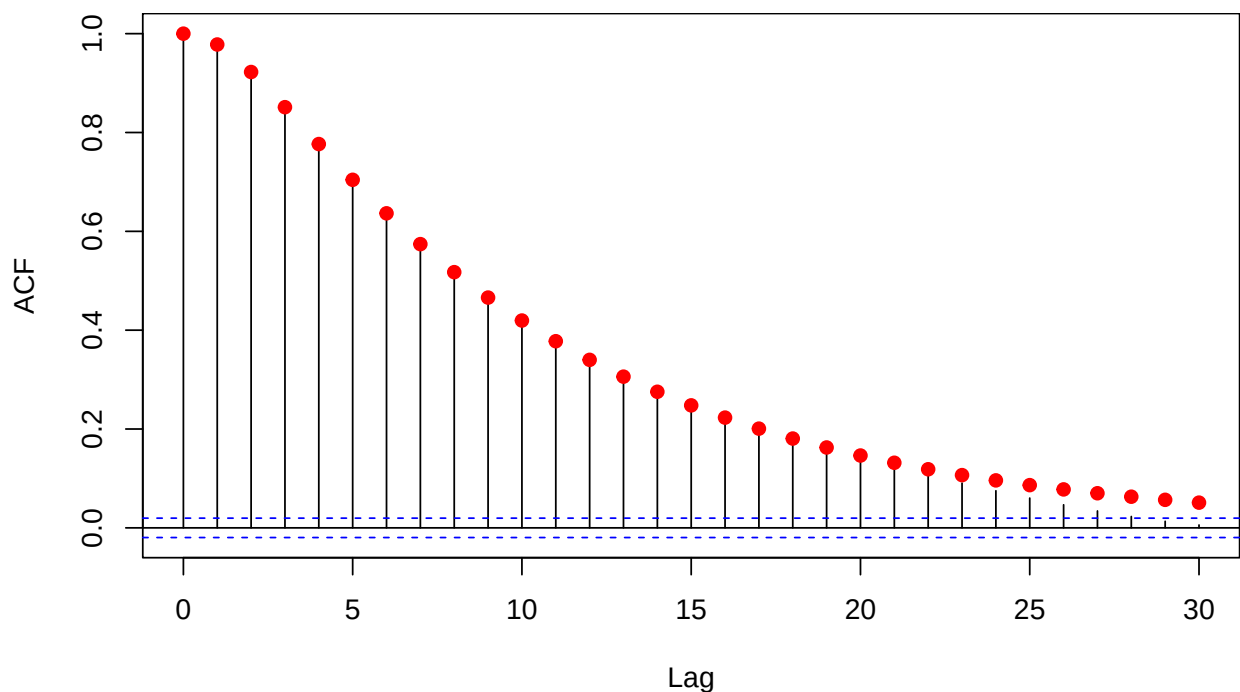
```
set.seed(3834); n<-10000; p<-2; q<-2; m<-max(p, q+1)
phi.vec<-theta.vec<-rep(0, m)
phi.vec[1:p]<-c(1.4, -0.45)
theta.vec[1:q]<-c(3, 2)
X<-matrix(0, ncol=n, nrow=m); Y<-mt<-rep(0, n)
F<-c(rev(theta.vec[1:(m-1)]), 1)
G<-matrix(0, m, m); for(j in 1:(m-1)){G[j, j+1]<-1}
G[m,]<-rev(phi.vec)
sigV<-0; sigZ<-1
V<-rnorm(n)*sigV; Z<-rnorm(n)*sigZ
X[, 1]<-rep(0, m)
Y[1]<-(F%*%X[, 1])[1, 1]+V[1]
for(i in 2:n){
  Wi<-c(rep(0, m-1), Z[i])
  X[, i]<-G%*%X[, i-1]+Wi
  mt[i]<-(F%*%X[, i])[1, 1]
  Y[i]<-mt[i]+V[i]
}
par(mar=c(2, 3, 0, 0)); plot(mt[1:500], type='l', ylim=range(Y));
points(Y[1:500], pch=19, cex=0.3)
legend(0, max(Y), c('X', 'Y'), col=c('black', 'black'), pch=c(-1, 19), lty=c(1, -1))
```



```
psi.vec<-ARMAtoMA(ar=phi.vec,ma=theta.vec,lag.max=30)
varY<-1+sum(psi.vec^2)
true.acf<-ARMAacf(ar=phi.vec,ma=theta.vec,lag.max=30)
true.acv<-true.acf*varY
varY;var(Y)
```

```
+ [1] 645.4781
+ [1] 664.6379
```

```
par(mar=c(4,4,0,0))
acf(Y,lag.max=30,main='ACF of Y');points(0:30,true.acf,col='red',pch=19)
```



The simulation using the state space form matches the target ACF.

To add the seasonal component, merely add the additional components to the state vector. In the deterministic seasonal model, we have that

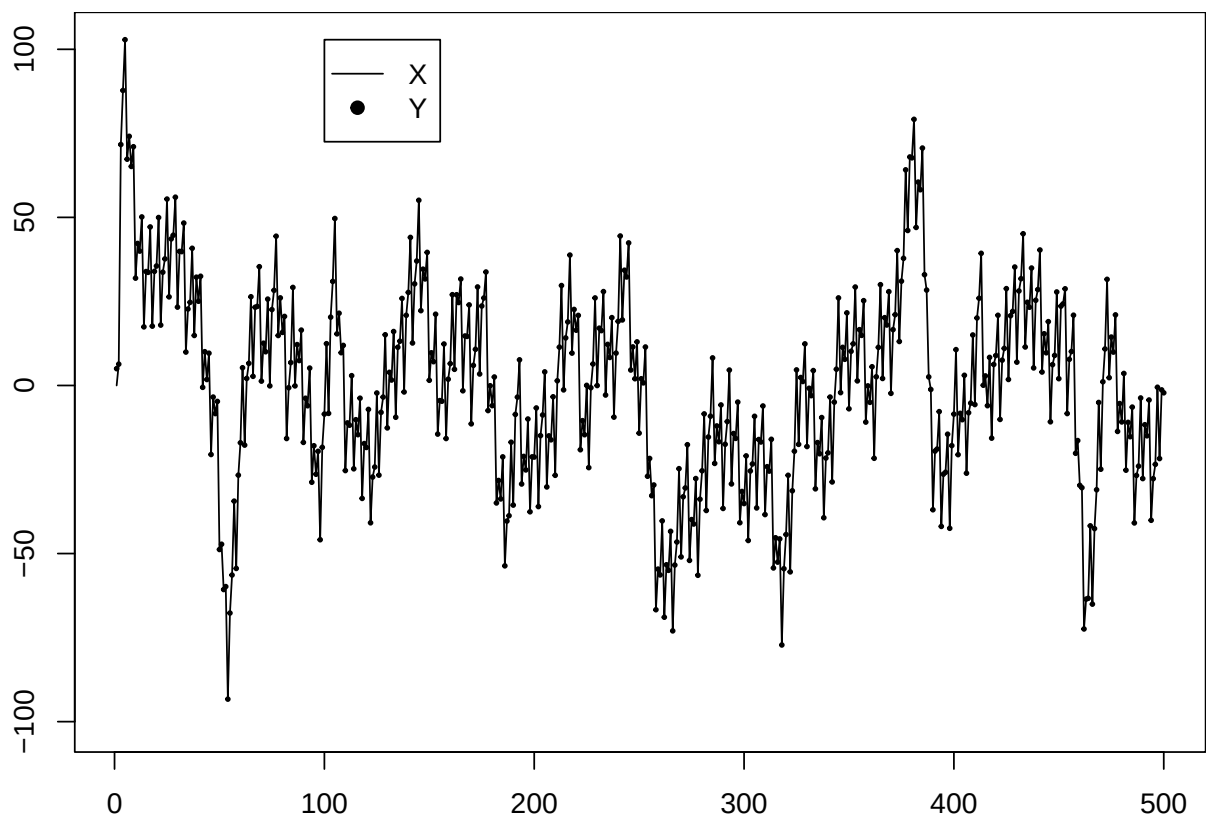
$$S_{t+1} = -S_t - S_{t-1} - S_{t-2}$$

so the state equation becomes

$$\begin{bmatrix} X_{1,t-1} \\ X_{1,t} \\ X_{1,t+1} \\ S_{t+1} \\ S_t \\ S_{t-1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \phi_2 & \phi_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{1,t-2} \\ X_{1,t-1} \\ X_{1,t} \\ S_t \\ S_{t-1} \\ S_{t-2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

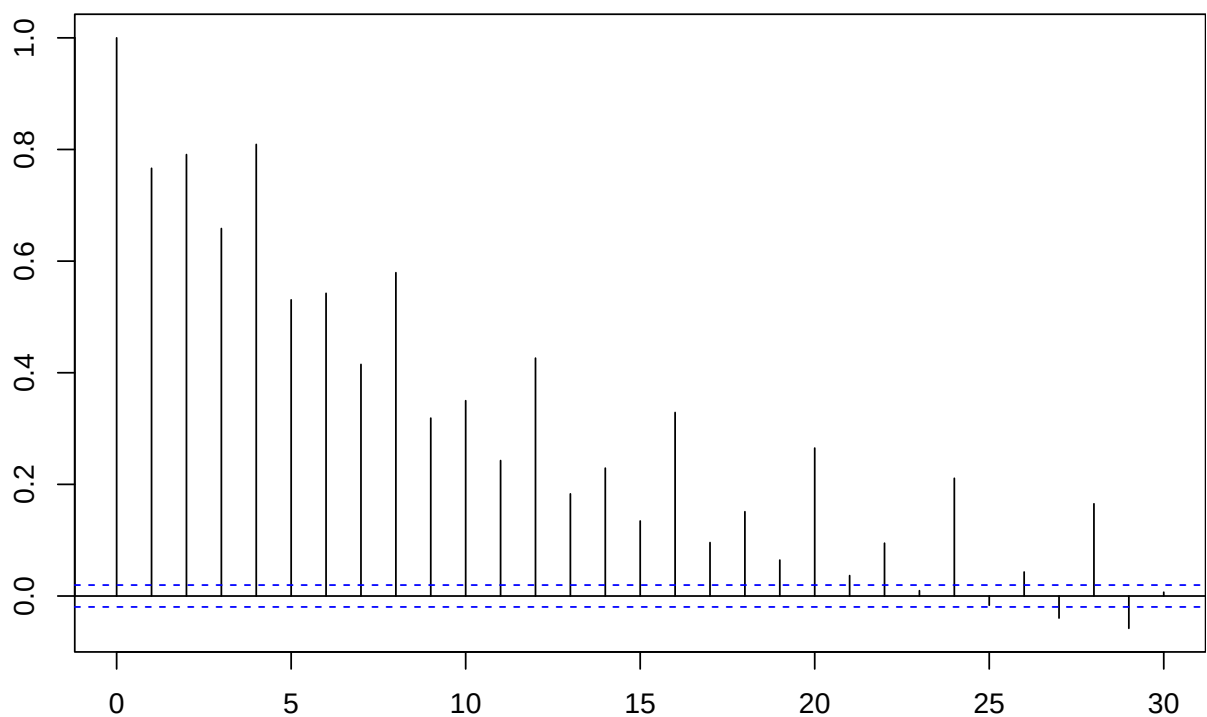
with $\mathbf{F} = [\theta_2 \ \theta_1 \ \theta_0 \ 1 \ 0 \ 0]$

```
set.seed(3834); n<-10000; p<-2; q<-2; m<-max(p,q+1)
phi.vec<-theta.vec<-rep(0,m)
phi.vec[1:p]<-c(1.4,-0.45)
theta.vec[1:q]<-c(3,2)
X<-matrix(0,ncol=n,nrow=m+3); Y<-mt<-rep(0,n)
F<-c(rev(theta.vec[1:q]),1,1,0,0)
G<-matrix(0,m+3,m+3); for(j in 1:(m-1)){G[j,j+1]<-1}
G[m,]<-rev(phi.vec)
G[m+1,m+c(1:3)]<--1
G[m+2,m+1]<-1
G[m+3,m+2]<-1
sigV<-0; sigZ<-1
V<-rnorm(n)*sigV; Z<-rnorm(n)*sigZ
X[,1]<-c(rep(0,m),5,-5,10)
Y[1]<-(F%*%X[,1])[1,1]+V[1]
for(i in 2:n){
  Wi<-c(rep(0,m-1),Z[i],rep(0,3))
  X[,i]<-G%*%X[,i-1]+Wi
  mt[i]<-(F%*%X[,i])[1,1]
  Y[i]<-mt[i]+V[i]
}
par(mar=c(2,3,0,0)); plot(mt[1:500],type='l',ylim=range(Y));
points(Y[1:500],pch=19,cex=0.3)
legend(100,max(Y),c('X','Y'),col=c('black','black'),pch=c(-1,19),lty=c(1,-1))
```



```
par(mar=c(2,3,3,0))
acf(Y,lag.max=30,main='ACF of Y')
```

ACF of Y

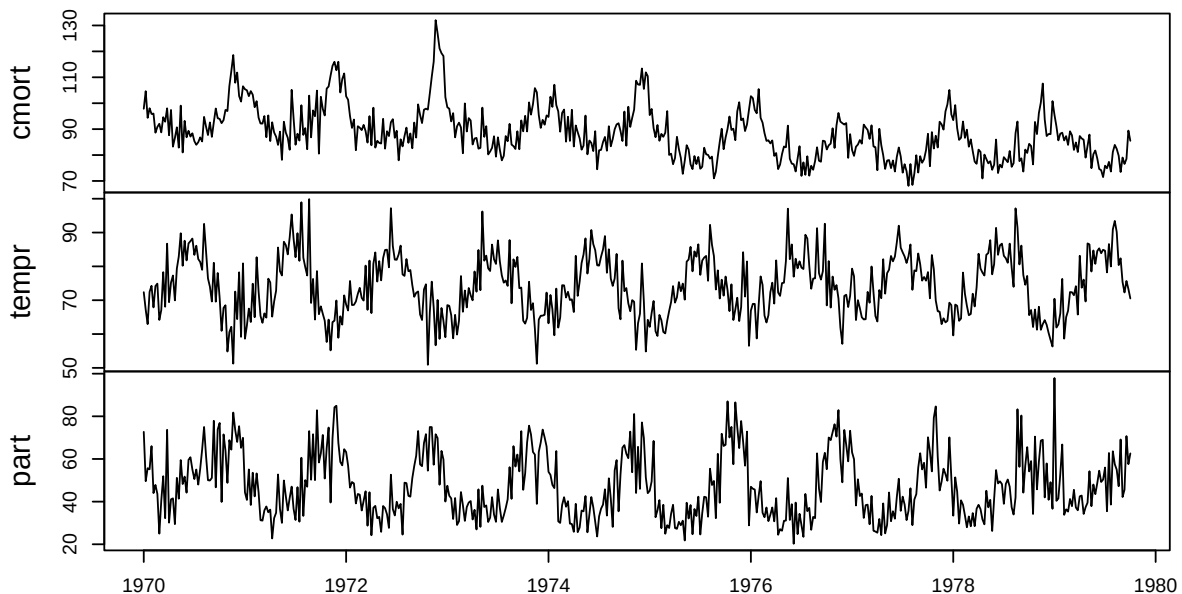


4. The following data from the *astsa* library comprise

- *cmort*: average weekly cardiovascular mortality in Los Angeles County, US. The data are 508 six-day averages obtained from daily values over the period 1970-1979;
- *tempr*: the corresponding weekly temperature data for the same period;
- *part*: a corresponding particulate matter series for the same period.

studied in the book by Shumway et al.

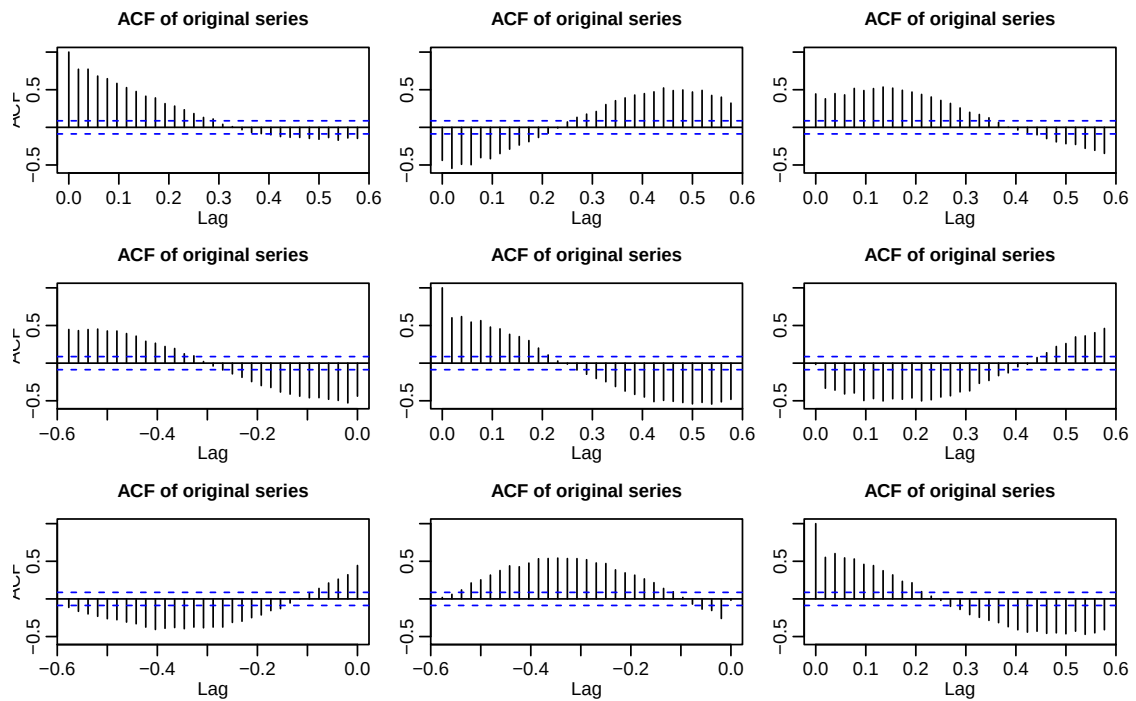
```
suppressMessages(library(vars));library(astsa)
x = cbind(cmort, tempr, part)
par(mar=c(4,4,0,2))
plot.ts(x , main = "", xlab = "")
```



- (a) Using the *VAR* function from the *vars* library, fit $VAR(p)$ models for various p , and compare them using the AIC.

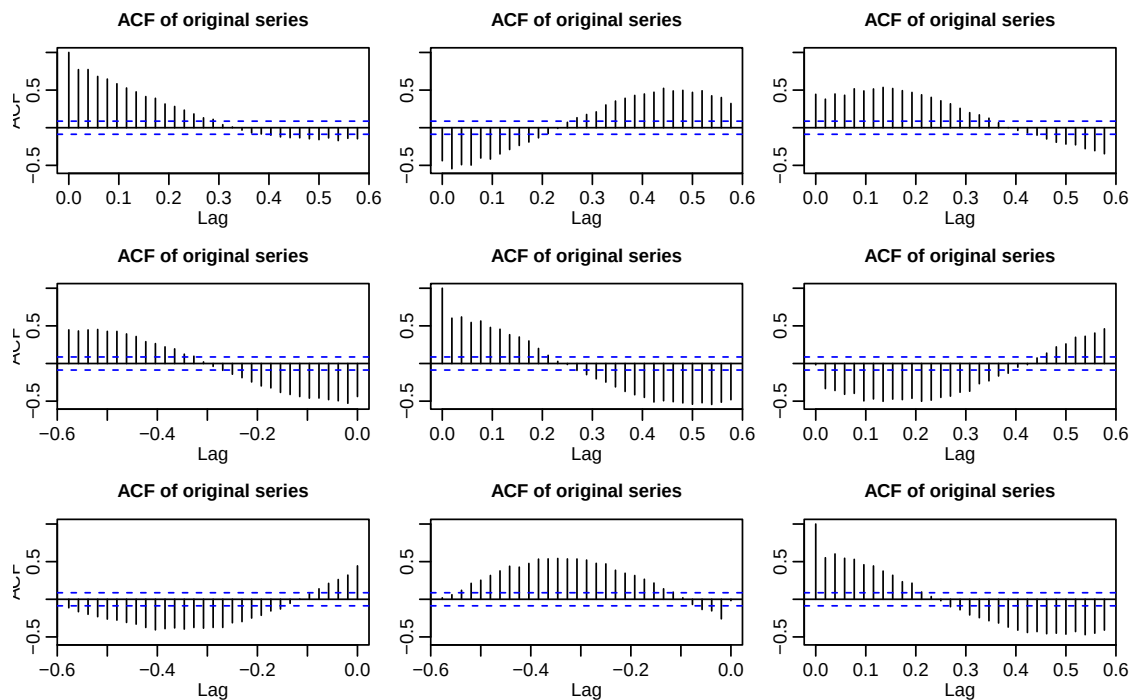
Solution: The ACF of the series indicates non-stationarity.

```
par(mar=c(4,4,0,2))
acf(x, lag.max=30, main='ACF of original series')
```



The following code performs automatic selection of the model order

```
par(mar=c(4,4,0,2))
acf(x,lag.max=30,main='ACF of original series')
```



```

fit1<-VAR(x,lag.max=12,type='const',ic='AIC')
summary(fit1)

##
## VAR Estimation Results:
## =====
## Endogenous variables: cmort, tempr, part
## Deterministic variables: const
## Sample size: 499
## Log Likelihood: -4826.462
## Roots of the characteristic polynomial:
## 0.9797 0.9797 0.8843 0.8321 0.8321 0.8253 0.8253 0.8164 0.8064 0.8064 0.7812 0.7812 0.
## Call:
## VAR(y = x, type = "const", lag.max = 12, ic = "AIC")
##
##
## Estimation results for equation cmort:
## =====
## cmort = cmort.l1 + tempr.l1 + part.l1 + cmort.l2 + tempr.l2 + part.l2 + cmort.l3 + tem
##
##           Estimate Std. Error t value      Pr(>|t|)
## cmort.l1  0.378738   0.047351   7.998 0.00000000000000986 ***
## tempr.l1 -0.154433   0.052069  -2.966   0.00317 **
## part.l1  -0.014931   0.031250  -0.478   0.63302
## cmort.l2  0.357661   0.051103   6.999 0.000000000000892736 ***
## tempr.l2 -0.079027   0.054682  -1.445   0.14907
## part.l2   0.028226   0.031243   0.903   0.36676
## cmort.l3  0.007157   0.053784   0.133   0.89420
## tempr.l3 -0.011944   0.056726  -0.211   0.83333
## part.l3  -0.011089   0.032969  -0.336   0.73676
## cmort.l4 -0.005313   0.053945  -0.098   0.92159
## tempr.l4  0.069525   0.057205   1.215   0.22484
## part.l4   0.053687   0.033382   1.608   0.10845
## cmort.l5  0.051962   0.053639   0.969   0.33318
## tempr.l5 -0.073936   0.058469  -1.265   0.20667
## part.l5   0.031820   0.033928   0.938   0.34879
## cmort.l6  0.007050   0.053453   0.132   0.89512
## tempr.l6  0.002408   0.057441   0.042   0.96658
## part.l6   0.008711   0.033540   0.260   0.79519
## cmort.l7 -0.067092   0.053199  -1.261   0.20787
## tempr.l7  0.047379   0.057220   0.828   0.40808
## part.l7   0.048044   0.032991   1.456   0.14598
## cmort.l8 -0.070100   0.050022  -1.401   0.16176
## tempr.l8  0.099119   0.055069   1.800   0.07251 .
## part.l8  -0.030699   0.031549  -0.973   0.33103
## cmort.l9  0.136537   0.046722   2.922   0.00364 **
## tempr.l9  0.029748   0.053623   0.555   0.57933
## part.l9  -0.051063   0.030906  -1.652   0.09916 .
## const    20.341224   6.855235   2.967   0.00316 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 5.36 on 471 degrees of freedom

```

```

## Multiple R-Squared: 0.7299, Adjusted R-squared: 0.7144
## F-statistic: 47.14 on 27 and 471 DF, p-value: < 0.000000000000000022
##
##
## Estimation results for equation tempr:
## =====
## tempr = cmort.l1 + tempr.l1 + part.l1 + cmort.l2 + tempr.l2 + part.l2 + cmort.l3 + tempr.l3 + part.l3 + cmort.l4 + tempr.l4 + part.l4 + cmort.l5 + tempr.l5 + part.l5 + cmort.l6 + tempr.l6 + part.l6 + cmort.l7 + tempr.l7 + part.l7 + cmort.l8 + tempr.l8 + part.l8 + cmort.l9 + tempr.l9 + part.l9 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## cmort.l1 -0.0757936  0.0534601  -1.418  0.156922
## tempr.l1  0.2155844  0.0587860   3.667  0.000273 ***
## part.l1  -0.0609984  0.0352817  -1.729  0.084483 .
## cmort.l2  0.0006037  0.0576959   0.010  0.991656
## tempr.l2  0.2746954  0.0617366   4.449  0.0000108 ***
## part.l2  -0.0942159  0.0352741  -2.671  0.007825 **
## cmort.l3  0.1241234  0.0607227   2.044  0.041499 *
## tempr.l3  0.0094305  0.0640441   0.147  0.882998
## part.l3  -0.0168651  0.0372223  -0.453  0.650691
## cmort.l4  0.0164448  0.0609041   0.270  0.787269
## tempr.l4  0.1249866  0.0645847   1.935  0.053560 .
## part.l4   0.0126508  0.0376886   0.336  0.737273
## cmort.l5 -0.0194695  0.0605590  -0.321  0.747976
## tempr.l5  0.1148335  0.0660120   1.740  0.082585 .
## part.l5  -0.0878192  0.0383051  -2.293  0.022310 *
## cmort.l6 -0.0241960  0.0603482  -0.401  0.688647
## tempr.l6  0.0089605  0.0648511   0.138  0.890165
## part.l6   0.0010322  0.0378671   0.027  0.978266
## cmort.l7 -0.0150615  0.0600617  -0.251  0.802103
## tempr.l7 -0.0316992  0.0646013  -0.491  0.623874
## part.l7   0.0095034  0.0372470   0.255  0.798722
## cmort.l8 -0.0758701  0.0564754  -1.343  0.179783
## tempr.l8 -0.0036415  0.0621727  -0.059  0.953319
## part.l8   0.0471890  0.0356188   1.325  0.185868
## cmort.l9  0.0127084  0.0527490   0.241  0.809720
## tempr.l9  0.0305348  0.0605407   0.504  0.614238
## part.l9   0.0009306  0.0348934   0.027  0.978734
## const    33.0034575  7.7395964   4.264  0.0000242 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 6.052 on 471 degrees of freedom
## Multiple R-Squared: 0.5779, Adjusted R-squared: 0.5537
## F-statistic: 23.88 on 27 and 471 DF, p-value: < 0.000000000000000022
##
##
## Estimation results for equation part:
## =====
## part = cmort.l1 + tempr.l1 + part.l1 + cmort.l2 + tempr.l2 + part.l2 + cmort.l3 + tempr.l3 + part.l3 + cmort.l4 + tempr.l4 + part.l4 + cmort.l5 + tempr.l5 + part.l5 + cmort.l6 + tempr.l6 + part.l6 + cmort.l7 + tempr.l7 + part.l7 + cmort.l8 + tempr.l8 + part.l8 + cmort.l9 + tempr.l9 + part.l9 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## cmort.l1  0.14380    0.08898   1.616  0.106745
## tempr.l1 -0.20197    0.09784  -2.064  0.039541 *
## part.l1   0.08287    0.05872   1.411  0.158831

```

```

## cmort.l2 -0.08174    0.09603   -0.851   0.395085
## tempr.l2 -0.03519    0.10276   -0.342   0.732150
## part.l2   0.21186    0.05871    3.608   0.000341 ***
## cmort.l3  0.08529    0.10107    0.844   0.399146
## tempr.l3 -0.23915    0.10660   -2.243   0.025329 *
## part.l3   0.18070    0.06195    2.917   0.003706 **
## cmort.l4  0.02143    0.10137    0.211   0.832686
## tempr.l4 -0.30823    0.10750   -2.867   0.004325 **
## part.l4   0.25187    0.06273    4.015   0.0000691 ***
## cmort.l5 -0.06219    0.10080   -0.617   0.537530
## tempr.l5  0.17715    0.10987    1.612   0.107562
## part.l5   0.01167    0.06376    0.183   0.854794
## cmort.l6 -0.06638    0.10044   -0.661   0.509004
## tempr.l6  0.14820    0.10794    1.373   0.170408
## part.l6   0.04227    0.06303    0.671   0.502797
## cmort.l7  0.02987    0.09997    0.299   0.765260
## tempr.l7 -0.02032    0.10752   -0.189   0.850162
## part.l7   0.06547    0.06199    1.056   0.291443
## cmort.l8 -0.16736    0.09400   -1.780   0.075648 .
## tempr.l8  0.27359    0.10348    2.644   0.008469 **
## part.l8   -0.06225    0.05928   -1.050   0.294235
## cmort.l9  0.06510    0.08780    0.742   0.458736
## tempr.l9  0.28904    0.10076    2.868   0.004310 **
## part.l9   -0.19733    0.05808   -3.398   0.000737 ***
## const    16.40813    12.88187    1.274   0.203384
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 10.07 on 471 degrees of freedom
## Multiple R-Squared:  0.5828, Adjusted R-squared:  0.5589
## F-statistic: 24.37 on 27 and 471 DF,  p-value: < 0.00000000000000022
##
##
## Covariance matrix of residuals:
##      cmort  tempr  part
## cmort 28.731  6.986  13.97
## tempr  6.986 36.622  38.23
## part  13.968 38.227 101.45
##
## Correlation matrix of residuals:
##      cmort  tempr  part
## cmort 1.0000 0.2154 0.2587
## tempr 0.2154 1.0000 0.6272
## part  0.2587 0.6272 1.0000

```

The selected model under AIC is $p = 9$. Under BIC (or Schwarz criterion), the selected model order is $p = 2$.

```

fit2<-VAR(x,lag.max=12,type='const',ic='SC')
summary(fit2)

##
## VAR Estimation Results:

```

```

## =====
## Endogenous variables: cmort, tempr, part
## Deterministic variables: const
## Sample size: 506
## Log Likelihood: -5002.728
## Roots of the characteristic polynomial:
## 0.8995 0.8995 0.6184 0.4889 0.4709 0.4709
## Call:
## VAR(y = x, type = "const", lag.max = 12, ic = "SC")
##
##
## Estimation results for equation cmort:
## =====
## cmort = cmort.l1 + tempr.l1 + part.l1 + cmort.l2 + tempr.l2 + part.l2 + const
##
##           Estimate Std. Error t value      Pr(>|t|)
## cmort.l1  0.36966    0.04295   8.607 < 0.0000000000000002 ***
## tempr.l1 -0.17253    0.04530  -3.809    0.000157 ***
## part.l1   0.03153    0.02465   1.279    0.201479
## cmort.l2  0.34065    0.04148   8.213 0.000000000000000186 ***
## tempr.l2 -0.03657    0.04530  -0.807    0.419874
## part.l2   0.05154    0.02585   1.994    0.046690 *
## const    37.24529    4.98318   7.474 0.000000000000035106 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 5.45 on 499 degrees of freedom
## Multiple R-Squared: 0.7056, Adjusted R-squared: 0.7021
## F-statistic: 199.3 on 6 and 499 DF, p-value: < 0.00000000000000022
##
##
## Estimation results for equation tempr:
## =====
## tempr = cmort.l1 + tempr.l1 + part.l1 + cmort.l2 + tempr.l2 + part.l2 + const
##
##           Estimate Std. Error t value      Pr(>|t|)
## cmort.l1 -0.07763    0.04850  -1.601    0.110078
## tempr.l1  0.27258    0.05115   5.329 0.00000014953457 ***
## part.l1  -0.05527    0.02784  -1.985    0.047641 *
## cmort.l2 -0.01312    0.04683  -0.280    0.779477
## tempr.l2  0.37400    0.05115   7.312 0.000000000000105 ***
## part.l2  -0.10224    0.02918  -3.503    0.000501 ***
## const    41.76327    5.62671   7.422 0.0000000000000050 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 6.154 on 499 degrees of freedom
## Multiple R-Squared: 0.5406, Adjusted R-squared: 0.5351
## F-statistic: 97.87 on 6 and 499 DF, p-value: < 0.00000000000000022
##
##
## Estimation results for equation part:

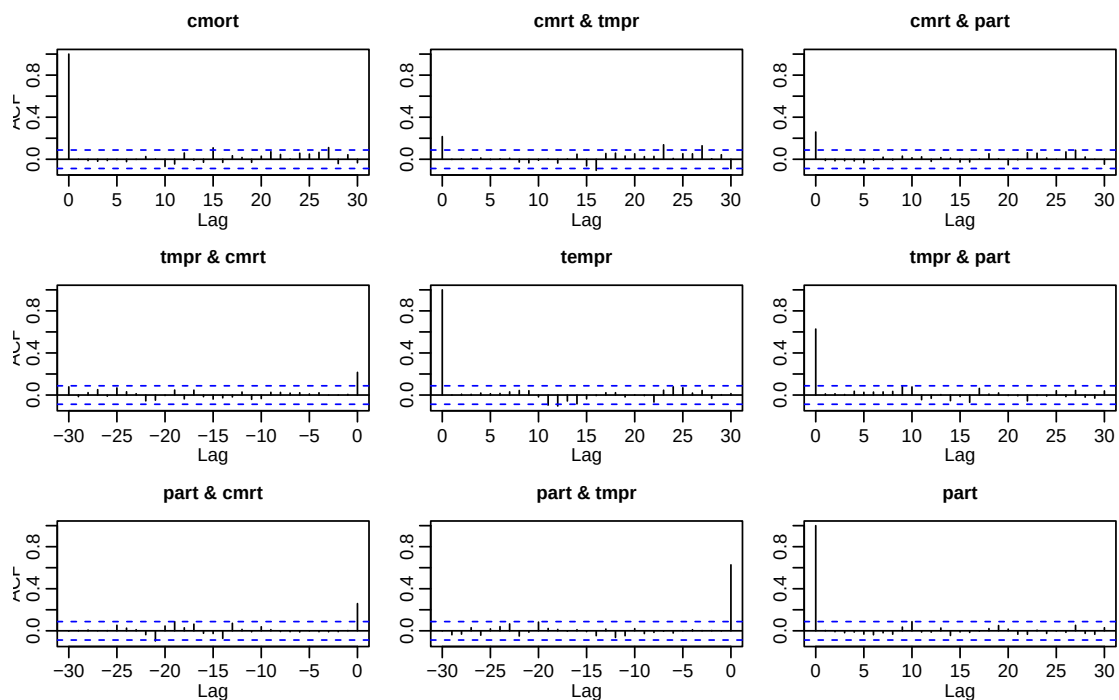
```

```
## =====
## part = cmort.l1 + tempr.l1 + part.l1 + cmort.l2 + tempr.l2 + part.l2 + const
##
##           Estimate Std. Error t value      Pr(>|t|)
## cmort.l1  0.12879    0.08776   1.467      0.142880
## tempr.l1 -0.37028    0.09255  -4.001 0.000072716673675 ***
## part.l1   0.38127    0.05038   7.568 0.0000000000000184 ***
## cmort.l2 -0.28086    0.08475  -3.314      0.000987 ***
## tempr.l2  0.08215    0.09256   0.887      0.375242
## part.l2   0.37083    0.05281   7.022 0.00000000000007197 ***
## const    46.64036   10.18202   4.581 0.000005861936587 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 11.14 on 499 degrees of freedom
## Multiple R-Squared:  0.4644, Adjusted R-squared:  0.458
## F-statistic: 72.11 on 6 and 499 DF,  p-value: < 0.00000000000000022
##
##
## Covariance matrix of residuals:
##           cmort  tempr  part
## cmort 29.704   7.805  17.48
## tempr  7.805  37.871  41.31
## part  17.478  41.308 124.01
##
## Correlation matrix of residuals:
##           cmort  tempr  part
## cmort 1.0000  0.2327  0.2880
## tempr 0.2327  1.0000  0.6028
## part  0.2880  0.6028  1.0000
```

- (b) Interpret the model selected using AIC, and check the fit of this model by inspecting the residual series.

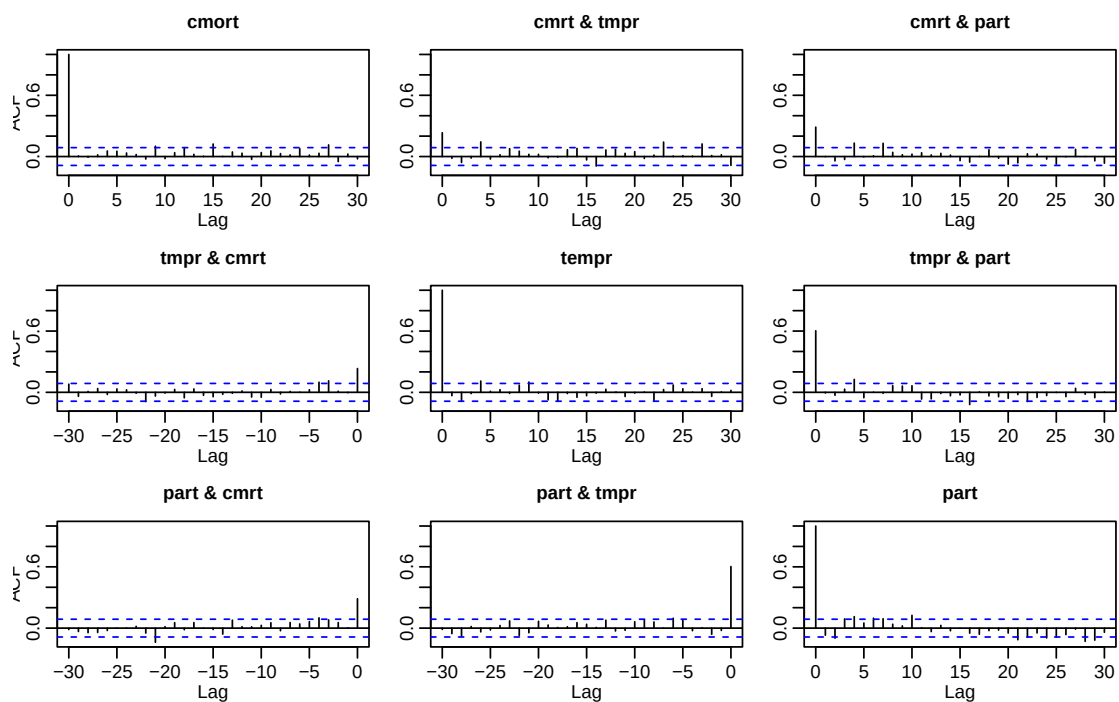
The $p = 9$ selected by AIC suggests a complex pattern of cross-correlation.

```
acf(residuals(fit1),lag.max=30)
```



The $p = 2$ selected by BIC is simpler, but apparently still gives a reasonable fit, and residuals that are largely uncorrelated.

```
acf(residuals(fit2), lag.max=30)
```

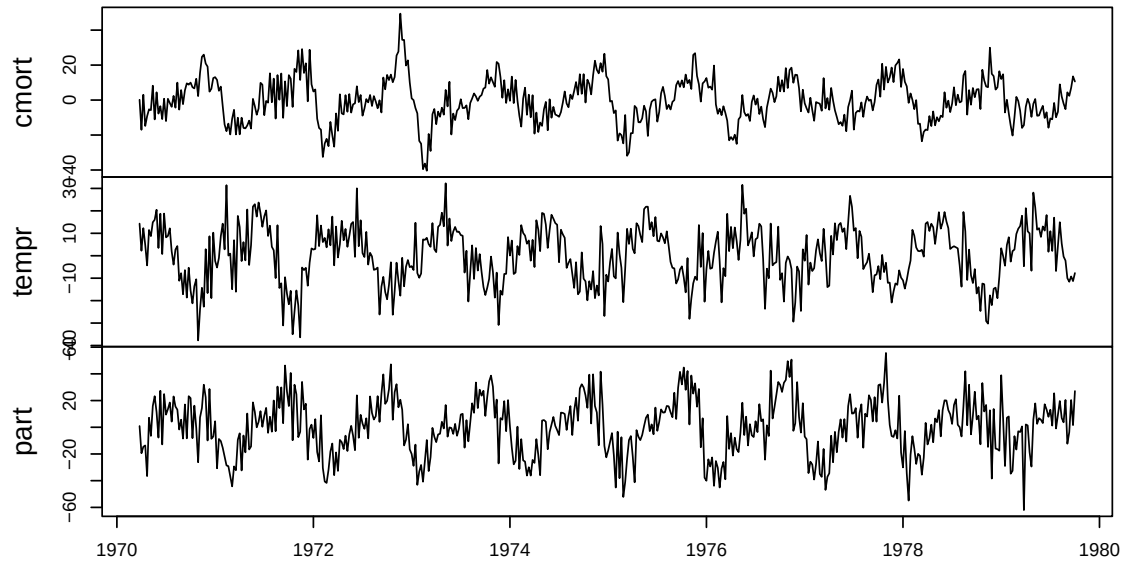


(c) Assess whether seasonal differencing of the series improves the quality of fit.

```

xd<-diff(x,12)
par(mar=c(4,4,0,2))
plot.ts(xd , main = "", xlab = "")

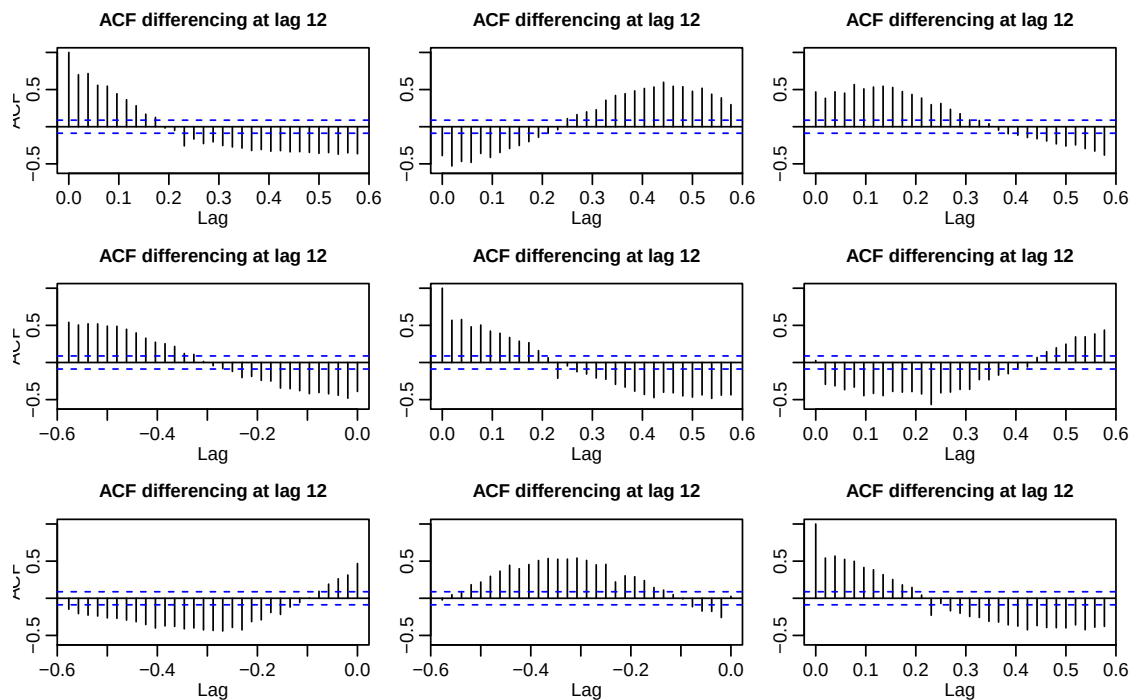
```



```

acf(xd,lag.max=30,main='ACF differencing at lag 12')

```



```

fit3<-VAR(xd,lag.max=12,type='const',ic='AIC')
fit3$p

## AIC(n)
##      12

fit4<-VAR(xd,lag.max=12,type='const',ic='SC')
fit4$p

## SC(n)
##      12

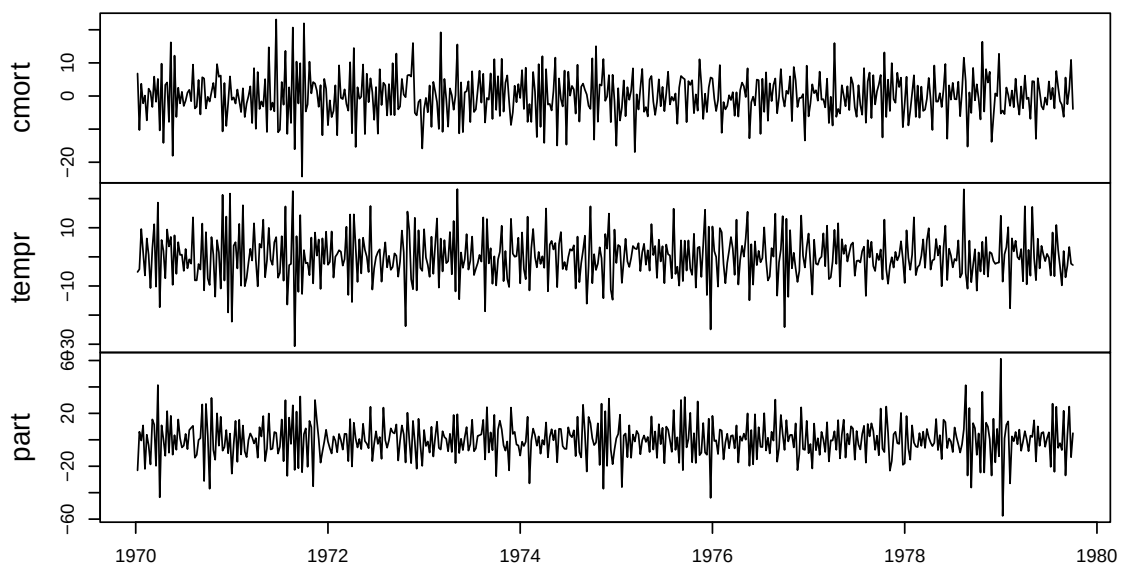
```

Thus it seems seasonal difference does not change the complexity of the inferred models, in part because the ACF indicates remaining non-stationarity. First differencing, however, does seem to have an impact

```

xd1<-diff(x,1)
par(mar=c(4,4,0,2))
plot.ts(xd1 , main = "", xlab = "")

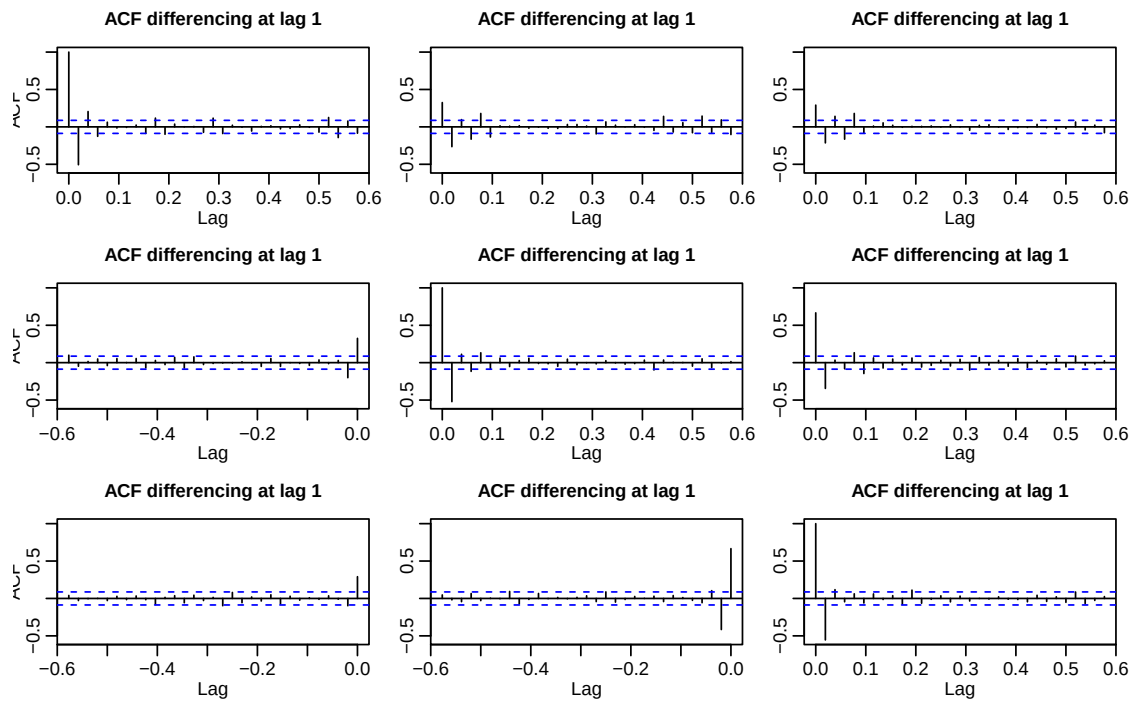
```



```

acf(xd1,lag.max=30,main='ACF differencing at lag 1')

```



```
fit5<-VAR(xd1,lag.max=12,type='const',ic='AIC')
fit5$p

## AIC(n)
##      9

fit6<-VAR(xd1,lag.max=12,type='const',ic='SC')
fit6$p

## SC(n)
##      3
```

The conclusion from this analysis is that the series should be first differenced, and this effectively removes the apparent seasonal structure, but that seasonal differencing does not really simplify the data structure.