Module 4 - Time Series Regression Models II - Dynamic Models

MATH1307 Forecasting

RMIT University, School of Science, Mathematical Sciences

Introduction

Inclusion of outside information in addition to the serial correlation contained in the variable of interest helps to improve both fit and predictive accuracy of the models in time series analysis.

In module 3, we dealt with incorporation of independent time series into the modelling process in terms of distributed lag models.

Because these models are regression models, they, in general, suffer from autocorrelated errors and multicollinearity. Also, it is not an easy task to include seasonality in these kinds of models.

To go further, we will focus on a more general class of time series regression models called dynamic linear regression models.

With this class of models, we can account for trends, cyclic patterns, seasonality, and serial correlation in the presence of predictor series and/or instrumental variables.

Specifically, in relation to dynamic models, we will cover

- intervention analysis,
- the context of spurious correlation and regression, which is about the correlation between series that is artificial and will not help model the dependent series,
- prewhitening, which helps us discover meaningful relationships.

And, we will conclude the module by a practical application to a real dataset.

In this module, we will utilize the R packages `TSA` and `dynlm` for most of the tasks.

Intervention analysis

Intervention analysis was introduced by Box and Tiao (1975), and it provides a framework for assessing the effect of an intervention on a time series of interest.

An intervention is an event that affects the process by changing the mean function or trend of a time series.

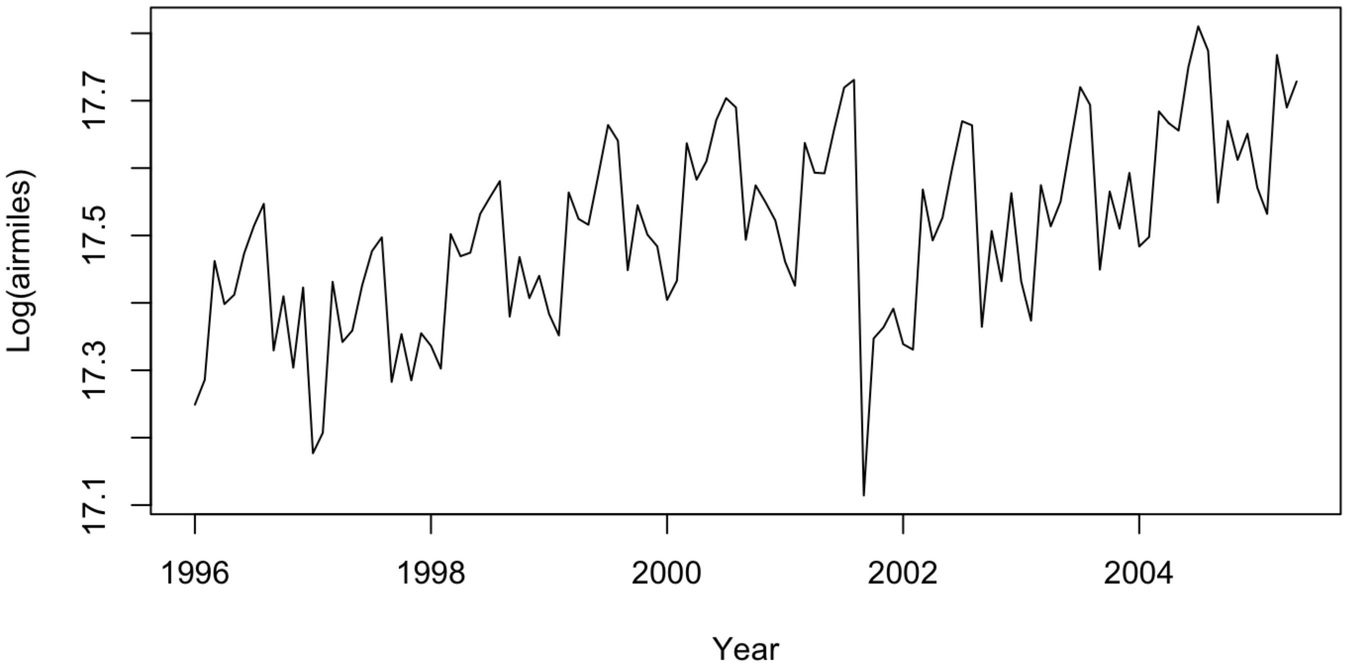
A good example of this is the effect 9/11 terrorist attacks on airline passenger-miles in the US.

This event deeply depressed air traffic around that period, but air traffic gradually regained the losses as time went on.

In this case, the trend of related time series has changed.

The following is the time series plot of the logarithms of monthly airline passenger-miles in the United States from January 1996 through May 2005.

```
data(airmiles)
plot(log(airmiles),ylab='Log(airmiles)',xlab='Year')
```



This series is subject to a trend and seasonality in addition to the mentioned intervention in 2001.

There are also other examples of interventions:

- An animal population level would crash to a very low level due to extreme weather conditions in a particular year. However, it would recover after some time.
- An increase in the speed limit for a highway would change both the number of fatalities in traffic accidents over this highway and the length of stay on the highway after the policy change.

Also notice that, in addition to the mean level, other characteristics like conditional variance and/or serial correlation characteristics of a time series would also be changed by an intervention.

For a time series $\{Y_t\}$, the general model is given by

$$Y_t = m_t + N_t, \tag{1}$$

where m_t is the change in the mean function and $\{N_t\}$ is modelled as another process like a seasonal or harmonic one or a trend model.

The process $\{N_t\}$ represents the component where there is no intervention and is referred to as the *natural* or *unperturbed* process, which may be stationary or nonstationary, seasonal or nonseasonal.

Suppose the time point T is the time the intervention occurred. Before T, m_t is assumed to be identically zero.

The series $\{Y_t, t < T\}$ is called **preintervention data** and can be used to specify the model for the unperturbed process $\{N_t\}$.

We will specify the effect of the intervention on the mean function using another function.

A useful function in this specification is the **step function**:

$$S_t^{(T)} = \begin{cases} 1, & \text{if } t \geq T, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

So, the values of $S_t^{(T)}$ will be 0 during the preintervention period and 1 in the postintervention period.

Then, we define the **pulse** function as follows:

$$P_t^{(T)} = S_t^{(T)} - S_{t-1}^{(T)}. \tag{3}$$

The pulse function is equal to 1 at the time point T and 0 elsewhere.

So, it indicates the point where the intervention occurred.

The intervention would affect the dependent series either immediately or gradually and the effect of the intervention would be either permanent or temporary.

We need to consider these while defining the form of m_t to represent the nature of the intervention.

If the intervention results in an immediate and permanent shift in the mean function, the shift can be modelled as

$$m_t = \omega S_t^{(T)} \tag{4}$$

where ω is the unknown permanent change in the mean due to the intervention.

If there is a delay of d time units before the intervention takes effect and d is known, then we can specify

$$m_t = \omega S_{t-d}^{(T)}. \tag{5}$$

If the intervention is affecting the mean function gradually, with its full force reflected only in the long run, we can use

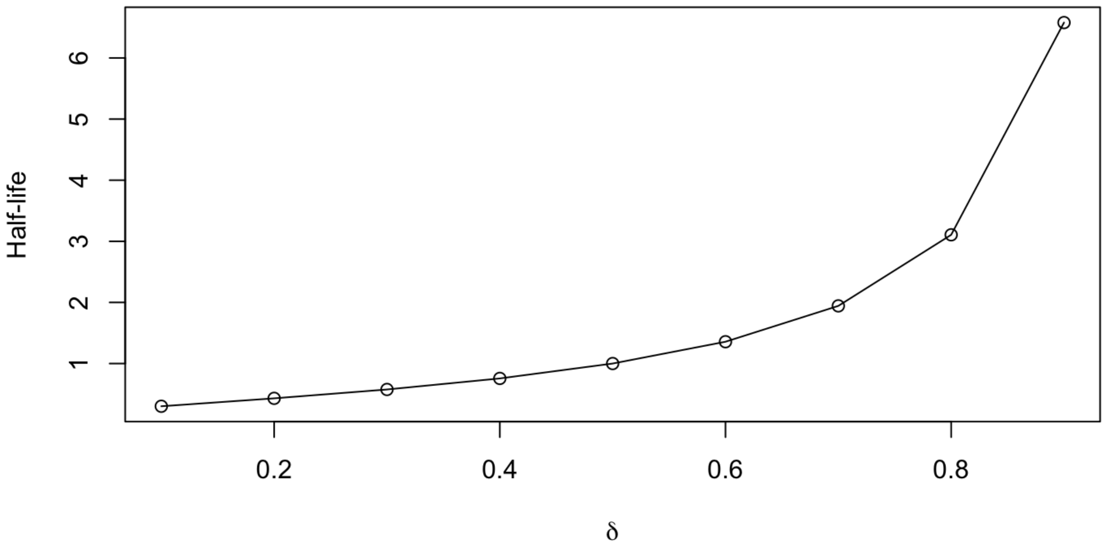
$$m_t = \delta m_{t-1} + \omega S_{t-1}^{(T)}. \tag{6}$$

In this model, the duration $\log(0.5)/\log(\delta)$ is called the *half-life* of the intervention effect, and the shorter it is, the quicker the ultimate change is felt by the system.

The following plot shows the relationship between δ and half-life.

Half-life goes infinity very quickly as δ approaches to 1.

Change of half-life according to the value of the parameterdelta



Short-lived intervention effects may be specified using the pulse dummy variable $P_t^{(T)}$.

For example, if the intervention affects the mean function only at $t = T$, then we can specify

$$m_t = \omega P_t^{(T)}. \tag{7}$$

If the intervention effects are dying gradually, we can use

$$m_t = \delta m_{t-1} + \omega P_{t-1}^{(T)}. \tag{8}$$

To write these models in general terms, we utilize the **backshift operator**, B.

The backshift operator takes the following series one step back such that

$$Bm_t = m_{t-1}. \tag{9}$$

In general,

$$B^k m_t = m_{t-k}. \tag{10}$$

Then, we can write the Eq. (8) in the following equivalent form:

$$(1 - \delta B)m_t = \omega B P_t^{(T)} \tag{11}$$

or

$$m_t = \frac{\omega B}{1 - \delta B} P_t^{(T)}. \tag{12}$$

Or we can write Eq. (3) such that $(1 - B)S_t^{(T)} = P_t^{(T)}$; and hence,

$$S_t^{(T)} = \frac{1}{1 - B} P_t^{(T)}. \tag{13}$$

Several specifications can be combined to model more sophisticated intervention effects such as the following ones:

$$m_t = \frac{\omega_1 B}{1 - \delta B} P_t^{(T)} + \frac{\omega_2 B}{1 - B} P_t^{(T)} \tag{14}$$

which would be useful for cases like Figure 2 (b).

$$m_t = \omega_0 P_t^{(T)} + \frac{\omega_1 B}{1 - \delta B} P_t^{(T)} + \frac{\omega_2 B}{1 - B} P_t^{(T)} \tag{15}$$

Which may successfully account for the case in Figure 1 (c) with both ω_1 and ω_2 are negative.

The following figures, which are given by Chan and Cryer (2011) summarize the specification of m_t for step and pulse interventions. In both figures, all are shown with a delay of 1-time unit.

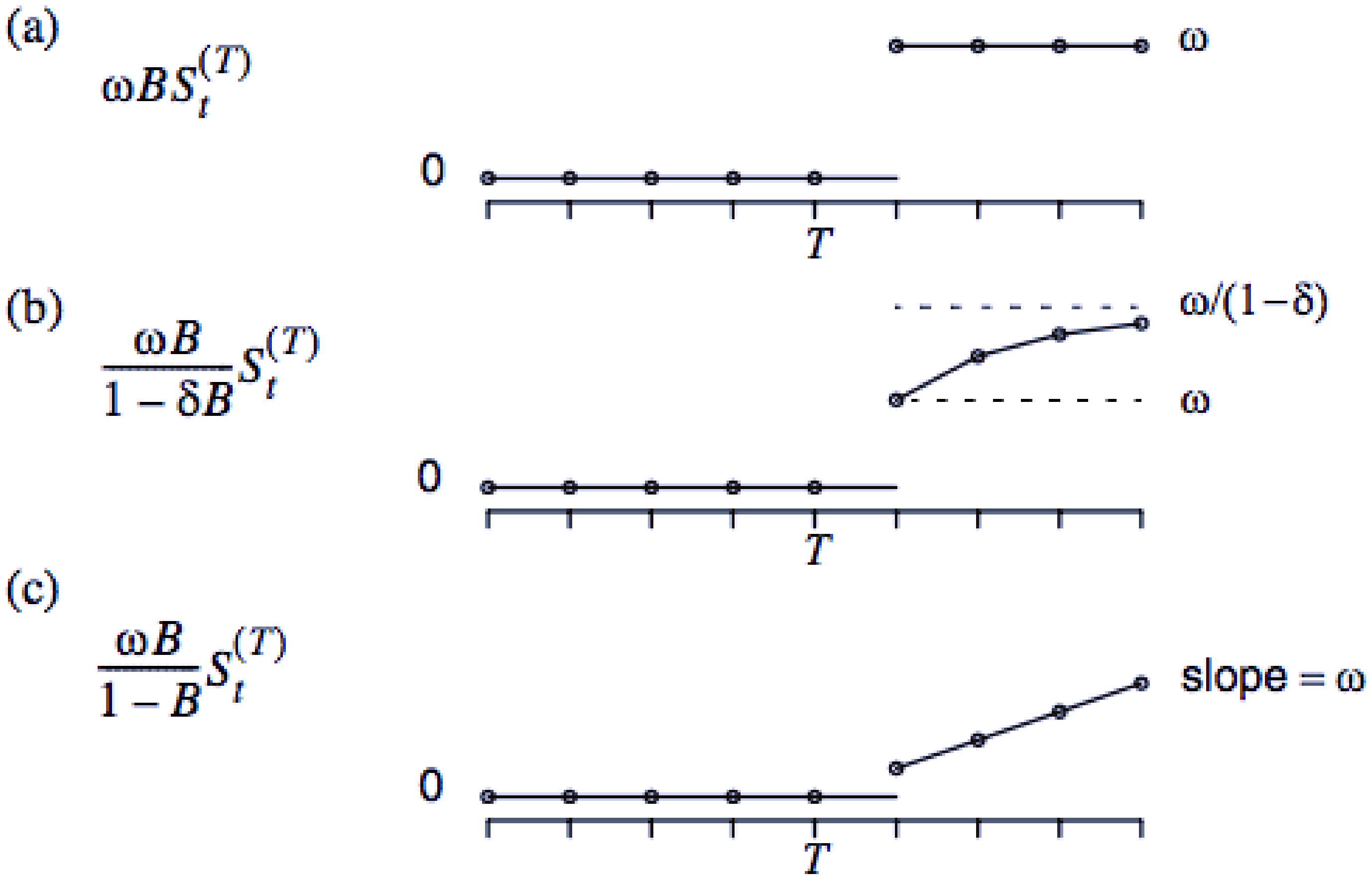
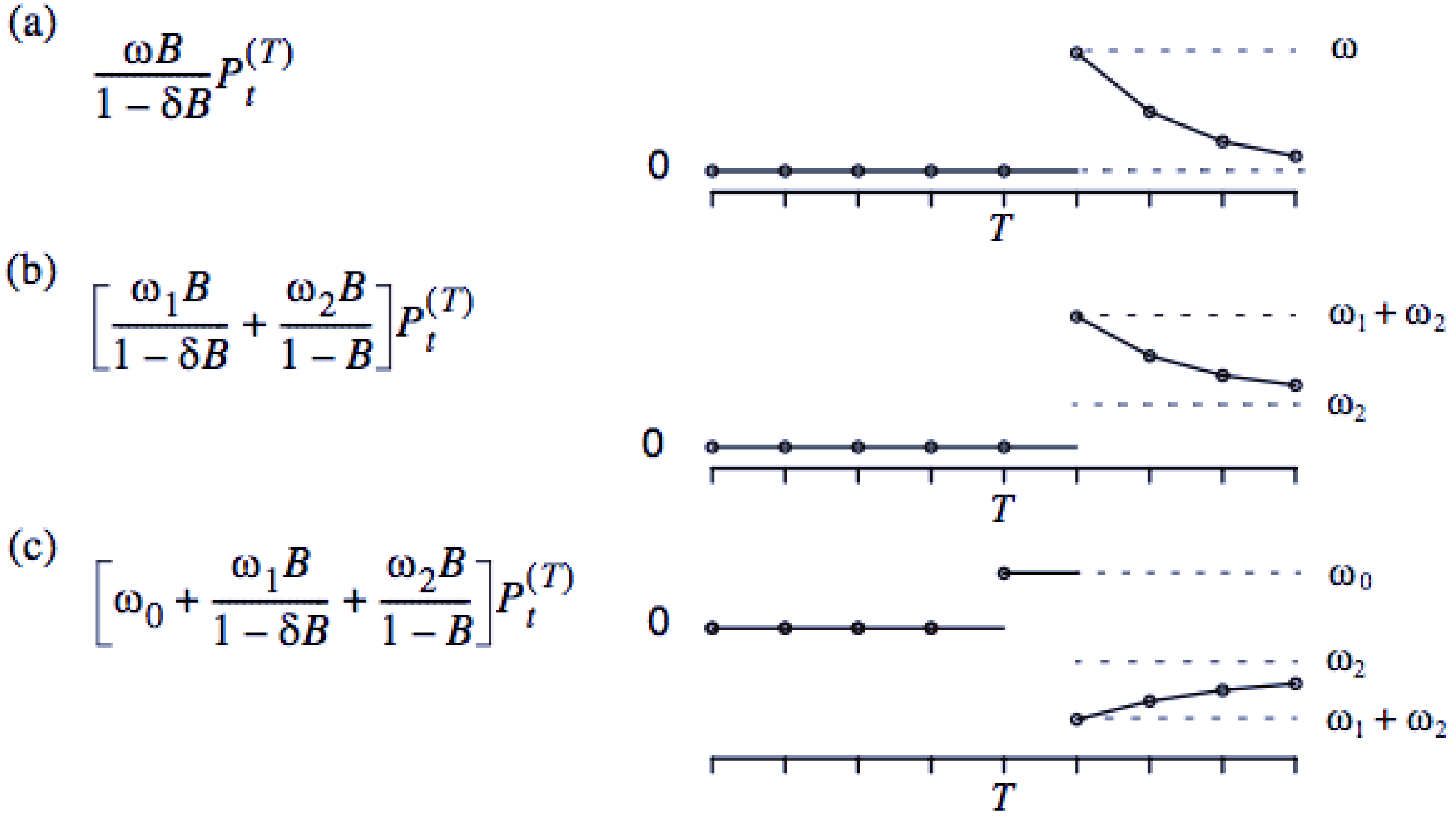


Figure 1. Some Common Models for Step Response Interventions



(a) $$\frac{\omega B}{1-\delta B}P_t^{(T)}$$

(b) $$\left[\frac{\omega_1 B}{1-\delta B}+\frac{\omega_2 B}{1-B}\right]P_t^{(T)}$$

(c) $$\left[\omega_0+\frac{\omega_1 B}{1-\delta B}+\frac{\omega_2 B}{1-B}\right]P_t^{(T)}$$

Figure 2. Some Common Models for Pulse Response Interventions

A Numerical example

Let’s focus on the monthly passenger-Airmiles data.

The terrorist acts in September 2001 had an intervention effect on air traffic.

The unexpected turn of events in September 2001 had a strong instantaneous chilling effect on air traffic.

Thus, Chan and Cryer (2011) model the intervention effect (the 9/11 effect) using the following model for m_t

$$m_t = \omega_0 P_t^{(T)} + \frac{\omega_1}{1 - \omega_2 B} P_t^{(T)}. \tag{16}$$

To apply the dynamic linear regression model, we need to rewrite the whole model in the regular model format as follows:

$$\begin{aligned}
Y_t &= m_t + M_t \\
Y_t &= \omega_0 P_t^{(T)} + \tfrac{\omega_1}{1-\omega_2 B} P_t^{(T)} + M_t \\
(1-\omega_2 B)Y_t &= (1-\omega_2 B)\omega_0 P_t^{(T)} + \omega_1 P_t^{(T)} + (1-\omega_2 B)M_t \\
Y_t - \omega_2 B Y_t &= \omega_0 P_t^{(T)} - \omega_2 \omega_0 B P_t^{(T)} + \omega_1 P_t^{(T)} + N_t \\
Y_t &= \omega_2 Y_{t-1} + (\omega_0 + \omega_1) P_t^{(T)} - \omega_2 \omega_0 P_{t-1}^{(T)} + N_t
\end{aligned} \tag{17}$$

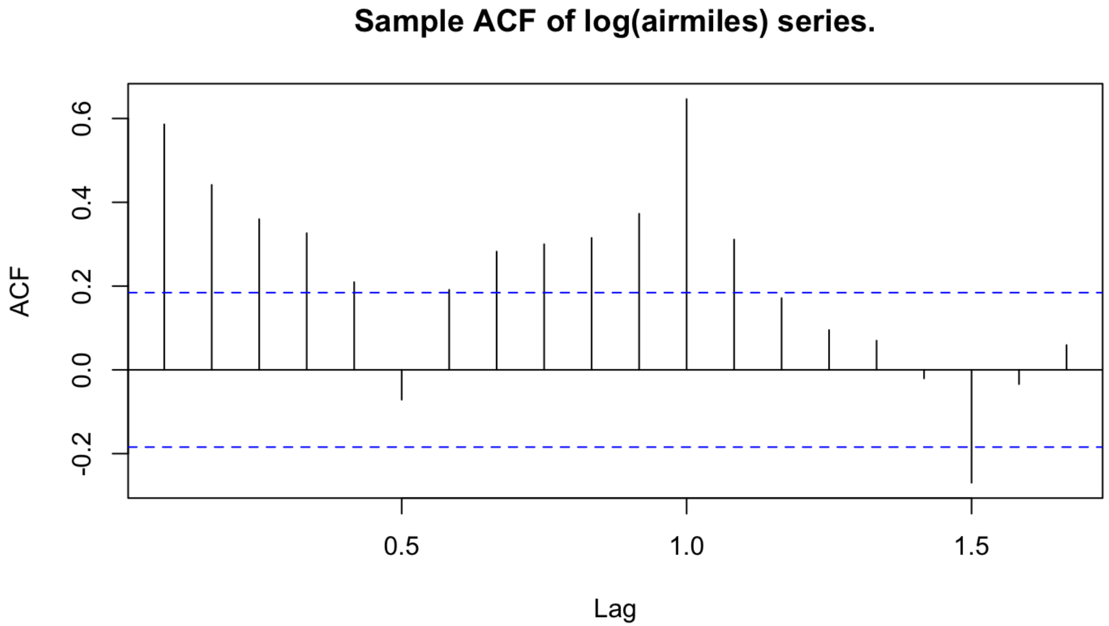
Now, the model is written in terms of past changes in the mean function and current and past pulse effects.

We will directly use this form in the dynamical model formula.

In this specification, $\omega_0 + \omega_1$ represents the instantaneous 9/11 effect,

As always we do, we will have a look at ACF plot of the log of Airmiles series first.

```
acf(log(airmiles),main="Sample ACF of log(airmiles) series.")
```



From the ACF plot, the existence of a seasonal effect is obvious.

Also, we have a decaying pattern in ACF plot which implies the existence of a trend.

This trend would be a deterministic or a stochastic trend.

We will consider both in $\{N_t\}$ component of our dynamic model.

In our implementation, we will include a trend and a seasonal component in the model as suggested by the sample ACF.

To fit the model, we will use `dynlm` function of `dynlm` package.

In the model formulation, the argument `formula` can contain `trend()` to include a trend component and `season()` to include a seasonal component in the model.

The following code chunk implements the dynamic linear regression model for the logarithm of the `airmiles` series.

```r
Y.t = log(airmiles)
T = 69 # The time point when the intervention occurred
P.t = 1*(seq(airmiles) == T)
P.t.1 = Lag(P.t,+1)
```

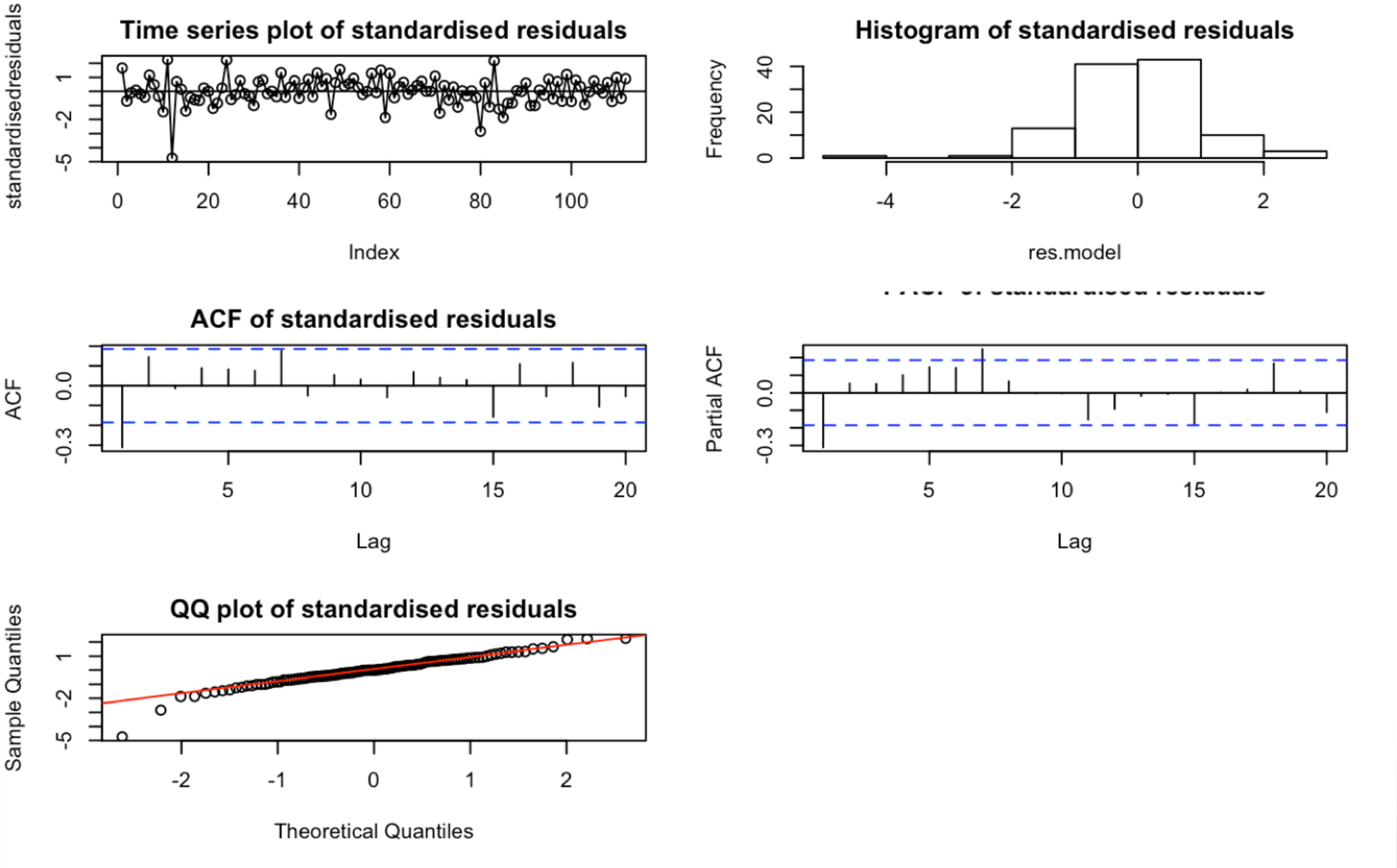
```
model1 = dynlm(Y.t ~ L(Y.t , k = 1 ) + P.t.1 + P.t + trend(Y.t) + season(Y.t))
summary(model1)
```

```
##
## Time series regression with "ts" data:
## Start = 1996(2), End = 2005(5)
##
## Call:
## dynlm(formula = Y.t ~ L(Y.t, k = 1) + P.t.1 + P.t + trend(Y.t) +
##     season(Y.t))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.140171 -0.014808  0.000093  0.019544  0.066866
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.517902   0.950555   3.701 0.000358 ***
## L(Y.t, k = 1)   0.791595   0.054812  14.442  < 2e-16 ***
## P.t.1           0.063730   0.038348   1.662 0.099800 .
## P.t            -0.380751   0.033947 -11.216  < 2e-16 ***
## trend(Y.t)      0.006818   0.001974   3.454 0.000822 ***
## season(Y.t)Feb  0.063452   0.015535   4.084 9.16e-05 ***
## season(Y.t)Mar  0.279352   0.015769  17.715  < 2e-16 ***
## season(Y.t)Apr  0.057974   0.015596   3.717 0.000339 ***
## season(Y.t)May  0.116566   0.014794   7.879 5.11e-12 ***
## season(Y.t)Jun  0.174090   0.015243  11.421  < 2e-16 ***
## season(Y.t)Jul  0.172774   0.016311  10.593  < 2e-16 ***
## season(Y.t)Aug  0.126407   0.017647   7.163 1.60e-10 ***
## season(Y.t)Sep -0.097835   0.017735  -5.517 2.92e-07 ***
## season(Y.t)Oct  0.178257   0.015914  11.201  < 2e-16 ***
## season(Y.t)Nov  0.041544   0.015008   2.768 0.006767 **
## season(Y.t)Dec  0.133498   0.015234   8.763 6.74e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03182 on 96 degrees of freedom
## Multiple R-squared:  0.9542, Adjusted R-squared:  0.947
## F-statistic: 133.2 on 15 and 96 DF,  p-value: < 2.2e-16
```

```
residual.analysis((model1) , std=TRUE , Ljung.Box=FALSE)
```

```
AIC(model1)
```

```
## [1] -437.6654
```



Our model and all coefficients but $\omega_2 \omega_0$ are significant at 5% level of significance.

Although most of the residual diagnostics are suitable, there is still some serial correlation left in the residuals.

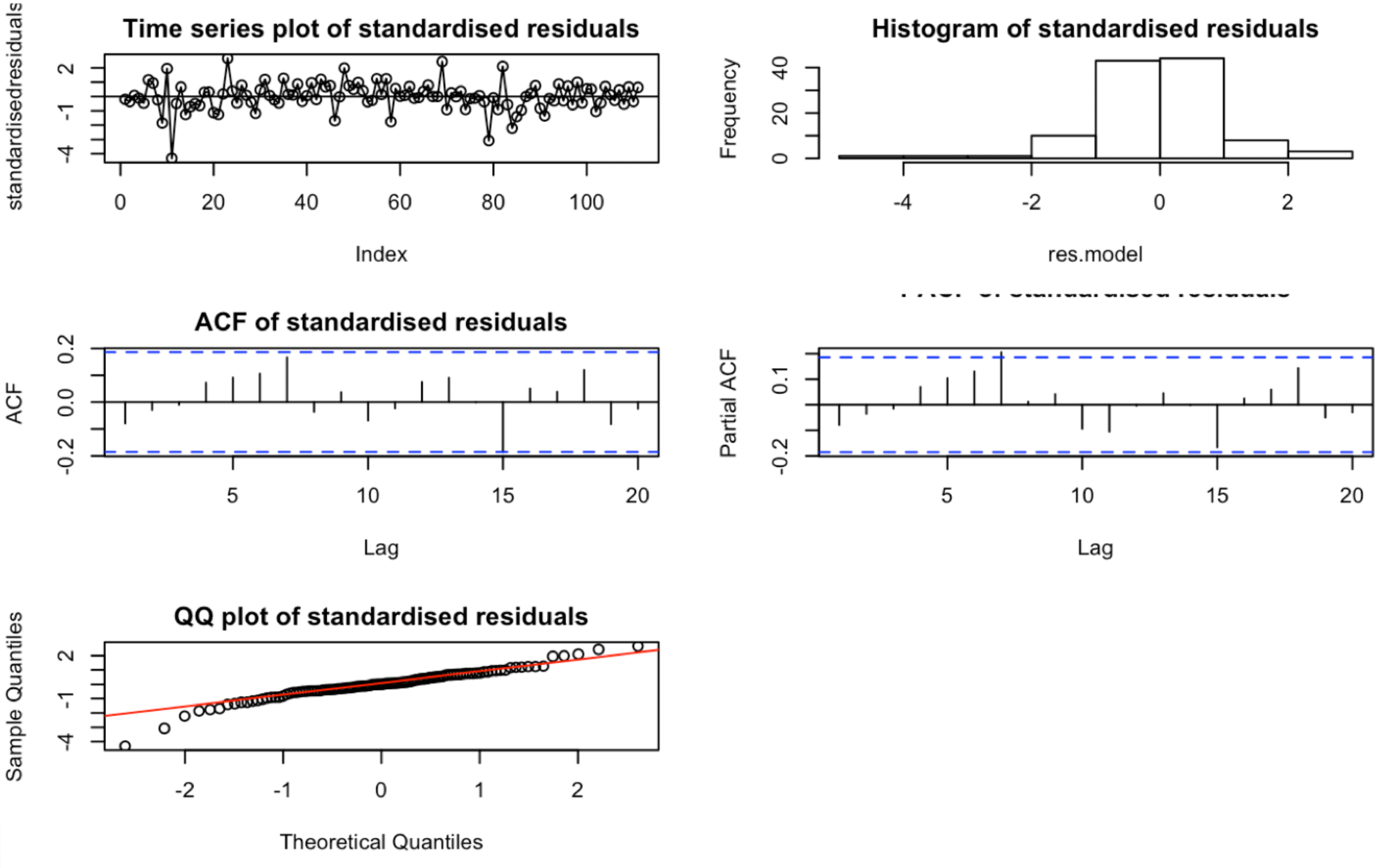
To overcome this issue, we add Y_{t-2} to the model as another predictor variable.

```
model2 = dynlm(Y.t ~ L(Y.t , k = 1 ) + P.t.1 + P.t + L(Y.t , k = 2 ) + trend(Y.t) + season(Y.t))
summary(model2)
```

```
##
## Time series regression with "ts" data:
## Start = 1996(3), End = 2005(5)
##
## Call:
## dynlm(formula = Y.t ~ L(Y.t, k = 1) + P.t.1 + P.t + L(Y.t, k = 2) +
##     trend(Y.t) + season(Y.t))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.11962 -0.01260  0.00000  0.01795  0.07362
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.288939   0.897396   3.665 0.000409 ***
## L(Y.t, k = 1)   0.522713   0.095447   5.476 3.60e-07 ***
## P.t.1          -0.043327   0.048152  -0.900 0.370525
## P.t            -0.377919   0.031975 -11.819  < 2e-16 ***
## L(Y.t, k = 2)   0.282768   0.084374   3.351 0.001159 **
## trend(Y.t)      0.006906   0.001867   3.698 0.000366 ***
## season(Y.t)Feb  0.018688   0.019003   0.983 0.327927
## season(Y.t)Mar  0.263916   0.015556  16.966  < 2e-16 ***
## season(Y.t)Apr  0.101411   0.019574   5.181 1.26e-06 ***
## season(Y.t)May  0.086024   0.016652   5.166 1.34e-06 ***
## season(Y.t)Jun  0.162926   0.014744  11.050  < 2e-16 ***
## season(Y.t)Jul  0.177135   0.015408  11.497  < 2e-16 ***
## season(Y.t)Aug  0.125593   0.016617   7.558 2.67e-11 ***
## season(Y.t)Sep -0.115165   0.017489  -6.585 2.58e-09 ***
## season(Y.t)Oct  0.101744   0.027341   3.721 0.000337 ***
## season(Y.t)Nov  0.059499   0.015106   3.939 0.000157 ***
## season(Y.t)Dec  0.104941   0.016688   6.288 1.00e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02996 on 94 degrees of freedom
## Multiple R-squared:  0.9592, Adjusted R-squared:  0.9523
## F-statistic: 138.2 on 16 and 94 DF,  p-value: < 2.2e-16
```

```
residual.analysis((model2) , std=TRUE , Ljung.Box=FALSE)
AIC(model2)
AIC(model2)
```

```
## [1] -446.1882
```

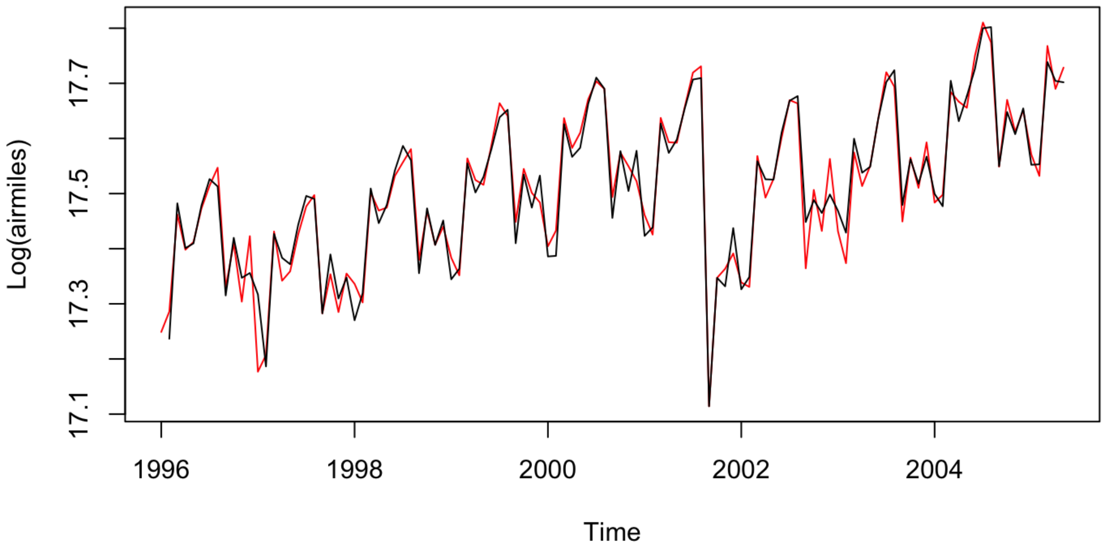


In the new model, nothing changed about the significance of the coefficients but we do not have any residue serial correlation in the error terms and AIC of the new model is decreased to -446.

Chen and Cryer (2011) find a model with a different approach to $\{N_t\}$ component finds an AIC of -424 with the same dependent variable.

Observed and fitted values are plotted below. This plot indicates a good agreement between the model and the original series.

```
plot(log(airmiles),ylab='Log(airmiles)',type="l",col="red")
lines(model1$fitted.values)
```



Forecasting

The dynamic linear model structure used for intervention analysis is suitable for calculation of forecasts.

We will use the coefficients of the fitted model to obtain forecasts. When the function `modelX = dynlm()` is used to fit the model, the output `modelX$model` returns a `data.frame` including the numerical values of the components appear in the model.

Then, we can multiply these values with corresponding coefficients and generate point forecasts.

Suppose that we fitted the following model to the series Y_t:

$$Y_t = Y_{t-1} + Y_{t-2} + S_t + trend(Y_t) + season(Y_t). \tag{18}$$

Here the component $trend(Y_t)$ is calculated by using the following formula:

$$trend(Y_t) = trend(Y_{t-1}) + 1/s, \tag{19}$$

where s is the period. Notice that if there is no seasonality, the value of s will be equal to 1.

The component $season(Y_t)$ creates a vector of $s - 1$ elements starting from the second period.

For example, if we have monthly data because the period is $s = 12$ then $season(Y_t)$ creates a vector of 11 elements corresponding to February, March, …, December.

Thus, $[0, 0, \dots, 0]$ represents January, $[1, 0, \dots, 0]$ represents February, $[0, 1, \dots, 0]$ represents March and so on.

For a one-step ahead forecast in a monthly setting, we write the model as

$$Y_{t+1} = Y_t + Y_{t-1} + S_{t+1} + trend(Y_{t+1}) + season(Y_{t+1}). \tag{20}$$

To find Y_{t+1}, which corresponds to January, we need to feed the model with such a vector

$$\begin{aligned} Y_t \quad &, Y_{t-1}, S_{t+1}, trend(Y_t) + 1/s, \\ &0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. \end{aligned} \tag{21}$$

To find the next one Y_{t+1}, which corresponds to February, we need to construct the following:

$$\begin{aligned} Y_{t+1} \quad &, Y_t, S_{t+2}, trend(Y_{t+1}) + 1/s, \\ &1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. \end{aligned} \tag{22}$$

We can keep going in the same manner to find further forecasts.

Now, let’s find 2 years ahead point forecasts for monthly passenger-Airmiles series using the `model1`.

```r
q = 24
n = nrow(model1$model)
airmiles.frc = array(NA , (n + q))
airmiles.frc[1:n] = Y.t[4:length(Y.t)]
```

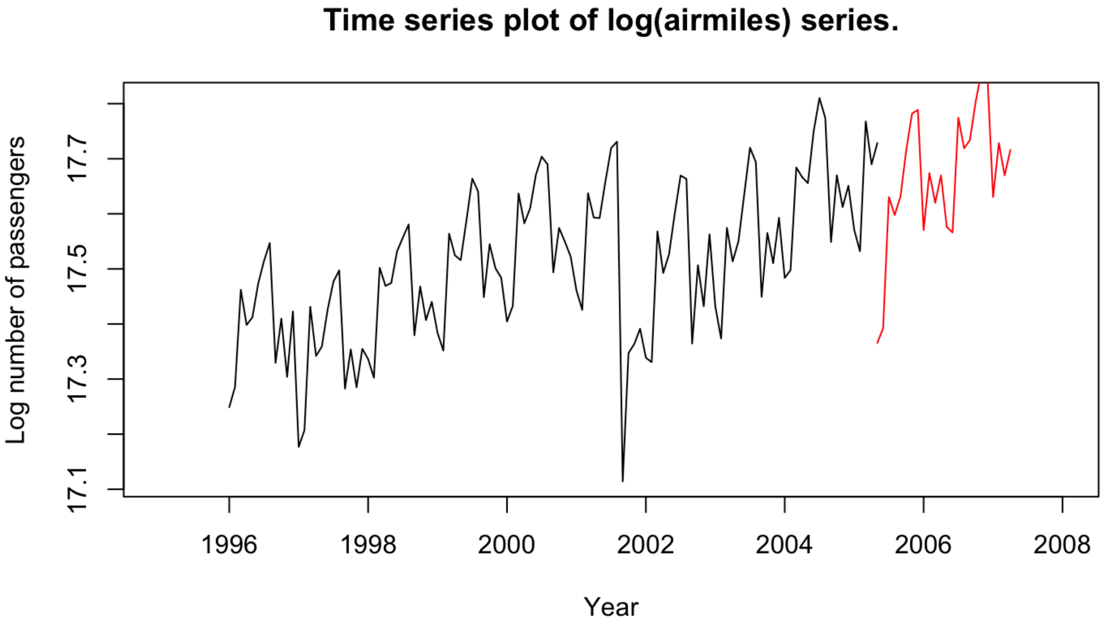
```
## Warning in airmiles.frc[1:n] = Y.t[4:length(Y.t)]: number of items to
## replace is not a multiple of replacement length
```

```r
trend = array(NA,q)
trend.start = model1$model[n,"trend(Y.t)"]
trend = seq(trend.start , trend.start + q/12, 1/12)
```

```r
for (i in 1:q){
  months = array(0,11)
  months[(i-1)%%12] = 1
  data.new = c(1,airmiles.frc[n-1+i], P.t.1[n] , P.t[n] ,trend[i],months)
  airmiles.frc[n+i] = as.vector(model1$coefficients) %*% data.new
}
```

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

plot(Y.t,xlim=c(1995,2008),ylab='Log number of passengers',xlab='Year',main = "Time series plot of log(airmiles)
lines(ts(airmiles.frc[(n+1):(n+q)],start=c(2005,5),frequency = 12),col="red")
```



Spurious Correlation

A main purpose of building a time series model is for forecasting.

Often, the time series under study may be related to or led by, some other covariate time series.

In such cases, a better understanding of the underlying process and/or more accurate forecasts may be achieved by incorporating relevant covariates into the time series model.

Let $Y = \{Y_t\}$ be the time series of the response variable and $X = \{X_t\}$ be a covariate time series that we hope will help explain or forecast Y.

To explore the correlation structure between independent variable X and dependent variable Y and their lead-led relationship, we define the *cross-covariance* function (CCF) $\gamma_{t,s}(X, Y) = Cov(X_t, Y_s)$ for each pair of integers t and s.

X and Y are jointly (weakly) stationary if their means are constant and the covariance $\gamma_{t,s}(X, Y)$ is a function of the time difference $t - s$.

For jointly stationary processes, the cross-correlation function between X and Y at lag k can then be defined by $\rho_k(X, Y) = Corr(X_t, Y_t - k) = Corr(X_t + k, Y_t)$.

The coefficient $\rho_0(Y, X)$ measures the contemporaneous linear association between X and Y, whereas $\rho_k(X, Y)$ measures the linear association between X_t and that of Y_{t-k}.

Notice that $Corr(X_t, Y_{t-k})$ need not equal $Corr(X_t, Y_{t+k})$.

The CCF can be estimated by the sample cross-correlation function (sample CCF) defined by

$$r_k(X, Y) = \frac{\Sigma(X_t - \bar{X})(Y_t - \bar{Y})}{\sqrt{\Sigma(X_t - \bar{X})^2}\sqrt{\Sigma(Y_t - \bar{Y})^2}} \tag{23}$$

The sample CCF becomes the sample ACF when $Y = X$. Sample cross-correlations that are larger than $1.96/n$ in magnitude are then deemed significantly different from zero.

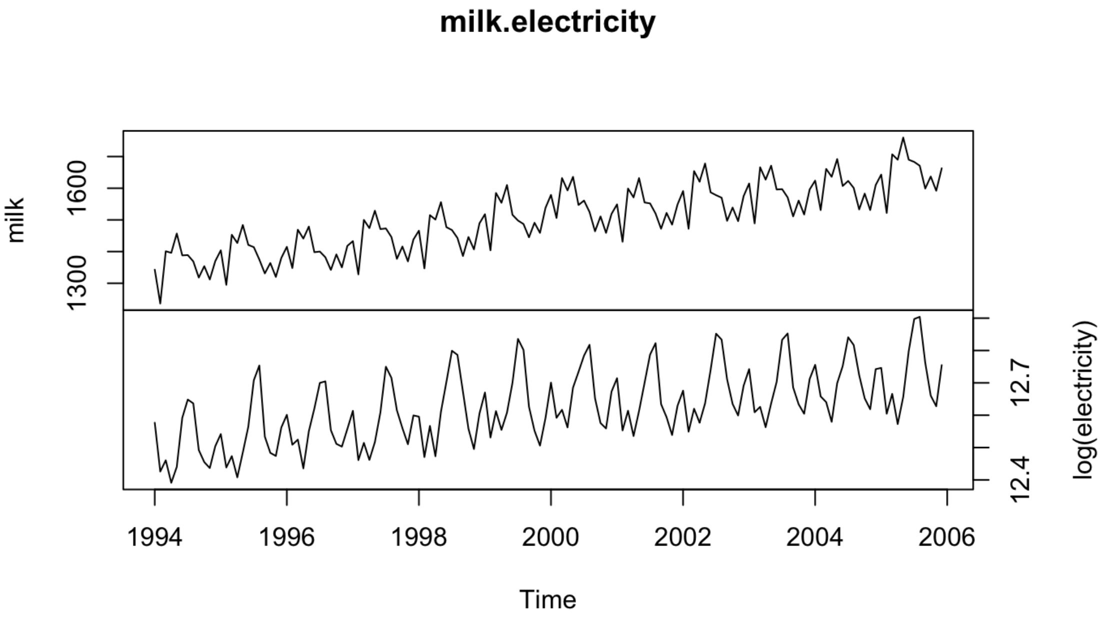
There is an important drawback of CCF that use of the $1.96/n$ rule to decide the significance of the sample CCF may lead to more false positives than the nominal 5% error rate, even though the response and covariate time series are independent of each other.

This is due to an inflated variance problem of sample cross-correlation coefficients and is more serious for nonstationary data.

These issues provide insight into why we sometimes obtain a nonsense (spurious) correlation between time series variables.

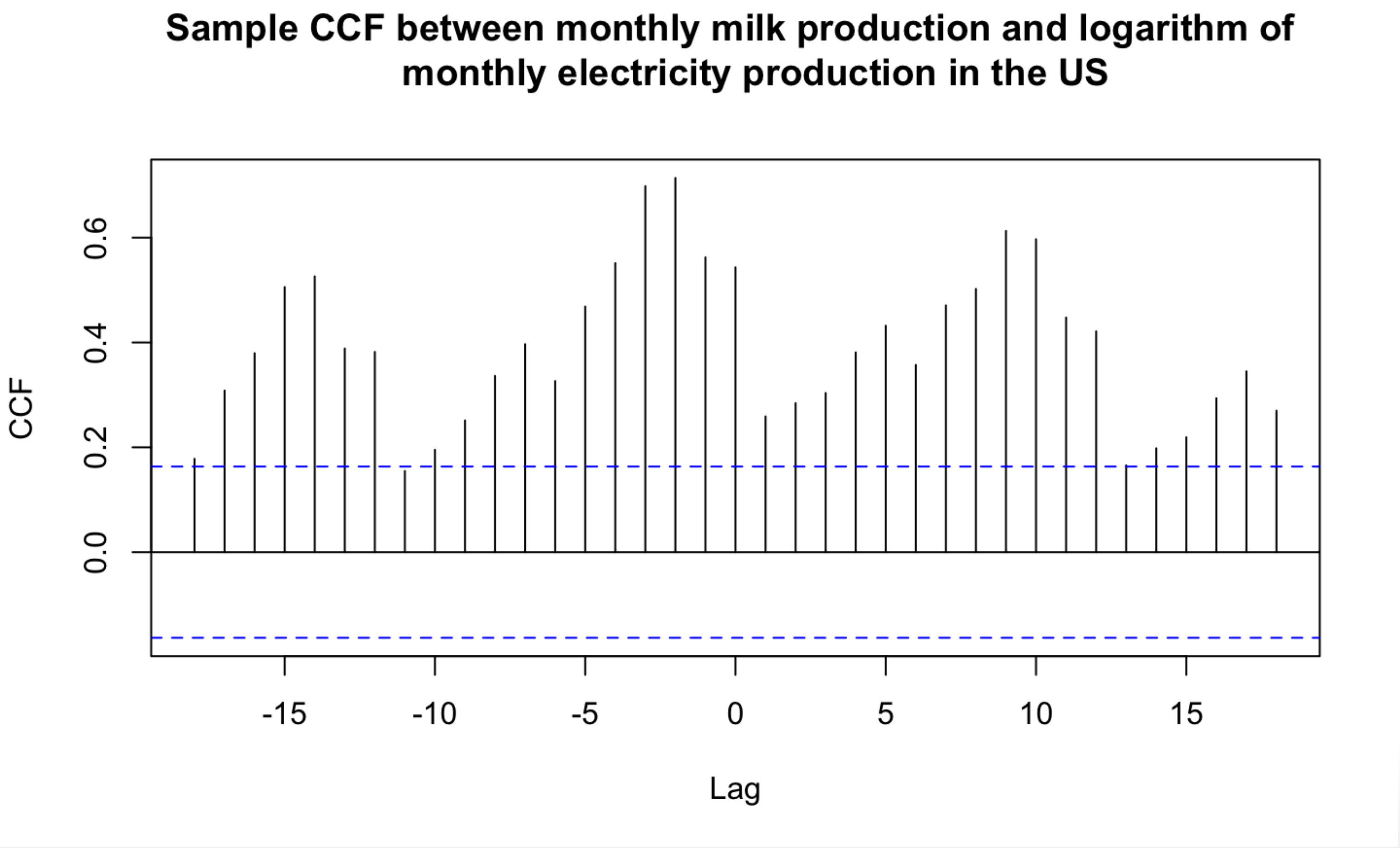
We can demonstrate this issue by using the monthly milk production and the logarithms of monthly electricity production in the US from January 1994 to December 2005. The series is displayed below.

```
data(milk); data(electricity)
milk.electricity=ts.intersect(milk,log(electricity))
plot(milk.electricity,yax.flip=T)
```



The value of sample cross-correlation between these series is 0.54 at lag zero and the sample CCF is displayed as follows:

```
ccf(as.vector(milk.electricity[,1]), as.vector(milk.electricity[,2]),ylab='CCF', main = "Sample CCF between mont
    monthly electricity production in the US")
```



Nearly all of the cross-correlations are significantly different from zero according to the $1.96/n$ rule.

Obviously, it is difficult to come up with a plausible reason for such a strong relationship between monthly electricity production and monthly milk production.

The nonstationarity in the milk production series and in the electricity series is more likely the cause of the spurious correlations found between the two series.

As seen here, while including an independent variable in the time series analysis models, there is a danger of getting significant regression results from unrelated data especially when working over nonstationary data in regression analysis.

These regressions are called **spurious** regressions. We need to be aware of this problem while working on the inclusion of independent series in time series analysis regression models.

Prewhitening

We observed that with strongly autocorrelated data it is difficult to assess the dependence between the two processes.

A useful approach to disentangle the linear association between X and Y series from their autocorrelation is *prewhitening*.

Basically, the process of transforming the X's to the $\tilde{X}$'s via the filter $\pi(B) = 1 - \pi_1 B - \pi_2 B^2 - \cdots$ is known as **whitening** or **prewhitening**.

We can study the CCF between X and Y by prewhitening the Y and X using the same filter based on the X process and then computing the CCF of prewhitened $\tilde{\tilde{Y}}$ and $\tilde{X}$.

Since *prewhitening is a linear operation*, any linear relationships between the original series will be preserved after prewhitening.

We assume, furthermore, that $\tilde{Y}$ is stationary. This approach has two advantages:

- the statistical significance of the sample CCF of the prewhitened data can be assessed using the cutoff $1.96/n$, and
- the theoretical counterpart of the CCF so estimated is proportional to certain regression coefficients.

The main aim of prewhitening is to transform one of the two series such that it is uncorrelated (white noise).

To apply prewhitening, we will follow the steps:

1. Make sure that both X and Y series are stationary. You can use ordinary and seasonal differencing to ensure this.
2. Fit the following model to X series:

   $$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + e_t \tag{24}$$

   where e_t is error process. The order p can be specified based on the AICs of a set of tentative models. And, store the residuals arising from the model fitted for X series.
3. Replace X-variables in the fitted form of the above model by Y-variables (filter Y series using the model specified for the X). Store the differences between observed and fitted Y series.
4. Display the sample CCF between the residuals from Step 2 and filters Y series of Step 3.

An approximate prewhitening can be done easily by first differencing the data.

For the implementation of prewhitening in R, we can use `prewhiten()` function from the `TSA` package.

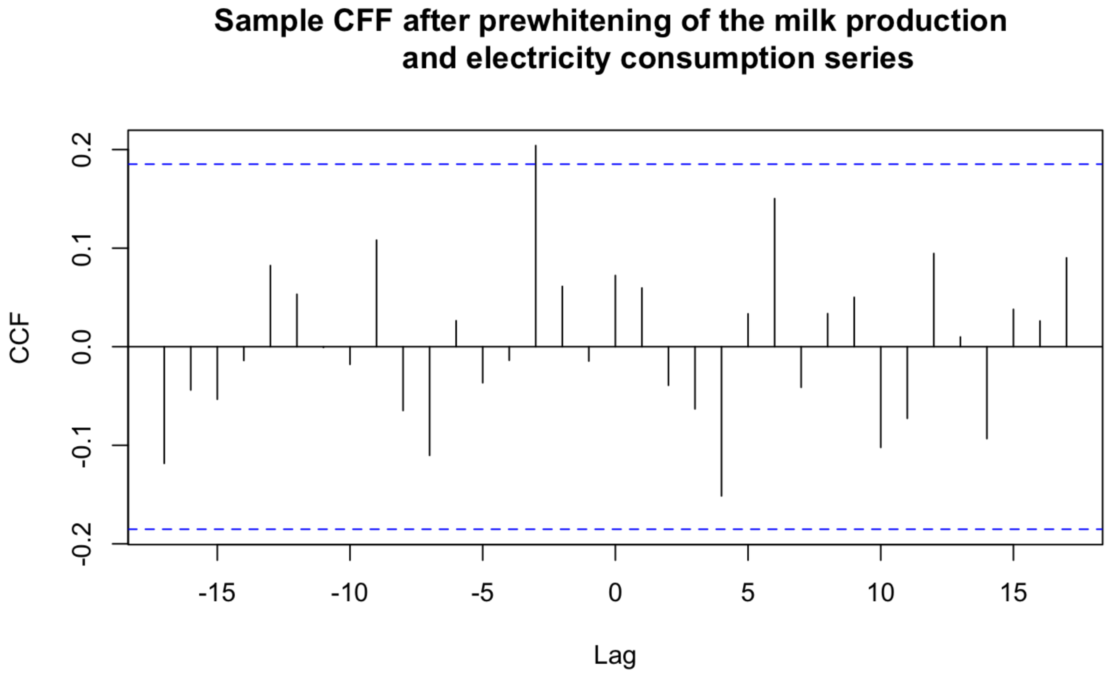
Alternatively, one can use `ar()` and `filter()` functions of the `base` package to do prewhitening.

For example, for milk production and electricity consumption data, both are highly seasonal and contain trends.

Consequently, they can be differenced with both regular differencing and seasonal differencing, and then the prewhitening can be carried out by filtering both differenced series by an AR model fitted to the differenced milk data.

The following code chunk applies prewhitening to milk production and electricity consumption data.

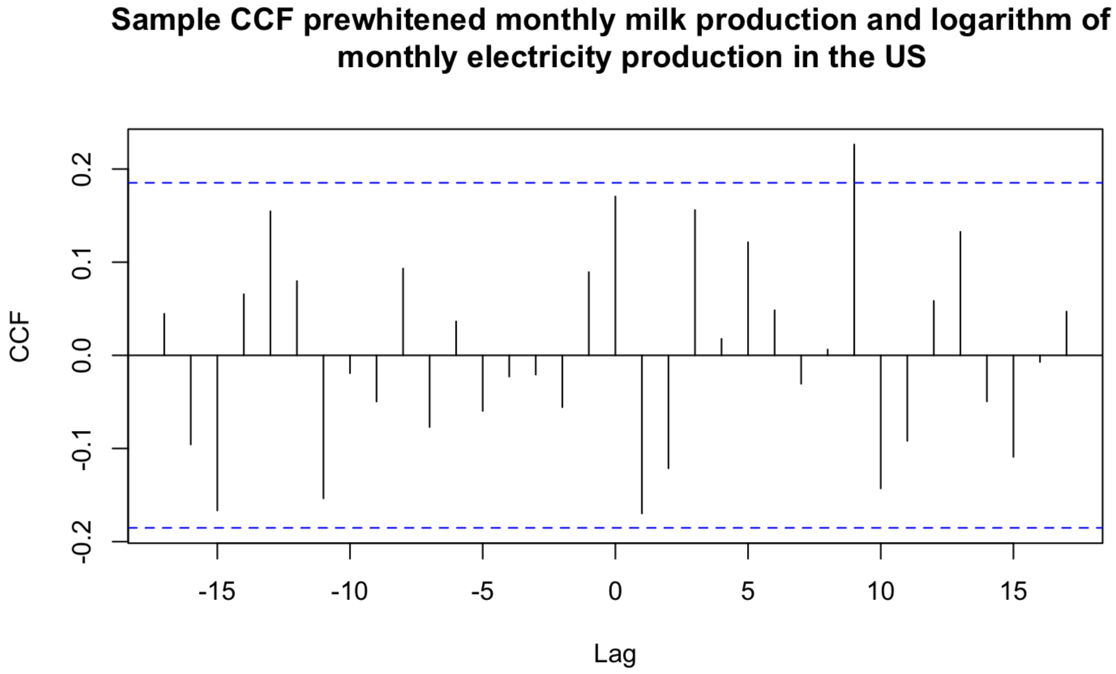
```
me.dif=ts.intersect(diff(diff(milk,12)),diff(diff(log(electricity),12)))
prewhiten(as.vector(me.dif[,1]),as.vector(me.dif[,2]),ylab='CCF', main="Sample CFF after prewhitening of the mil
          and electricity consumption series")
```



The following chunk uses the step-by-step way of prewhitening and displays CCF of prewhitened series.

```r
x.series = diff(diff(milk,12))
y.series = diff(diff(log(electricity),12))
p = 19
model.x <- ar(x.series,order=p)
coeffs = model.x$ar
filter = c(1, -coeffs)
y.filtered <- filter(y.series, method="convolution", filter = filter, sides = 1)
```

```r
ccf(as.vector(model.x$resid)[(p+1):131], as.vector(y.filtered)[(p+1):131],ylab='CCF', main = "Sample CCF prewhit
    monthly electricity production in the US")
```



The slight difference between the CCFs above is due to the use of different p orders in the prewhitening process.

The significant correlations in the above plots can be related to the false alarm rate of the CCF.

Thus, it seems that milk production and electricity consumption are in fact largely uncorrelated, and the strong cross-correlation pattern found between the raw data series is indeed spurious.

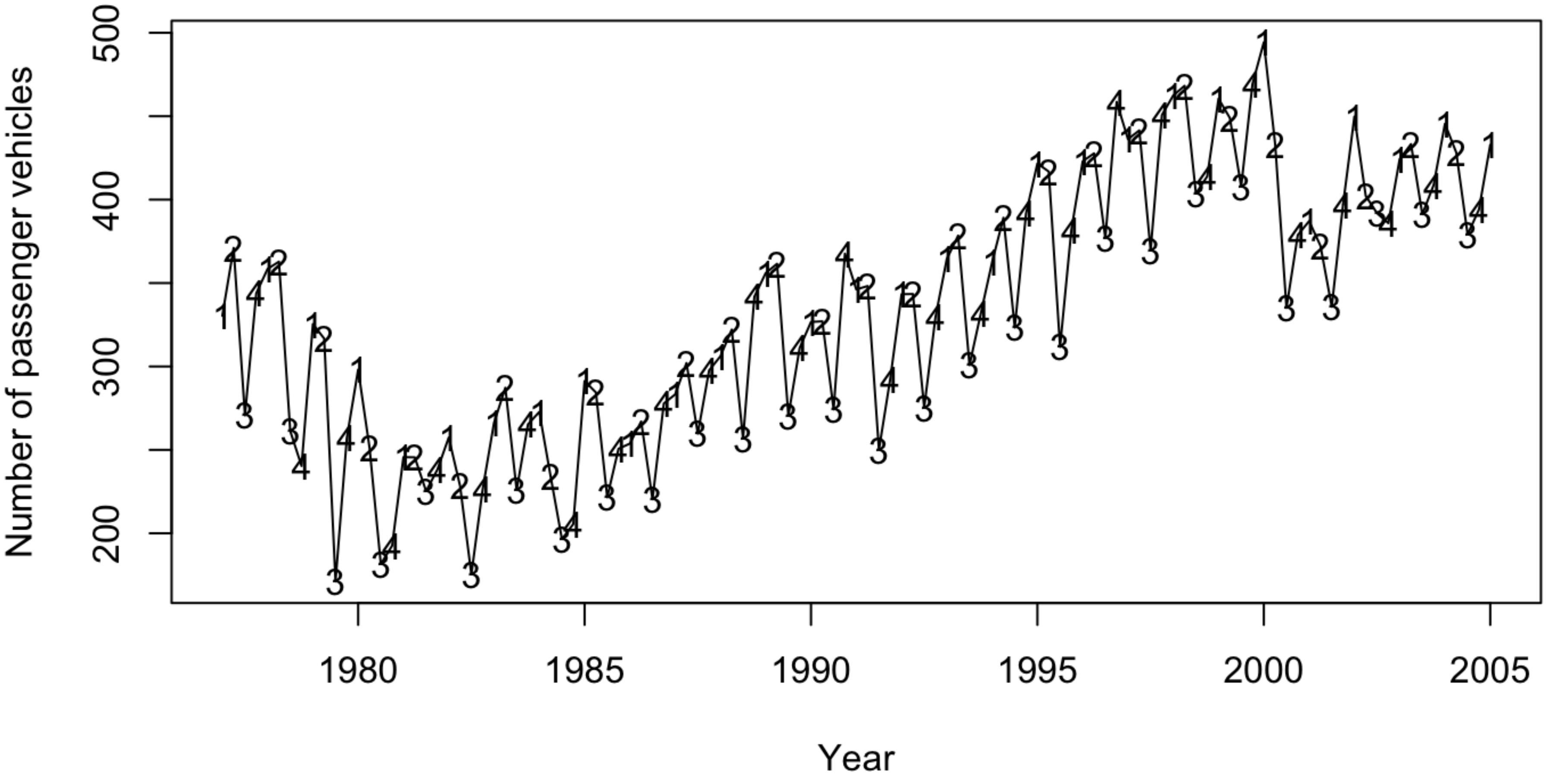
Practical applications

Intervention analysis

Recall that we have decomposed the ukcars series including the number of passenger vehicles produced in the UK from the first quarter of 1977 through the first quarter of 2005.

Let’s load the series into R and display time series plot for this series with the following code chunk:

```r
data("ukcars") # This data is available in the expsmooth package
plot(ukcars,ylab='Number of passenger vehicles',xlab='Year')
points(y=ukcars,x=time(ukcars), pch=as.vector(season(ukcars)))
```


Number of passenger vehicles
500
400
300
200
1980
1985
1990
1995
2000
2005
Year

There are multiple trends seen in the series.

Seasonality is obvious from the plot with labels as higher productions come from the second and fourth quarters and lower records come from mostly the third quarter.

There is no obvious evidence for seasonality.

Because the observations are bouncing around a mean level, we think that there is a moving average behaviour in the series.

There is an intervention point around the year 2000. After the intervention productions suddenly drop down a mean level for a short time period and then the mean level suddenly increase and series starts bouncing around the new mean level.

Let’s revisit Figure 1 and 2 to identify a suitable step or pulse function for this kind of intervention.

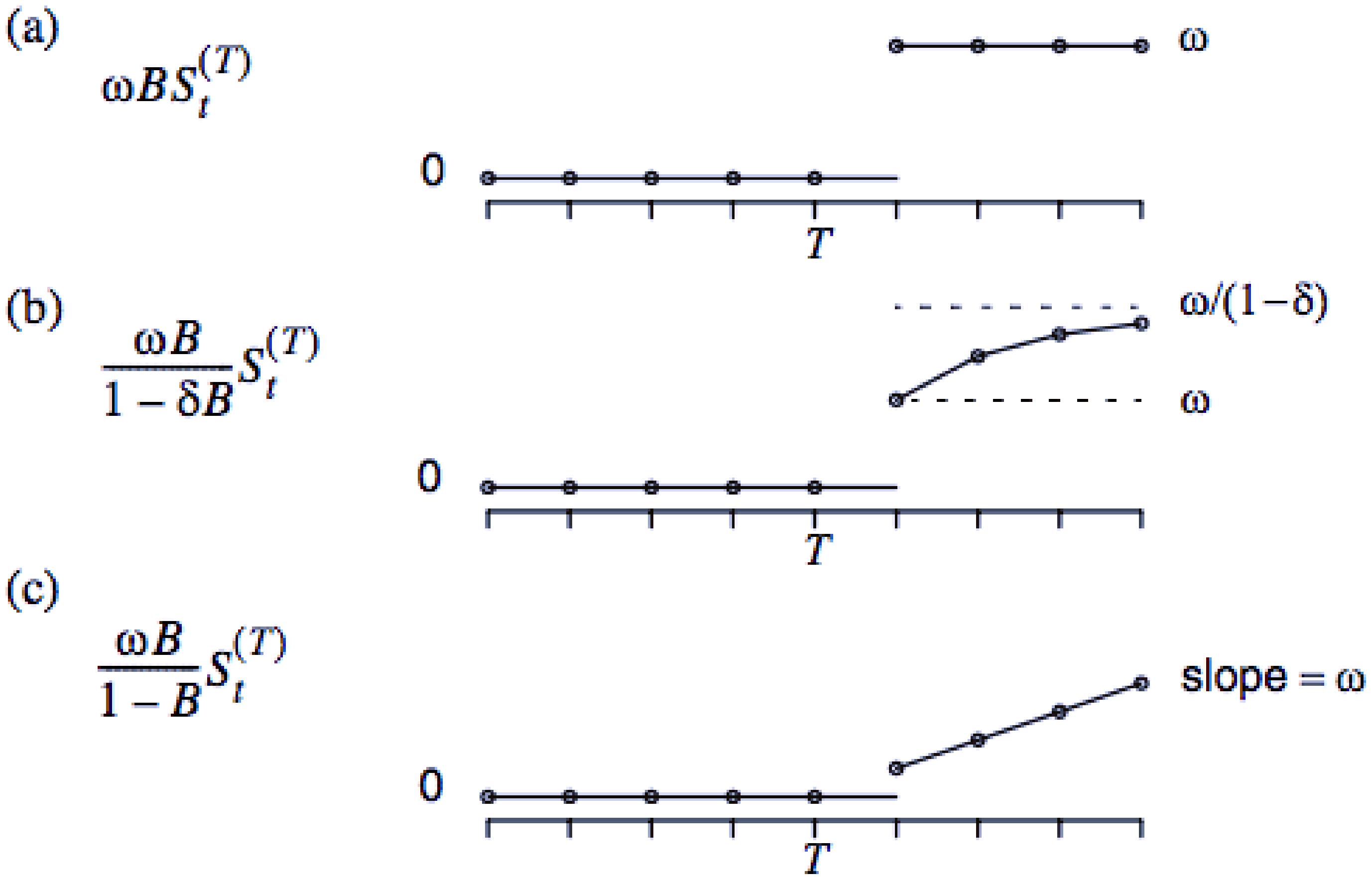


Figure 1. Some Common Models for Step Response Interventions

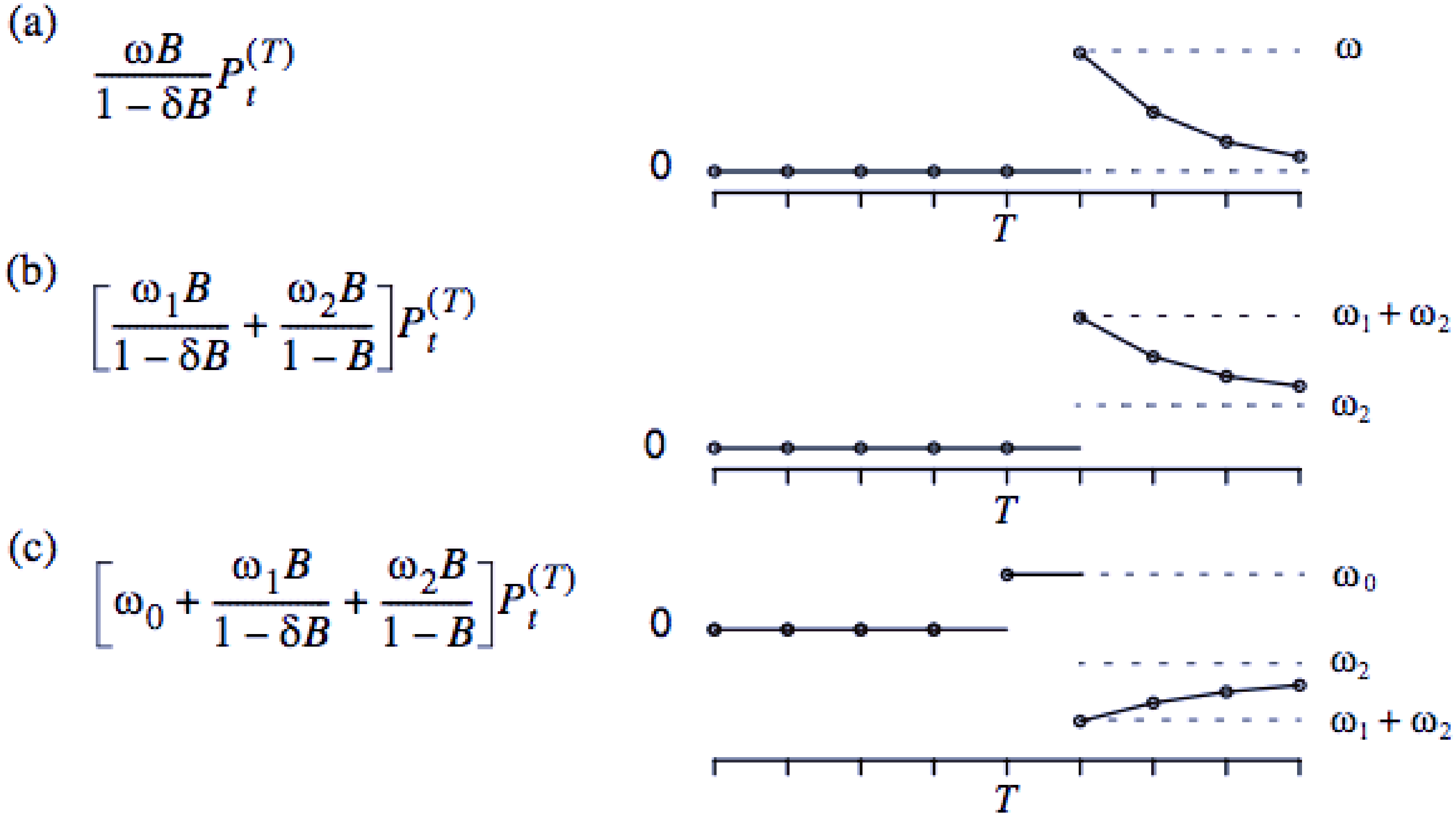

(a)
$\frac{\omega B}{1-\delta B}P_t^{(T)}$
ω
0
T
(b)
$\left[\frac{\omega_1 B}{1-\delta B}+\frac{\omega_2 B}{1-B}\right]P_t^{(T)}$
$\omega_1+\omega_2$
ω_2
0
T
(c)
$\left[\omega_0+\frac{\omega_1 B}{1-\delta B}+\frac{\omega_2 B}{1-B}\right]P_t^{(T)}$
ω_0
0
ω_2
$\omega_1+\omega_2$
T

Figure 2. Some Common Models for Pulse Response Interventions

For the ukcars series, we will look for a sudden drop and then another change in the mean level to recover from the effect of the intervention.

This behaviour is in accordance with the panel (c) of Figure 2.

So, we will use

$$m_t = \left[\omega_0 + \frac{\omega_1 B}{1-\delta B} + \frac{\omega_2 B}{1-B}\right] P_t^T \tag{25}$$

and fit the model

$$Y_t = m_t + M_t \tag{26}$$

where N_t is the component that includes the trend and seasonal effects.

To fit the model, we need to identify the lags of the series Y_t and P_t^T to be included in the model.

There are two ways of doing this. The longer way is to expand the model analytically.

$$\begin{aligned}
Y_t &= m_t + M_t \\
&= \left[\omega_0 + \frac{\omega_1 B}{1-\delta B} + \frac{\omega_2 B}{1-B}\right] P_t^T + M_t \\
Y_t(1-\delta B)(1-B) &= \left[\omega_0(1-\delta B)(1-B) + \omega_1 B(1-B) + \omega_2 B(1-\delta B)\right] P_t^T \\
&\quad +(1-\delta B)(1-B)M_t \\
Y_t - Y_{t-1} - \delta Y_{t-1} + \delta Y_{t-2} &= \omega_0 P_t^T - \omega_0 P_{t-1}^T - \omega_0 \delta P_{t-1}^T + \omega_1 P_{t-1}^T - \omega_1 P_{t-2}^T + \omega_2 P_{t-1}^T - \omega_2 \delta P_{t-2}^T \\
&\quad +N_t \\
Y_t &= (1+\delta)Y_{t-1} - \delta Y_{t-2} + \omega_0 P_t^T - [\omega_0(1+\delta) - \omega_1 - \omega_2]P_{t-1}^T \\
&\quad -(\omega_1 + \omega_2)P_{t-2}^T + N_t
\end{aligned}$$

Thus, we will have the first and second lags of the original series and pulse series and its the first and second lags in the model.

Also, we will include relevant components in the N_t series. These components for the ukcars series are a trend and seasonal components as per the existence of trend and seasonality are obvious from the time series plot.

So, we need to set intervention point T and the series Y_t, P_t^T, P_{t-1}^T, and P_{t-2}^T:

```
Y.t = ukcars
T = 95 # The time point when the intervention occurred
P.t = 1*(seq(ukcars) == T)
P.t.1 = Lag(P.t,+1)
P.t.2 = Lag(P.t,+2)
```

```r
model1 = dynlm(Y.t ~ L(Y.t , k = 1 ) + L(Y.t , k = 2 ) +
                 P.t + P.t.1 + P.t.2 + trend(Y.t) + season(Y.t))
summary(model1)
```

```
##
## Time series regression with "ts" data:
## Start = 1977(3), End = 2005(1)
##
## Call:
## dynlm(formula = Y.t ~ L(Y.t, k = 1) + L(Y.t, k = 2) + P.t + P.t.1 +
##     P.t.2 + trend(Y.t) + season(Y.t))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.737 -17.152   2.706  13.391  49.961
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     79.23572   13.97655   5.669 1.37e-07 ***
## L(Y.t, k = 1)    0.54585    0.09384   5.817 7.08e-08 ***
## L(Y.t, k = 2)    0.25841    0.09376   2.756  0.00694 **
## P.t            -50.93881   25.78406  -1.976  0.05093 .
## P.t.1          -17.54467   25.29748  -0.694  0.48957
## P.t.2          -29.24781   25.13591  -1.164  0.24733
## trend(Y.t)       1.80419    0.53586   3.367  0.00108 **
## season(Y.t)Q2  -33.30585    7.12195  -4.677 9.06e-06 ***
## season(Y.t)Q3 -100.25988    8.27130 -12.121  < 2e-16 ***
## season(Y.t)Q4  -21.06810   11.73872  -1.795  0.07568 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.47 on 101 degrees of freedom
## Multiple R-squared:  0.9123, Adjusted R-squared:  0.9045
## F-statistic: 116.8 on 9 and 101 DF,  p-value: < 2.2e-16
```

At 5% level of significance,

- trend and seasonal effects of the first three quarters are all significant,
- the pulse effect is slightly insignificant and its both lags are insignificant,
- effects of both lags of the original series are significant, and
- the model is significant.

We got a high adjusted R-square for this model.

Let’s have a residual check for this model.

```
residual.analysis(model1 , std=TRUE , Ljung.Box=FALSE)
```

```
## Loading required package: leaps
```

```
## Loading required package: ltsa
```

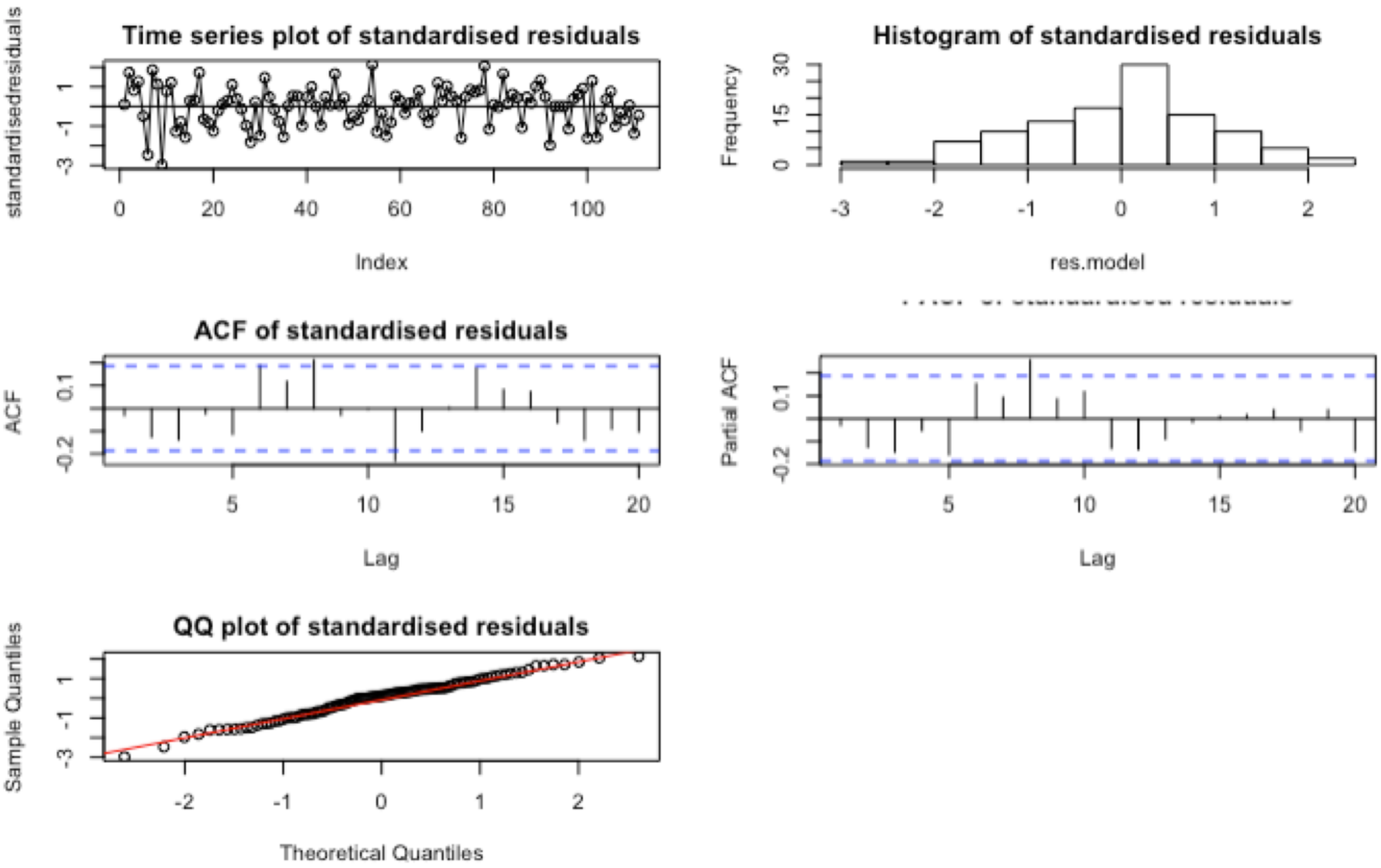
```
## Loading required package: bestglm
```

```
##
## Attaching package: 'FitAR'
```

```
## The following object is masked from 'package:forecast':
##
##     BoxCox
```

```
## The following object is masked from 'package:car':
##
##     Boot
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.98631, p-value = 0.32
```


Time series plot of standardised residuals
standardisedresiduals
-3
-1
1
0
20
40
60
80
100
Index
Histogram of standardised residuals
Frequency
0
15
30
-3
-2
-1
0
1
2
res.model
ACF of standardised residuals
ACF
-0.2
0.1
5
10
15
20
Lag
Partial ACF
-0.2
0.1
5
10
15
20
Lag
QQ plot of standardised residuals
Sample Quantiles
-3
-1
1
-2
-1
0
1
2
Theoretical Quantiles

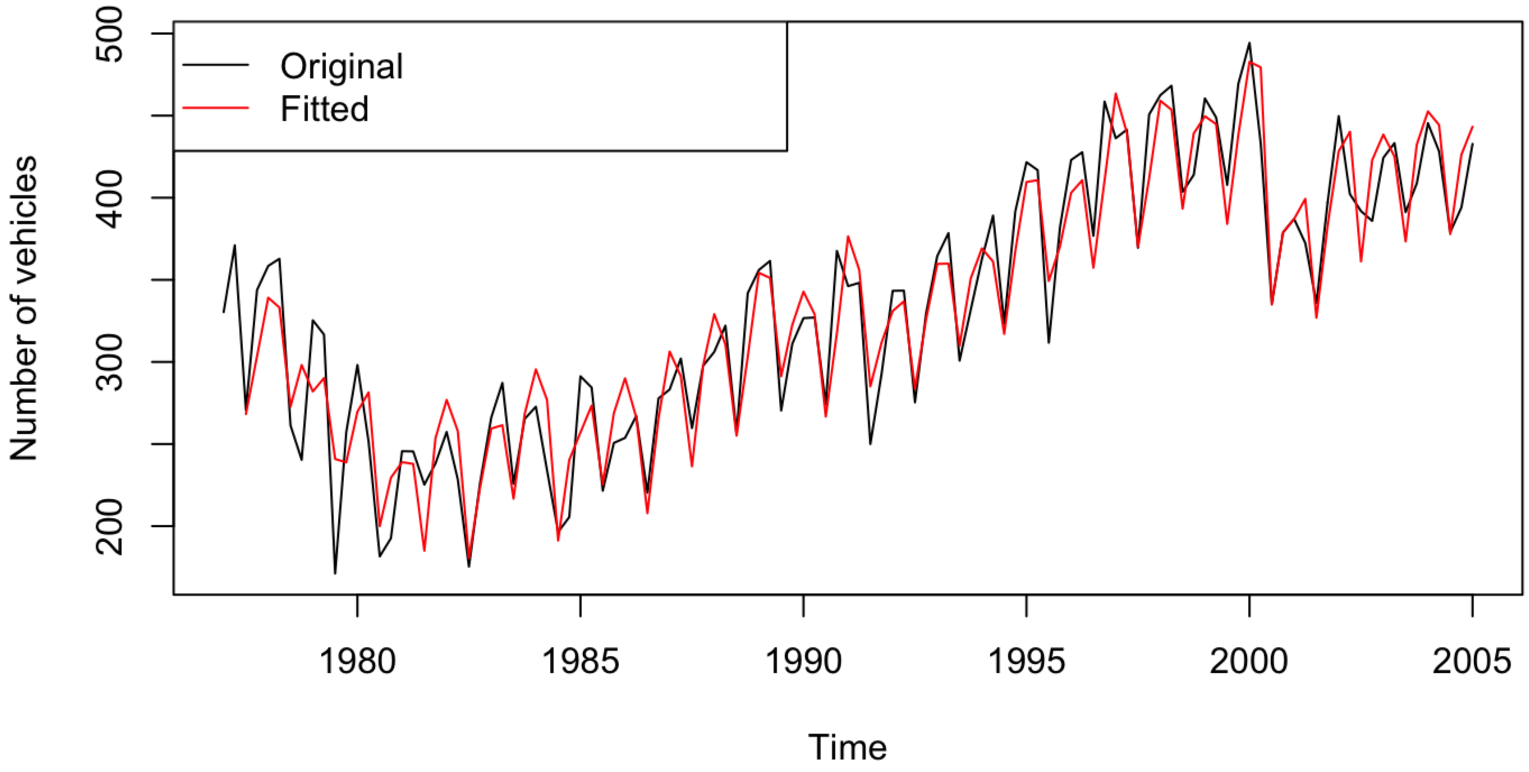
There is no non-random pattern with the standardised residuals and their histogram nearly symmetric and distributed within (-3,3) interval.

Normality of residuals can be concluded from both the QQ plot and Shapiro-Wilk test at 5% level of significance.

Although there are some slightly significant serial correlations in the sample ACF plot, it is not enough to conclude that the model is unable to capture the serial correlation in the series.

Let’s display the fitted and observed series and then add the forecasts to the fitted series.

```
plot(ukcars,ylab='Number of vehicles',type="l",col="black")
lines(model1$fitted.values, col = "red")
legend("topleft",lty=1, pch = -1, text.width = 11,
       col=c("black" , "red"), c("Original", "Fitted"))
```



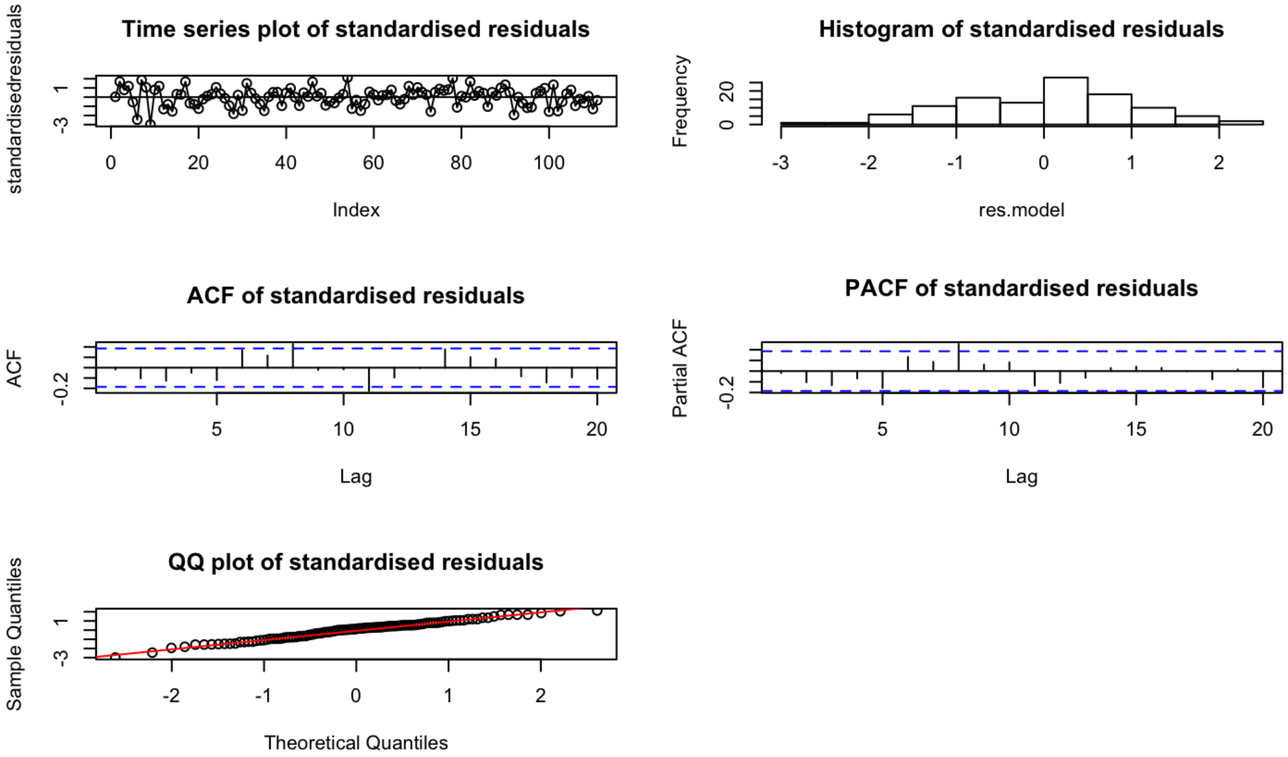
We can drop insignificant P^T_{t-1} and P^T_{t-2} from the model and fit it again.

```
model2 = dynlm(Y.t ~ L(Y.t , k = 1 ) + L(Y.t , k = 2 ) +
                 P.t + trend(Y.t) + season(Y.t))
summary(model2)
```

```
##
## Time series regression with "ts" data:
## Start = 1977(3), End = 2005(1)
##
## Call:
## dynlm(formula = Y.t ~ L(Y.t, k = 1) + L(Y.t, k = 2) + P.t + trend(Y.t) +
##     season(Y.t))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -70.655 -17.934   2.949  14.252  50.527
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     77.78825   13.91621   5.590 1.87e-07 ***
## L(Y.t, k = 1)    0.55684    0.09297   5.989 3.12e-08 ***
## L(Y.t, k = 2)    0.25231    0.09306   2.711  0.00785 **
## P.t            -49.98424   25.74037  -1.942  0.05489 .
## trend(Y.t)       1.70311    0.53003   3.213  0.00175 **
## season(Y.t)Q2  -32.31285    7.07184  -4.569 1.36e-05 ***
## season(Y.t)Q3  -99.11189    8.21958 -12.058  < 2e-16 ***
## season(Y.t)Q4  -19.79057   11.68781  -1.693  0.09343 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.45 on 103 degrees of freedom
## Multiple R-squared:  0.9108, Adjusted R-squared:  0.9047
## F-statistic: 150.2 on 7 and 103 DF,  p-value: < 2.2e-16
```

```
residual.analysis(model2 , std=TRUE , Ljung.Box=FALSE)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.98754, p-value = 0.3981
```



Now, there is no highly insignificant variable in the model. And residual diagnostics are the same as the first model.

Let’s compare AICs of the two models:

```
AIC(model1,model2)
```

```
##        df      AIC
## model1 11 1036.363
## model2  9 1034.334
```

```
BIC(model1,model2)
```

```
##        df      BIC
## model1 11 1066.168
## model2  9 1058.720
```

The second models are slightly better in terms of AIC and BIC and it includes less number of coefficients.

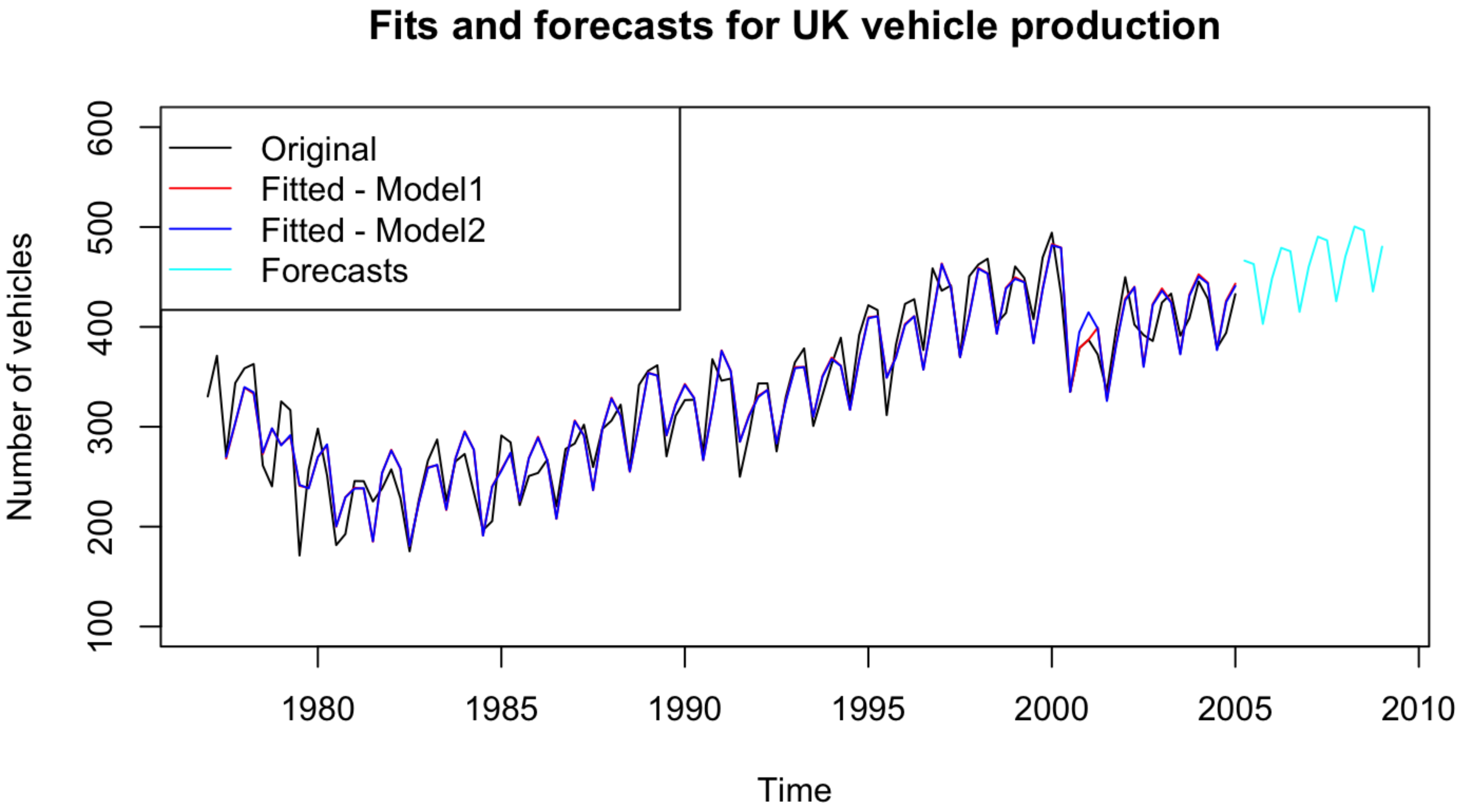
```r
q = 16 # The number of steps for forecasts
freq = 4 # The freuency of series
n = nrow(model2$model)
ukcars.frc = array(NA , (n + q))
ukcars.frc[1:n] = Y.t[3:length(Y.t)]

trend = array(NA,q)
trend.start = model2$model[n,"trend(Y.t)"] # Get the trend series
trend = seq(trend.start , trend.start + q/freq, 1/freq) # Add new trend points

for (i in 1:q){
  quarters = array(0,(freq-1))
  quarters[(i-1)%%freq] = 1 # Create new observations of the seasonal component
  data.new = c(1,ukcars.frc[n-1+i], ukcars.frc[n-2+i] ,
               P.t[n] ,  trend[i] , quarters)
  # Create a vector of new data composed of the model components
  ukcars.frc[n+i] = as.vector(model2$coefficients) %*% data.new
  # Calculate one-step ahead forecast
}
ukcars.frc[(n+1):(n+q)]
```

```
##  [1] 466.3201 462.8785 403.0474 448.6097 479.1010 475.6886 415.1083
##  [8] 460.2609 490.3351 486.5870 425.7146 470.6198 500.4825 496.5543
## [15] 435.5282 480.3024
```

```
plot(ukcars,ylab='Number of vehicles',type="l",col="black",
     xlim = c(1977, 2009), ylim = c(100,600), main = "Fits and forecasts for UK vehicle production")
lines(model1$fitted.values, col = "red")
lines(model2$fitted.values, col = "blue")
lines(ts(ukcars.frc[(n+1):(n+q)],start=c(2005,2),frequency = 4),col="cyan")
legend("topleft",lty=1, pch = -1, text.width = 11,
       col=c("black" , "red" , "blue", "cyan"), c("Original", "Fitted - Model1" , "Fitted - Model2" , "Forecasts
```


Fits and forecasts for UK vehicle production
Original
Fitted - Model1
Fitted - Model2
Forecasts
Number of vehicles
100
200
300
400
500
600
1980
1985
1990
1995
2000
2005
2010
Time

Notice that two models fit the series nearly at the same quality. However, the difference between the two models is seen in the period just after the intervention.

As expected, P_{t-1}^{T} and P_{t-2}^{T} are both help to capture the period after the intervention.

So, it would be a good idea to include these lags of the pulse series even if they are insignificant. Or at least have a check of the fit after the intervention is highly recommended.

Spurious correlation and prewhitening

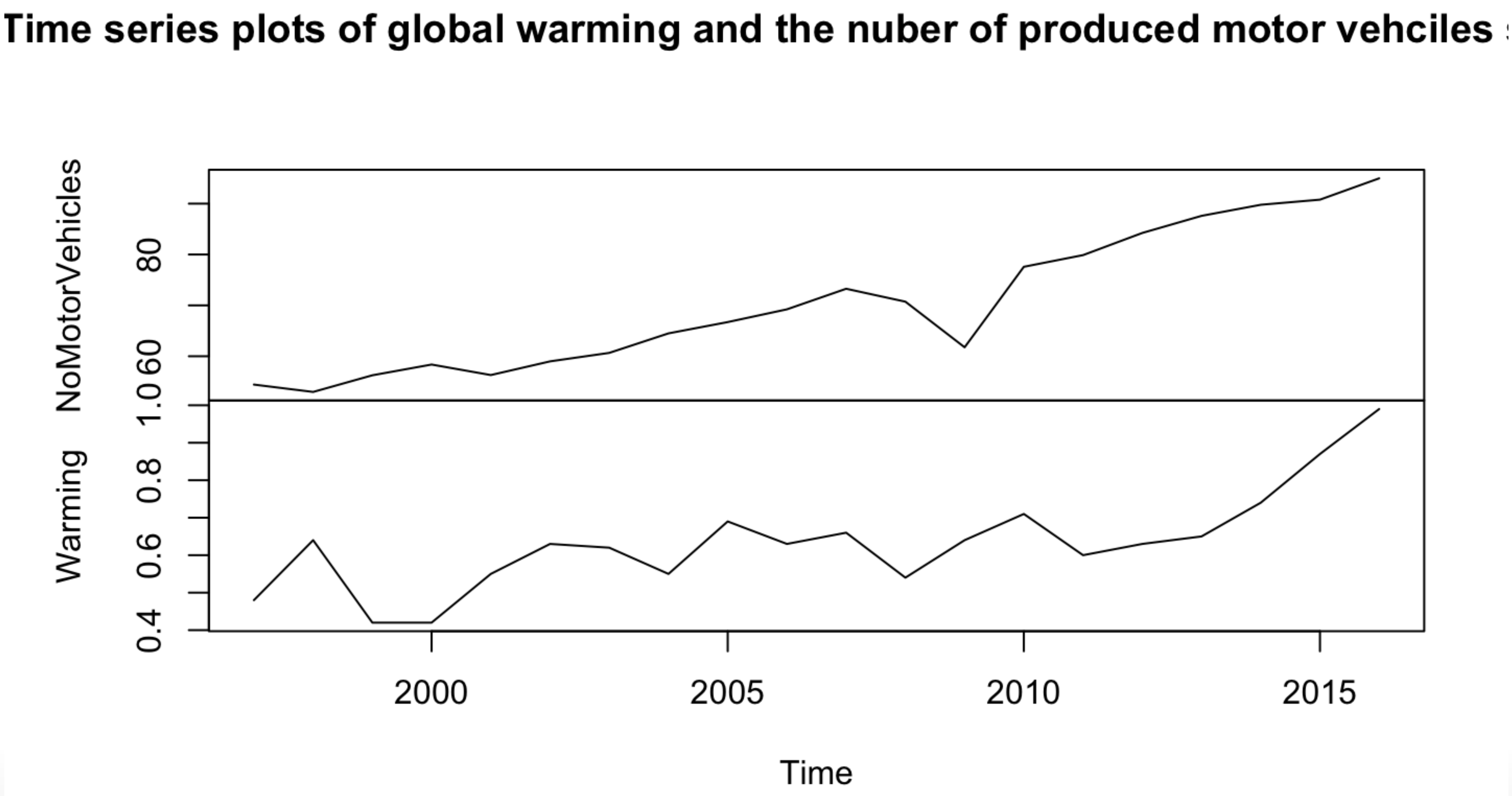
We will investigate whether the correlation between global warming and then umber of produced cars series focused on in Module 3 is spurious.

Let’s read data into R and convert the series into `ts` object.

```
vehicleWarming = read.csv("~/Documents/MATH1307_Forecasting/presentations/Module 3/vehicleWarming.csv")
vehicleWarming.ts = ts(vehicleWarming[,2:3], start = 1997)
```

And then let’s display time series plot.

```
plot(vehicleWarming.ts, main="Time series plots of global warming and the nuber of produced motor vehciles serie
```

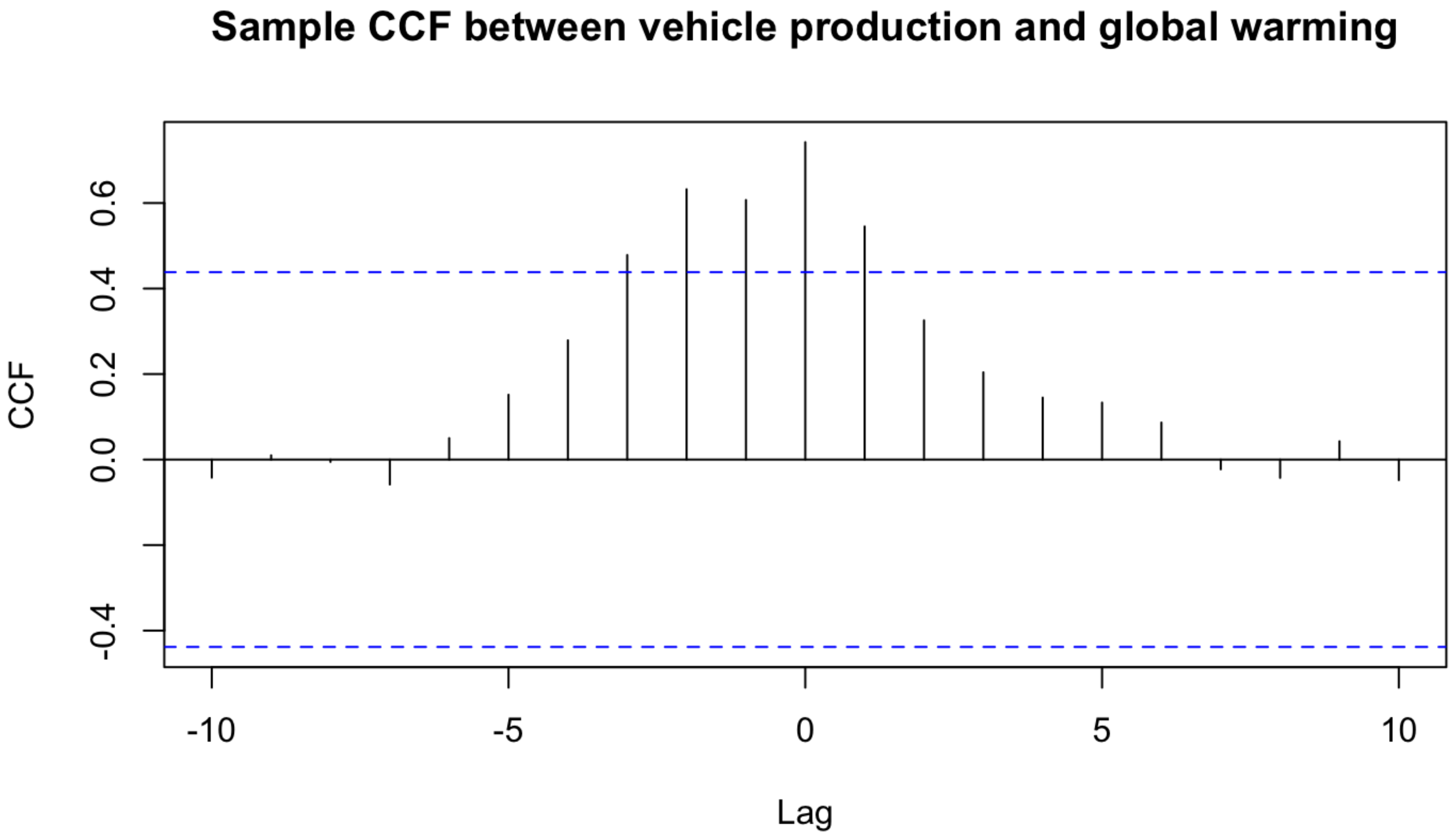


In both of the series, there are simultaneously increasing patterns in the time series plots.

This implies that the series is at least moderately correlated.

Let’s display sample CCF function to have a detailed look into the correlation structure between global warming and vehicle production series.

```
ccf(as.vector(vehicleWarming.ts[,1]), as.vector(vehicleWarming.ts[,2]),
    ylab='CCF', main = "Sample CCF between vehicle production and global warming")
```



There is not an unusually high correlation structure seen in the sample CCF plot.

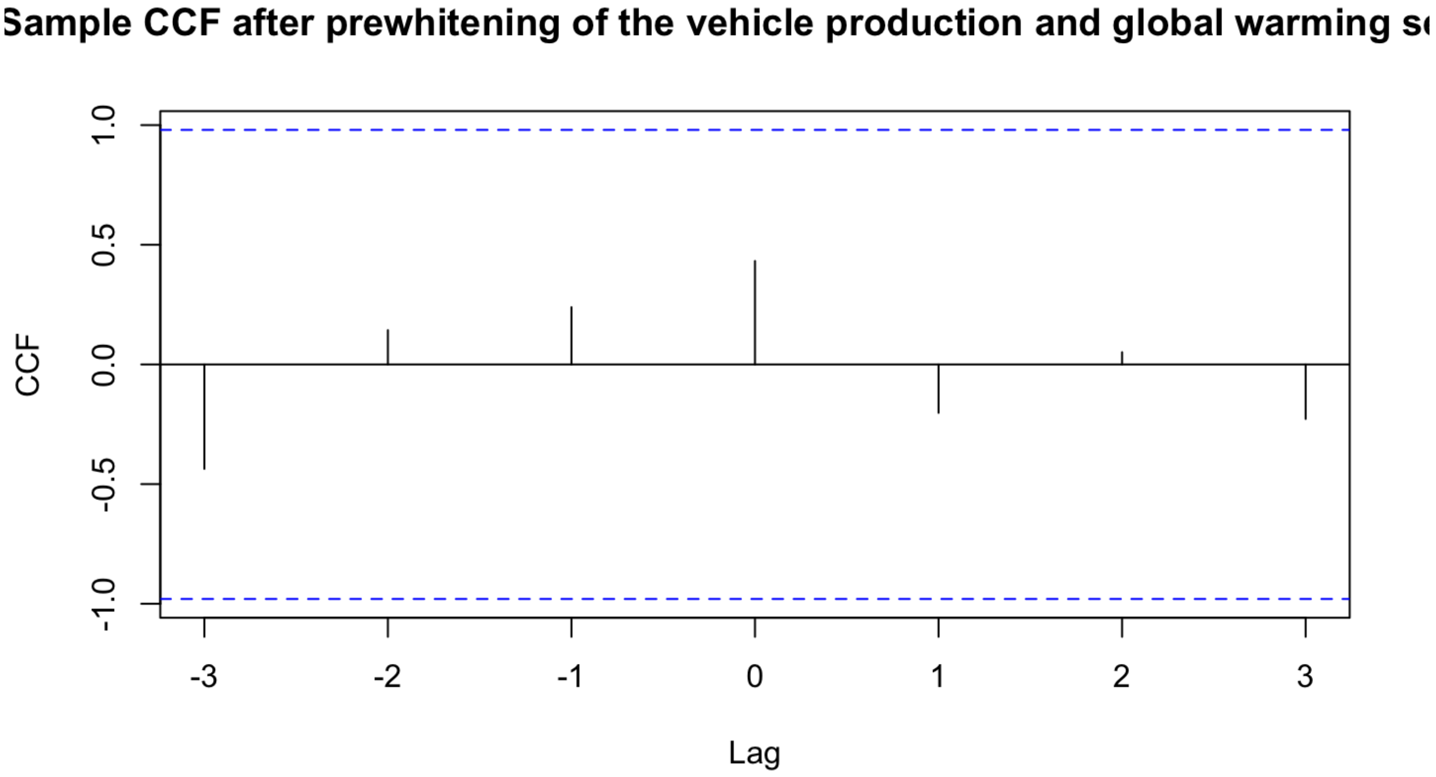
So, we can conclude that there is no significant evidence for the existence of a spurious correlation.

Now, although we did not find any spurious correlation, we will see how to prewhiten these series.

The expectation is to kill all existing correlation between the series through the prewhitening process.

```
me.dif=ts.intersect(diff(diff(vehicleWarming.ts[,1],12)),
                    diff(diff(vehicleWarming.ts[,2],12)))
prewhiten(as.vector(me.dif[,1]),as.vector(me.dif[,2]),ylab='CCF',
          main="Sample CCF after prewhitening of the vehicle production and global warming series")
```

```
## Warning in ar.ols(x, ...): model order: 4 singularities in the computation
## of the projection matrix results are only valid up to model order 3
```



As expected the prewhitening process removed all existing correlation structure between the global warming and vehicle production series.

Summary

In this module, we focused on some aspects of time series regression models to use information from other events or other time series to help model the time series of main interest.

We dealt with some interventions in the series. By this way, we attempted to incorporate known external events that we believe have a significant effect on the time series of interest into the model.

Then we illustrated the spurious correlation between two-time series, which we should be aware of while building time series regression models. To deal with the spurious correlation, we discussed methods involving prewhitening.

What’s next?

We will start with exponential smoothing and state-space models next week. Particularly, we will focus on

- Exponential smoothing methods
  - Simple Exponential smoothing
  - Holt’s method
  - Holt-Winters method
- State-space representation of these exponential smoothing methods.

Thanks for your attendance! Please use Socrative.com with room *FORECASTINGPG* to give feedback!