Multivariate Time Series

10

10.1 INTRODUCTION

In previous chapters, we have considered univariate time series, that is, those which involve a single variable. In that scenario, we utilize only the particular time series variable and its past values to fit a model and forecast the future. But in practice, we are often interested in forecasting quantities such as sales or costs which are influenced by other variables as well as past behavior of the dependent variable. In this chapter we will discuss two multivariate time series models:

(1) multiple linear regression (MLR) with correlated errors, and
(2) vector autoregressive (VAR) models.

(1) Multiple linear regression (MLR) with correlated errors
Multiple linear regression with correlated errors is, as its name suggests, very similar to standard multiple linear regression, except that past values of each predictor variable and possibly of the dependent variable can be used as independent variables when predicting the dependent variable. Furthermore, since the error terms are not uncorrelated as is assumed in standard multiple regression, time series methodology must be utilized for meaningful and valid analyses.

(2) Vector autoregressive (VAR) models
Vector autoregressive (VAR) modeling is another technique that is designed to accommodate more than one predictor variable (for example, forecasting sales when considering previous history of both sales *and* advertising). An unusually flexible characteristic of VAR models is that they do not require the analyst to specify which variables are dependent or independent, since there is no distinction between the two.

Examples using simulated and real data will be given to illustrate the two methods.

10.2 MULTIPLE REGRESSION WITH CORRELATED ERRORS

As mentioned, the method of multiple regression with correlated errors is a direct extension of the multiple regression model. However, in the time series setting, either the independent or dependent variables (or some combination of both) may depend on time and occur as realizations of the same length. Because of this dependence on time, it is common for the associated errors to be autocorrelated.

DOI: 10.1201/9781003089070-10

Due to the presence of multiple variables, notation in this chapter will be important. Recall that in univariate modeling, we consider realizations x_t for $t = 1, \cdots, n$ from a time series, X_t. Previous analysis techniques have been presented which use such an observed realization of length n to model the correlation structure within the time series using AR, ARMA, ARIMA, seasonal, etc. models. These models are then used to forecast future values of the univariate variable, that is, $X_{n+1}, X_{n+2}, \ldots$. As mentioned earlier, this basic idea and strategy will be similar for multiple regression with correlated errors.

10.2.1 Notation for Multiple Regression with Correlated Errors

We will denote a multivariate time series regression model (with m independent variables) by[1]

$$Y_t = \beta_0 + \beta_1 X_{t1} + \beta_2 X_{t2} + \ldots + \beta_m x_{tm} + Z_t \tag{10.1}$$

where Z_t is a zero-mean, stationary process. It is also possible for (10.1) to include a trend term and/or previous values of Y_t (denoted Y_{t-k}) and/or previous values of the X_{tj} variables as predictor variables of Y_t. Since each realization is of length n, this model will yield a set of n equations (where $t = 1, \cdots, n$). Because there is one dependent variable and m independent variables, we denote the corresponding $m \times (n+1)$ observations as

$$Y_1, X_{11}, X_{21}, \ldots X_{n1}$$

$$Y_2, X_{12}, X_{22}, \ldots X_{n2}:$$

$$Y_m, X_{1m}, X_{2m}, \ldots X_{nm}$$

Throughout this chapter, an important notational convention to remember is that the first subscript refers to the time point, while the second subscript refers to the independent variable number.

Key Point: Remember in this chapter that for the term X_{tj}, the first subscript refers to the time point, while the second subscript refers to the independent variable number.

Our approach will be to proceed as in standard multiple regression, with the expectation that the error terms may be correlated, and can be modeled as such. The selection of useful independent variables is similar to that in multiple regression with uncorrelated errors. Then, the resulting correlated error terms, Z_t, are modeled using n AR, ARMA, etc. model, which is included as part of the final model.[2]

To use the independent variables at previous time points as predictors of the dependent variable Y_t, we simply define a separate variable for each independent variable that corresponds to the various lag(s) of interest and enter these variables into the model. For example, if it is hypothesized that both independent variables X_2 and X_3 at a lag of $k = 1$ are important for predicting Y_t (but that X_1 does not have such a lagged relationship with Y_t, and neither do the previous values of Y_t), then the appropriate model corresponding to (10.1) is:

1 Note that in previous chapters the time series of interest was typically denoted by X_t. However, in the multiple regression section, Y_t will denote the time series of interest (dependent variable) while X_{tj}s are the independent variables.

2 We have chosen to model the residual series, Z_t, using an AR model.

$$Y_t = \beta_0 + \beta_1 X_{t1} + \beta_2 X_{t-1,2} + \beta_3 X_{t-1,3} + Z_t \tag{10.2}$$

Again, it can be assumed that the residual series Z_t may be correlated and will be modeled accordingly as an AR model.

10.2.2 Fitting Multiple Regression Models to Time Series Data

The procedure we will use to estimate the parameters in (10.2) is as follows:

(a) Use standard MLR procedures to obtain estimates $\hat{\beta}_0, \hat{\beta}_1$, and $\hat{\beta}_3$.
(b) Transform the data using $\hat{Z}_t = Y_t - \hat{\beta}_0 - \hat{\beta}_1 X_{t1} - \hat{\beta}_2 X_{t2} - \hat{\beta}_3 X_{t3}$.
(c) Model $\hat{Z}_t$ as an AR(p) model $\hat{\phi}(B)\hat{Z}_t = a_t$ where a_t is well modeled as white noise.
(d) Find the ML estimates of the parameters $\beta_0, \beta_1, \beta_2, \phi_1, \cdots, \phi_p$ in the multiple regression model with correlated errors where the variance-covariance matrix of the error process is based on the covariance structure of the AR fit. The ML estimates are obtained using the Base R command **`arima`**.

The procedure is illustrated in Example 10.1.

Example 10.1
Suppose that a corporation is interested in forecasting future sales and has evidence or intuition that sales are influenced by the following independent variables: TV advertising expenditures, online advertising expenditures, and the product discount being offered. These data can be found in the ***tswge*** dataframe **`Bsales.`** Data points are provided for 100 weeks, meaning that for each of the four variables, there is a corresponding time series realization with length $n = 100$. The resulting notation is as follows.

Sales (in thousands of dollars): Y_t, for $t = 1, 2, \cdots 100$; that is, $Y_1, Y_2, \cdots Y_{100}$
TV advertising (in thousands of dollars): $X_{t1} : X_{11}, X_{21}, \ldots X_{100,1}$
Online advertising (in thousands of dollars): $X_{t2} : X_{12}, X_{22}, \ldots X_{100,2}$
Discount (in percent): $X_{t3} : X_{13}, X_{23}, \ldots X_{100,3}$

Time series plots for each of these four variables are shown in Figures 10.1(a)–(d). Figure 10.1(a) reveals the cyclic behavior of sales, with a period of noticeably less extreme fluctuation during weeks 40–55; that is, the sales appear to somewhat "flatten out". Figures 10.1(b) and (c) show similar cyclic behavior of TV advertising and online advertising, respectively, but with a less pronounced decrease in fluctuation during weeks 40−55. Figure 10.1(d) is much different from the previous three plots, in that discount remains at zero% or 10% for many consecutive weeks, and all other fluctuations in the plots are quite patterned. Overall, it is not clear from these plots how the independent variables are related to sales, if at all. The code that generates the plots is given below:

```
par(mfrow=c(2,2))
data(Bsales)
sales=Bsales$sales
ad_tv=Bsales$ad_tv
ad_online=Bsales$ad_online
discount=Bsales$discount
plotts.wge(sales,xlab="Week",ylab="Dollars (in thousands)c")
plotts.wge(ad_tv,xlab="Week",ylab=" Dollars (in thousands)")
plotts.wge(ad_online,xlab="Week",ylab=" Dollars (in thousands)")
plotts.wge(discount,xlab="Week",ylab=" Dollars (in thousands)")
```

As a first attempt to model this dataset, lag variables will not be included in the model, which can be written as

$$Sales_t = \beta_0 + \beta_1 ad\_tv_t + \beta_2 ad\_online_t + \beta_3 discount_t + Z_t.$$

The following code is used to perform the analysis, where **Bsales** is a ***tswge*** dataframe.

```
mlrfit = lm(sales~ad_tv+ad_online+discount)
#Base R function for multiple linear regression
aic.wge(mlrfit$residuals, p=0:8, q=0)
#Selects the optimal p for an AR(p) fit to the residuals- chooses p=7
#the residuals were stored in mlrfit.
# The following computes the ML estimates
fit = arima(sales,order=c(7,0,0),xreg=cbind(ad_tv,ad_online,discount))
fit
```

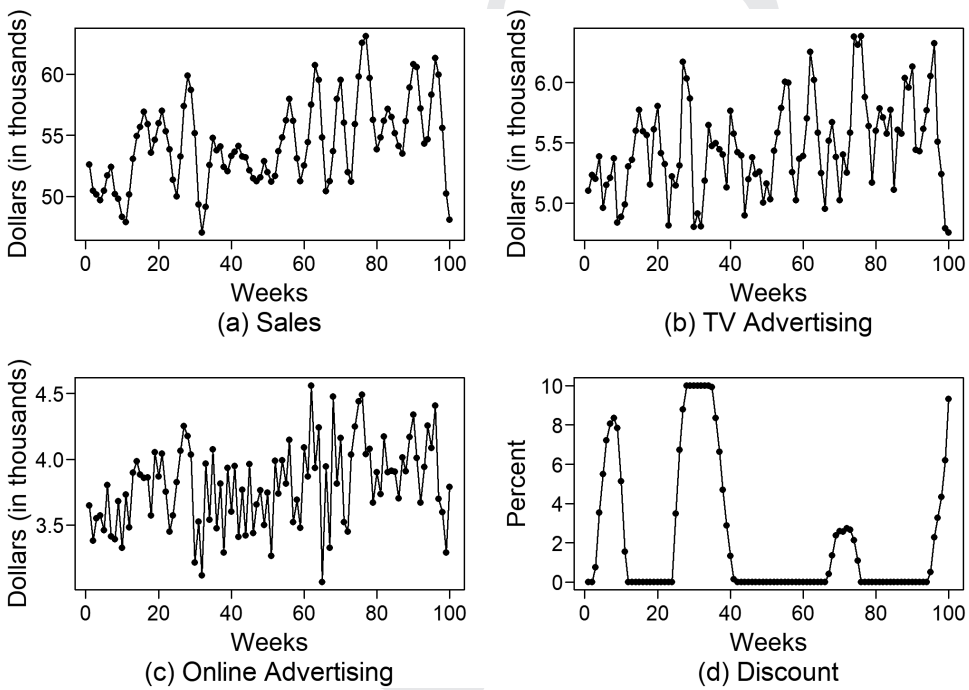


FIGURE 10.1 (a) Sales, (b) TV advertising expense, (c) online advertising expense, and (d) discount.

A summary of the output from the data object **fit** providing parameter coefficients, standard errors, and the ratios of coefficients to parameters is given in Table 10.1.

TABLE 10.1 Summary of Output with No Lagged Independent Variables

	INTERCEPT	*AD_TV*	*AD_ONLINE*	*DISCOUNT*
Coefficients	54.5513	0.0703	–0.0934	–0.1514
S.E	2.204	0.3434	0.2075	0.1315
Ratio	24.75	0.20	–0.45	–1.15

The resulting model is

$$Sales_t = 54.55 + 0.07ad\_tv_t - 0.09ad\_online_t - 0.1514discount_t + Z_t \quad (10.1)$$

where the error terms Z_t are modeled as an AR(7) process, according to the estimate of the optimal p in **aic.wge.** However, this overall model is unsatisfactory because none of the three independent variables is significant. While p-values are not provided by the function **arima,**[3] we will use a rule of thumb that a variable is significant (at the 5% significance level) if the absolute value of the coefficient exceeds two times the standard error. Consistent with the insignificant predictor variables, further evidence against the adequacy of this model is provided by the diagnostics of the residuals from the AR(7) fit to the residuals of the MLR model.[4] Figure 10.2 shows the sample autocorrelations resulting from the residuals of the AR(7) fit, and reveals that five of the 20 residuals extend beyond the 95% limit lines. Additional confirmation against white noise is suggested by the extremely small p-value from the Ljung-Box test for white noise, where $K = 24$.[5] Corresponding code and output are given below.

```
plotts.sample.wge(fit$resid,arlimits=TRUE)
ljung.wge(fit$resid,p=7)
ljung.wge(fit$resid,p=7,K=48)
```

While the residuals in Figure 10.2(a) from the MLR model fit have the general appearance of white noise, there are several large sample autocorrelations and the p-values from the Ljung-Box test at $K = 24$ and $K = 48$ are both less than .0001. Thus, the MLR has modeled the data adequately and we need to keep searching for a more appropriate model.

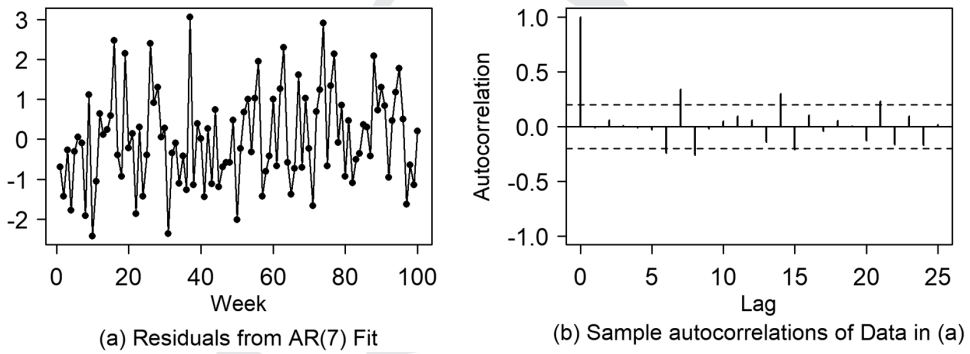


FIGURE 10.2 (a) Residuals from Model (10.1) and (b) sample autocorrelations.

Key Point: It is important to keep in mind that in an MLR with correlated errors analysis, there are two distinct sets of residuals.

1. The initial set of residuals are a result of the initial MLR fit to the data. These will be modeled as an AR(p) model.

3 The Base R function **arima** fits an ARIMA(p,d,q) model, where appropriate estimates of p, d, and q are selected by minimizing AIC.

4 Note that there are two sets of residuals in consideration; one set of residuals is the initial set resulting from the initial MLR fit to the data while the second set are residuals from an AR(p) fit to the residuals remaining after the MLR fit.

5 Recall that the null hypothesis in a Ljung-Box test is that the residuals are white noise, so that small p-values provide evidence against white noise. The default value for number of lags is $K = 24$. We typically use $K = 24$ and $K = 48$.

2. A second set of residuals are the residuals (hopefully white noise) from an AR(p) fit to the residuals remaining after the MLR fit. These will be called the "model residuals".

This chapter's diagnostic tests (sample autocorrelation plots and Ljung-Box tests) of the residuals refer to the final set of residuals, that is, from an AR(p) fit to the residuals remaining after the MLR fit.

10.2.2.1 Including a Trend Term in the Multiple Regression Model

In Chapter 8 the idea of including a trend term in a time series model was discussed. A trend term can also be included in the current setting, but the same caveat applies–that is, care must be taken to avoid adding a trend term to model a dataset that does not have a true deterministic trend. In this example, perhaps adding a trend term can improve the previous model. Here, trend will be the week number (from $t = 1,2,\cdots 100$) and the model of interest is given by

$$Sales_t = \beta_0 + \beta_1 t + \beta_2 ad\_tv_t + \beta_3 ad\_online_t + \beta_4 discount_t + Z_t.$$

We slightly modify the previous code (see code comments above) by adding a trend term, denoted as "t", where $t = 1,2,\cdots 100$. Note that the addition of the trend term results in AIC choosing an AR(6) model for the residuals. The code is as follows:

```
t = 1:100
mlrfit = lm(sales~ t+ad_tv+ad_online+discount)
aic.wge(mlrfit$residuals, p=0:8, q=0)
#AIC selects p=6 when fitting the residuals remaining after the MLR fit
fit=arima(sales, order=c(6,0,0),xreg=cbind(t,ad_tv,ad_online,discount))
fit
```

The summarized output providing parameter coefficients, standard errors, and the ratios of coefficients to parameters is given in Table 10.2.

TABLE 10.2 Summary of Output with No Lagged Independent Variables, but Including Trend Term

	INTERCEPT	*T*	*AD_TV*	*AD_ONLINE*	*DISCOUNT*
Coefficients	51.9224	0.0465	0.1123	–0.0508	–0.1701
S.E	2.2242	0.0148	0.3549	0.1939	0.1052
Ratio	23.34	3.14	0.32	–0.26	–1.62

This output suggests the candidate model

$$Sales_t = 51.92 + 0.05t + 0.11 ad\_tv_t - 0.05 ad\_online_t - 0.17 discount_t + Z_t \quad (10.2)$$

where Z_t is modeled as an AR(6).

In this case, the trend (week number) is significant, but the two advertising independent variables and discount variable are still insignificant, indicating that this model is probably not adequate. The AR(6) model selected as the best fit to the MLR residuals again produces final residuals that are inconsistent with white noise. Figure 10.3 shows the sample autocorrelations resulting from the residuals of the AR(6) fit to the MLR residuals, and again several autocorrelations fall outside the 95% limit lines. Furthermore, the

Ljung-Box test p-value is less than 0.0001 at both $K = 24$ and $K = 48$. Corresponding code and output are given below.

```
plotts.sample.wge(fit$resid,arlimits=TRUE)
ljung.wge(fit$resid,p=6)
ljung.wge(fit$resid,K=48,p=6)
```

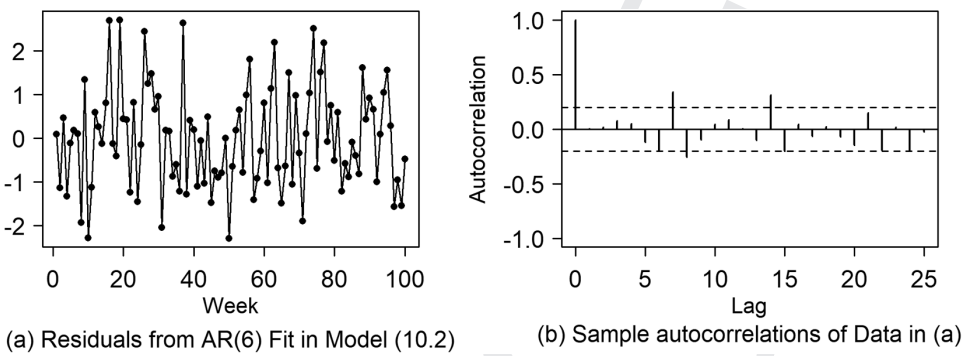


FIGURE 10.3 (a) Residuals from AR(6) fit in model (10.2) and (b) sample autocorrelations.

10.2.2.2 Adding Lagged Variables

The previous two modeling attempts have not taken into consideration the possibility that a relationship may exist between the dependent variable and lagged versions of various independent variables. That is, for example, it is quite possible that the independent variables could have a delayed effect on sales, meaning that advertising in week $t-1$ may affect sales in week t. So, we create lagged variables (of lag 1) for TV advertising and online advertising, *ad_tv_1* and *ad_online_1*, respectively. The R function used to create lagged variables is **`dplyr::lag`**. For a detailed tutorial on the **`dplyr::lag`** function, see the video link referenced by the following QR code. For this model, we will also include the discount variable at time t but will exclude trend, and will fit the model

$$Sales_t = \beta_0 + \beta_1 ad\_tv_{t-1} + \beta_2 ad\_online_{t-1} + \beta_3 discount_t + Z_t.$$

QR 10.1 Business Sales-Example 10.1

The code is as follows.

```
ad_tv1=dplyr::lag(ad_tv,1)#Creating lag 1 for ad_tv
ad_online1=dplyr::lag(ad_online,1)#Creating lag 1 for ad_online
discount=discount #No lag for discount
ad_tv1=ad_tv1 #Add lag (k=1) ad_tv1 to dataset
ad_online1=ad_online1 #Add lag (k=1) ad_online1 to dataset
mlrfit=lm(sales~ad_tv1+ad_online1+discount)#least sqr regression
aic.wge(mlrfit$residuals,p=0:8,q=0) #AIC selects p=7 #fit to residuals
fit=arima(sales,order=c(7,0,0), xreg=cbind(ad_tv1,ad_online1,discount))
fit
```

Key Point: Creating lagged variables introduces missing data ("NA") values in the new dataset. Some R functions, such as **`lm`** and **`arima`**, know by default to omit such a line of data that contains "NA" values. For other functions, such as **`VAR`** and **`VARSelect(`**to be introduced in Section 10.3) the analyst must take an extra step to subset the data to exclude the line(s) of data containing "NA" values.

A summary of the output produced by the above code is seen in Table 10.3.

TABLE 10.3 Summary of Output with Lagged Independent Variables, and No Trend Term

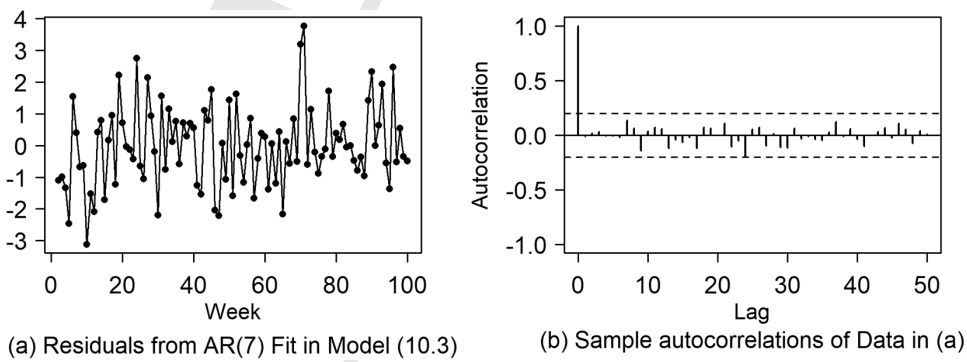
	INTERCEPT	*AD_TV(T-1)*	*AD_ONLINE(T-1)*	*DISCOUNT*
Coefficients	4.8382	3.4341	8.1152	–0.0573
S.E	2.827	0.6166	1.2447	0.0281
Ratio	1.71	5.57	6.52	–2.04

From the output, the final model fit is given by

$$Sales_t = 4.84 + 3.43 ad\_tv_{t-1} + 8.12 ad\_online_{t-1} - 0.06 discount_t + Z_t \tag{10.3}$$

where Z_t is fit by an AR(7) model. Interestingly, we find that at the 5% significance level, both advertising variables are now highly significant, and the discount variable is marginally significant. The residuals in Figure 10.4(a) have the appearance of white noise, and all sample autocorrelations in Figure 10.4(b) fall within the 95% limit lines. The Ljung-Box test has a p-value of 0.282 and 0.631, at $K = 24$ and $K = 48$, respectively. This suggests that we have an appropriate model. Corresponding code and output are given below.

```
plotts.sample.wge(fit$resid[2:100],lag.max=50,arlimits=TRUE)
ljung.wge(fit$resid[2:100],p=7)
ljung.wge(fit$resid[2:100],p=7,K=48)
```



(a) Residuals from AR(7) Fit in Model (10.3)

(b) Sample autocorrelations of Data in (a)

FIGURE 10.4 (a) Residuals from AR(7) fit in model (10.3) and (b) sample autocorrelations.

10.2.2.3 Using Lagged Variables and a Trend Variable

The inclusion of lagged variables produces much more satisfactory diagnostic test results overall, but for a final modeling attempt, trend is now included in the previous model containing the significant lagged variables and discount variable to fit the model

$$Sales_t = \beta_0 + \beta_1 t + \beta_2 ad\_tv_{t-1} + \beta_3 ad\_online_{t-1} + \beta_4 discount_t + Z_t.$$

The following code is used.

```
t=1:100 #Adding 100 trend weeks. Remaining code is similar to previous example.
ad_tv1=dplyr::lag(ad_tv,1)
ad_online1=dplyr::lag(ad_online,1)
mlrfit=lm(sales ~ t+ ad_tv1+ad_online1+discount)
aic.wge(mlrfit$residuals,p=0:8,q=0) #AIC selects p=7
fit=arima(sales,order=c(7,0,0),xreg=cbind(t,ad_tv1,ad_online1,discount))
fit
```

The resulting output is seen in Table 10.4.

TABLE 10.4 Summary of Output with Lagged Independent Variables, Including Trend Term

	INTERCEPT	*T*	*AD_TV(T-1)*	*AD_ONLINE(T-1)*	*DISCOUNT*
Coefficients	6.2215	0.0065	3.318	7.8248	–0.0453
S.E	2.782	0.0038	0.6288	1.302	0.0276
Ratio	2.24	1.71	5.28	6.01	–1.64

From the output we see that

$$Sales_t = 6.22 + 0.0065t + 3.32 ad\_tv_{t-1} + 7.82 ad\_online_{t-1} - 0.0453 discount_t + Z_t, \quad (10.4)$$

where Z_t is modeled by an AR(7).

Here, we observe again that both advertising variables are highly significant, but in this model, discount is insignificant. The trend variable (week) is technically insignificant (based on significance level .05), but AIC prefers the model including the trend variable over the previous model without trend.[6] The final residuals are modeled by an AR(7), and as was true of the previous model, the diagnostics for these residuals support white noise. In particular, Figure 10.5(a) shows the residuals which have the appearance of white noise, and the sample autocorrelations of the residuals are plotted in Figure 10.5(b) where it can be seen that all sample autocorrelations fall within the 95% limit lines. The Ljung-Box test has *p*-values 0.264 and 0.617, at $K = 24$ and $K = 48$, respectively. Corresponding code and output are given below.

```
plotts.sample.wge(fit$resid[2:100],lag.max=50,arlimits=TRUE)
ljung.wge(fit$resid[2:100],p=7)
ljung.wge(fit$resid[2:100],p=7,K=48)
```

6 The AIC value is provided in the data object **fit**, and in the previous model was given as 354.86; for this model including trend, the AIC value is 352.87. Remember, however, that it is not recommended to make final model decisions based completely on AIC values.

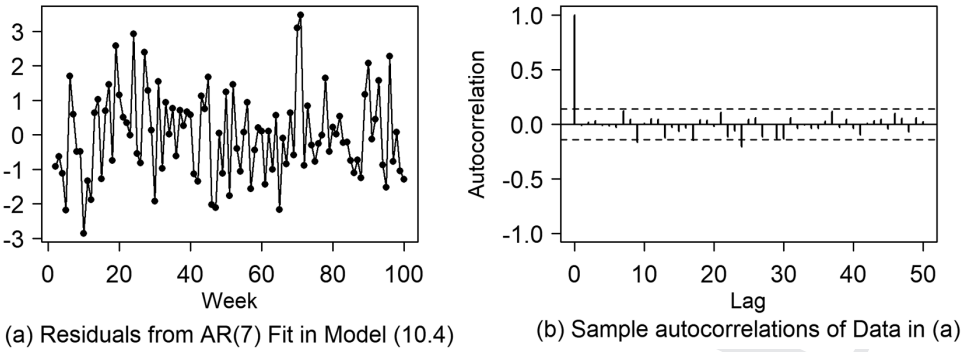


FIGURE 10.5 (a) Residuals from AR(7) fit in model (10.4) and (b) sample autocorrelations.

While including the trend term appears to have improved the model, remember that caution must be taken before including a trend in a model without strong evidence to do so. Because the significance of the trend variable is questionable, this is a case in which the more conservative decision is to not include the trend variable until additional data or domain knowledge gives reason to justify that the trend is real. However, the improvement in the model was noticeable enough that adding the trend term deserves attention and further consideration.

This example emphasizes the added value of using lagged variables for forecasting future values of a time-dependent response variable and also illustrates the strategy used to accommodate correlated residual terms.

10.2.3 Cross Correlation

In Example 10.1 it was shown that when forecasting sales at a given time point t, it was beneficial to use the previous time point $t-1$ of advertising expenses. A question that arises is how one knows, without testing many lags and assessing the significance of each, which lags of which variables should be included in the model. A useful statistical tool for detecting the existence of lagged relationships in time series data is the *cross-correlation function*. The cross-correlation between variables X_{t1} and X_{t2} at lag k is the correlation between $X_{t+k,1}$ and X_{t2}. The general formula for calculating the sample cross-correlations at lag k given a realization of length n for the ordered pairs (X_{t1}, X_{t2}) is given by

$$\hat{\rho}_{ij}(k) = \sum_{t=1}^{n-k} \frac{\left(X_{t+k,i} - \bar{X}_i\right)\left(X_{tj} - \bar{X}_j\right)}{\sqrt{\sum_{t=1}^{n}(X_{ti} - \bar{X}_i)^2}\sqrt{\sum_{t=1}^{n}(X_{tj} - \bar{X}_j)^2}}, \tag{10.3}$$

where i and j represent the variables and k represents the number of lags between the variables. For example, $\hat{\rho}_{12}(k)$ essentially calculates the correlation based on the ordered pairs $(X_{1+k,1}, X_{12}), (X_{2+k,1}, X_{22}), \cdots (X_{n1}, X_{n-k,2})$. Note that if $k = 0$, $\hat{\rho}_{12}(0)$ calculates the correlation from the ordered pairs $(X_{11}, X_{12}), (X_{21}, X_{22}), \cdots (X_{n,1}, X_{n,2})$, which is the setting of the familiar Pearson's correlation coefficient. However, when $k \neq 0$, the cross-correlation formula calculates correlation across (either forward or backward) between two variables. For example, if $k = 2$, then $\hat{\rho}_{12}(2)$ considers the ordered pairs $(X_{31}, X_{12}), (X_{41}, X_{22}), (X_{51}, X_{32}) \cdots (X_{n1}, X_{n-2,2})$, so that the correlation is calculated with variable X_2 being paired with data values of X_1 two time points ahead. However, if $k = -3$, then, $\hat{\rho}_{12}(-3)$ considers the ordered pairs $(X_{11}, X_{42}), (X_{21}, X_{52}), (X_{31}, X_{62}) \cdots (X_{n-3,1}, X_{n2})$, so that the correlation is calculated with variable X_2 being paired with data values of X_1 three time points ago.

Key Point: The cross-correlation between variables X_{t1} and X_{t2} at lag k is the correlation between $X_{t+k,1}$ and $X_{t,2}$. If $k = 0$, the definition is equivalent to the standard Pearson's correlation coefficient introduced in elementary statistics courses. Depending on whether k is positive or negative, the cross-correlation between $X_{t+k,1}$ and X_{t2} is calculated by pairing X_2 values at time t with X_1 values at k lags ahead of, or previous to, time t, respectively.

It is typical to plot the sample cross-correlations for variables X_{ti} and X_{tj} using a plot similar to the sample autocorrelation plot, where the height of a vertical bar indicates the strength of the cross-correlation at a specific lag. In Figures 10.6(a)–(c), we plot the sample cross-correlations from Example 10.1 between the dependent variable sales X_{t1} and each of the independent variables X_{t2} (TV advertising), X_{t3} (online advertising), and X_{t4} (discount). For the cross-correlations between sales and TV advertising (Figure 10.6(a)), and between sales and online advertising (Figure 10.6(b)), we see a positive spike at lag $k = -1$, indicating there is positive correlation between sales at time t and advertising (both TV and online) at time $t-1$. In other words, there is evidence of substantial positive correlation between sales and advertising lagged by a single week. The cross-correlation between sales and discount (Figure 10.6(c)) reveals no such apparent spike, which suggests that there is no lagged relationship between these two variables. Note that the results suggested by these cross-correlation function ("CCF") plots are consistent with the findings from the modeling procedure in Example 10.1. The Base R function which plots the cross-correlations is **ccf**; the commands to produce the plots in Figures 10.6(a)–(c) are given below.

```
ccf(ad_tv,sales) #Figure 10.6(a)[7]
ccf(ad_online,sales) #Figure 10.6(b)
ccf(discount,sales) #Figure 10.6(c)
```

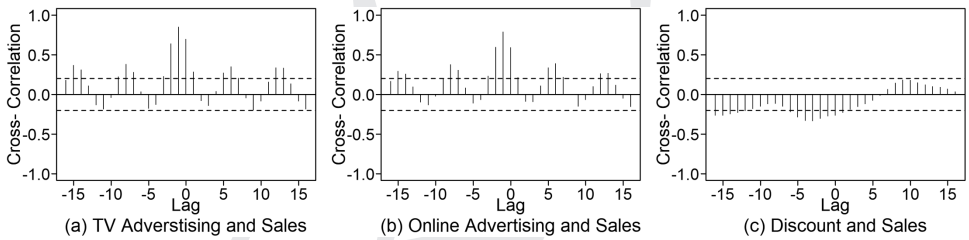


FIGURE 10.6 Cross-correlations between (a) TV advertising and sales, (b) online advertising and sales, and (c) discount and sales.

Caution should be advised due to inconsistencies among authors and software packages in defining the cross-correlation function. To be consistent with Base R, we have defined the cross-correlation function between X_{t1} and X_{t2} at lag k as the correlation between $X_{t+k,1}$ and X_{t2}. That is, the first variable reflects the time shift. The corresponding Base R syntax is **`ccf(x1,x2)`.** However, some authors (for example, Woodward et al. 2017) have defined the cross-correlation function between X_{t1} and X_{t2} at lag k as the correlation between X_{t1} and $X_{t+k,2}$. This is remedied in R by reversing the syntax as **`ccf(x2,x1)`.**

QR 10.2
Cross-correlation

7 If the order of the variables is reversed, that is, **`ccf(sales,ad_tv)`**, the spike would appear at lag $k = 1$.

Key Point: Be careful with the definition of cross-correlation among various authors and software packages. While we have defined the cross-correlation function between X_{t1} and X_{t2} at lag k as the correlation between $X_{t+k,1}$ and X_{t2} (as in Base R), it can also be defined as the correlation between X_{t1} and $X_{t+k,2}$. When using R, simply reverse the order of the variables in the syntax to make the definitions match your preference.

10.3 VECTOR AUTOREGRESSIVE (VAR) MODELS

The multiple linear regression with correlated errors method does not take into account the possible correlation structure within and among the independent variables. However, the objective in vector autoregressive (VAR) modeling is to investigate the interrelationships among all variables of interest in order to improve forecasts for one or more of the variables. Since vector autoregressive (VAR) models do not distinguish between dependent and independent variables, the goal is to use all of the variables and to simultaneously forecast all variables.

Key Point: In multiple linear regression (MLR) with correlated errors, the correlation between explanatory variables is somewhat of a liability due to multicollinearity concerns, while VAR leverages such correlation to improve model fitting and corresponding forecasts.

The notation for VAR will remain the same as for MLR with correlated errors, where again the first subscript refers to the time point while the second subscript refers to the variable number.

We first consider the "bivariate VAR(1) process", in which two dependent variables are being modeled but only one time lag is considered for both of the independent predictor variables (i.e. $m = 2$ and $p = 1$). Recall in the univariate case that we often write an AR(1) model as

$$X_t = (1 - \phi_1)\mu + \phi_1 X_{t-1} + a_t.$$

In the univariate AR(1) setting, the model involves the lag 1 value X_{t-1}.

In a bivariate VAR(1) model, there are two variables, X_{t1} and X_{t2}, and these values involve the lag 1 values $X_{t-1,1}$ and $X_{t-1,2}$. The equations for the bivariate VAR(1) model are more complex than for the AR(1) model due to the interrelationships between the two variables and their lag 1 time points. The bivariate VAR(1) model can be expressed as

$$\begin{aligned} X_{t1} &= (1 - \phi_{11})\mu_1 - \phi_{12}\mu_2 + \phi_{11}X_{t-1,1} + \phi_{12}X_{t-1,2} + a_{t1} \\ X_{t2} &= -\phi_{21}\mu_1 + (1 - \phi_{22})\mu_2 + \phi_{21}X_{t-1,1} + \phi_{22}X_{t-1,2} + a_{t2}, \end{aligned} \tag{10.4}$$

where a_{t1} and a_{t2} are the residuals for the models corresponding to Variable 1 and Variable 2, respectively.

Because of the notation complexity, VAR models are often written in a more convenient and abbreviated matrix form. For example, the matrix notation for a bivariate VAR(1) model (with zero mean) is

$$\begin{pmatrix} X_{t1} \\ X_{t2} \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{pmatrix} \begin{pmatrix} X_{t-1,1} \\ X_{t-1,2} \end{pmatrix} + \begin{pmatrix} a_{t1} \\ a_{t2} \end{pmatrix},$$

QR 10.3 VAR Model

where

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} (1-\phi_{11})\mu_1 - \phi_{12}\mu_2 \\ -\phi_{21}\mu_1 + (1-\phi_{22})\mu_2 \end{pmatrix}$$

If the bivariate VAR(1) model is extended to a bivariate VAR(2) model, there will be two matrices of coefficients ϕ_{ij}. For this reason, we modify the notation so that, for example, ϕ_{11} will be denoted as $\phi_{11(1)}$ for the lag 1 matrix component and denoted as $\phi_{11(2)}$ for the lag 2 matrix component:

$$\begin{pmatrix} X_{t1} \\ X_{t2} \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \phi_{11(1)} & \phi_{12(1)} \\ \phi_{21(1)} & \phi_{22(1)} \end{pmatrix} \begin{pmatrix} X_{t-1,1} \\ X_{t-1,2} \end{pmatrix} + \begin{pmatrix} \phi_{11(2)} & \phi_{12(2)} \\ \phi_{21(2)} & \phi_{22(2)} \end{pmatrix} \begin{pmatrix} X_{t-2,1} \\ X_{t-2,2} \end{pmatrix} + \begin{pmatrix} a_{t1} \\ a_{t2} \end{pmatrix}.$$

Here, the vector $\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}$ is an even more complicated pair of linear combinations of μ_1 and μ_2 than above for the bivariate VAR(1) model. If we multiply the set of matrices, we obtain the expanded version of the bivariate VAR(2) model:

$$\begin{aligned} X_{t1} &= \beta_1 + \phi_{11(1)}X_{t-1,1} + \phi_{12(1)}X_{t-1,2} + \phi_{11(2)}X_{t-2,1} + \phi_{12(2)}X_{t-2,2} + a_{t1} \\ X_{t2} &= \beta_2 + \phi_{21(1)}X_{t-1,1} + \phi_{22(1)}X_{t-1,2} + \end{aligned} \tag{10.5}$$

Note that both $X_{t1}$and X_{t2} depend on lagged $t-1$ and $t-2$ values of X_{t1} and X_{t2}.

Obviously, writing the VAR(p) equations in expanded form (and even in matrix form) will quickly become tedious for higher orders and for more than two variables. Next we will establish the forecasting methodology used for VAR(p) models.

10.3.1 Forecasting with VAR(p) models

Forecasting with VAR(p) models is an extension of forecasting with AR(p) models, which was introduced in Chapter 6. Recall that forecasts for an AR(p) model are based on the underlying AR(p) equation and are calculated by

$$\hat{X}_{t_0}(\ell) = \bar{x}(1-\phi_1-\ldots-\phi_p) + \phi_1\hat{X}_{t_0}(\ell-1) + \ldots + \phi_p\hat{X}_{t_0}(\ell-p).$$

Specifically, the ℓ-step ahead forecasts, $\hat{X}_{t_0}(\ell)$, for the univariate variable X_t depend on $\ell-1$ through $\ell-p$ step ahead forecasts, i.e. $\hat{X}_{t_0}(\ell-1),\ldots,\hat{X}_{t_0}(\ell-p)$, some of which may be actual observed values.

For the sake of simplicity, we will show forecasts for the bivariate VAR(1) model given in (10.4). The forecasts for $\hat{X}_{t_0 1}(\ell)$ and $\hat{X}_{t_0 2}(\ell)$ are given by

$$\hat{X}_{t_0 1}(\ell) = (1-\phi_{11})\bar{x}_1 - \phi_{12}\bar{x}_2 + \phi_{11}\hat{X}_{t_0 1}(\ell-1) + \varphi_{12}\hat{X}_{t_0 2}(\ell-1)$$

$$\hat{X}_{t_0 2}(\ell) = -\phi_{21}\bar{x}_1 + (1-\phi_{22})\bar{x}_2 + \phi_{21}\hat{X}_{t_0 1}(\ell-1) + \phi_{22}\hat{X}_{t_0 2}(\ell-1).$$

Specifically, the ℓ-step ahead forecasts $\hat{X}_{t_0 1}(\ell)$ for the variable X_{t1} and for X_{t2} depend on $\ell-1$ step ahead forecasts for both variables X_{t1} and X_{t2}.

QR 10.4 VAR Model Forecasts

Example 10.2 A Simulated Example

This example illustrates how multivariate techniques detect that one variable is a leading indicator of another variable and how such a relationship is used advantageously in VAR forecasts. The following two time series realizations of length $n = 25$ were generated from AR(2) models and are given below.

```
x1.25=c( -1.03, 0.11, -0.18, 0.20, -0.99, -1.63, 1.07, 2.26, -0.49, -1.54, 0.45,
0.92, -0.05, -1.18, 0.90, 1.17, 0.31, 1.19, 0.27, -0.09, 0.23, -1.91, 0.46,
3.61, -0.03)
```

```
x2.25=c( -0.82, 0.54, 1.13, -0.24, -0.77, 0.22, 0.46, -0.03, -0.59, 0.45, 0.59,
0.15, 0.60, 0.13, -0.04, 0.12, -0.96, 0.23, 1.81, -0.01, -0.95, -0.55, -0.15,
0.71, 0.90)
```

These time series are plotted in Figure 10.7(a), where **x1.25** values are shown as solid dots connected by a solid line and **x2.25** values are represented by open circles connected by dashed lines. From Figure 10.7(a), it is not clear that there is a relationship between the two datasets. However, for $t = 6,7,\ldots,20$, the data were created in such a way that $x_{t+5,1}$ is very close to the value $2x_{t2}$. This relationship is shown in Figure 10.7(b). The first five values of **x1.25** and the last five values of **x2.25** are not related in any special way. The cross-correlations between **x1.25** and **x2.25** are shown in Figure 10.7(c). The strong positive cross-correlation at $k = 5$ is the correlation between $X_{t+5,1}$ and X_{t2}, which is high by construction of the datasets.[8] The cross-correlations were obtained using the code

```
ccf(x1.25,x2.25) ## cross-correlation also shows the significant lag at 5 ##
```

8 Notice that three spikes exceed the limit lines. We conclude that lag five is most influential because it is the most extreme. Just as in a sample autocorrelation plot, it is common for a particular prominent "spike" at a given lag to have several nearby lags which also show relatively strong correlation. It is also important to note that if the **ccf** function in R had reversed the two variables, the most extreme lag would have appeared at $k = -5$. Be sure to pay close attention to this order, as cautioned in the previous Key Point.

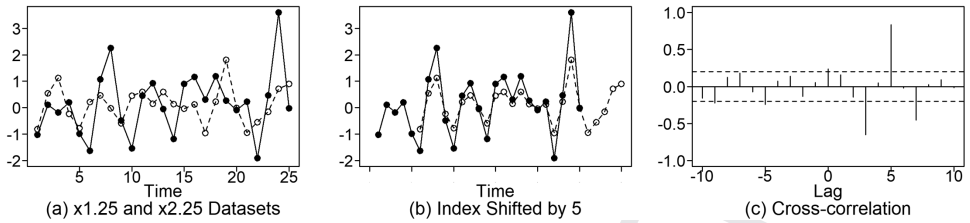


FIGURE 10.7 (a) Datasets `x1.25` and `x2.25`, (b) datasets `x1.25` and `x2.25` with `x2.25` shifted five time units to the right to show relationship, and (c) cross-correlation between `x1.25` and `x2.25`.

10.3.1.1 Univariate Forecasts

We first obtain univariate forecasts for each of the two time series realizations, using techniques discussed in Chapter 6. The following forecasts are obtained using the first 20 data values as a training set on which the forecasts are based, and the last five values are the test set which will be forecast. The training sets are

```
x1=x1.25[1:20]
x2=x2.25[1:20]
```

The univariate forecasts in Figure 10.8 were obtained using the following code.

```
p1=aic.wge(x1,p=0:8,q=0:0) # aic picks p=2
x1.est=est.ar.wge(x1,p=p1$p)
fore.arma.wge(x1.25,phi=x1.est$phi,n.ahead=5,lastn=TRUE,limits=FALSE)
#
p2=aic.wge(x2,p=0:8,q=0:0) # aic picks p=2
x2.est=est.ar.wge(x1,p=p2$p)
fore.arma.wge(x2.25,phi=x2.est$phi,n.ahead=5,lastn=TRUE,limits=FALSE)
#
```

When modeled as univariate models, the forecasts for both **`x1.25`** and **`x2.25`** are fairly poor. In Figure 10.8(a), the forecasts for times $t = 21$ to 25 for variable **`x1.25`** are superimposed on the last five corresponding actual points. The forecasts tend to remain relatively close to the last actual value, and therefore do not predict the strong oscillatory behavior of the actual data.

Similarly, in Figure 10.8(b), the last five forecasts for variable **`x2.25`** are overlaid on the last five corresponding actual points. These forecasts incorrectly predict the cycle length of the oscillatory behavior; the actual data steadily increase while the forecasts predict a sudden downturn at $t = 24$ which did not occur. Thus, univariate AR(p) forecasts are disappointing for both **`x1.25`** and **`x2.25`**.

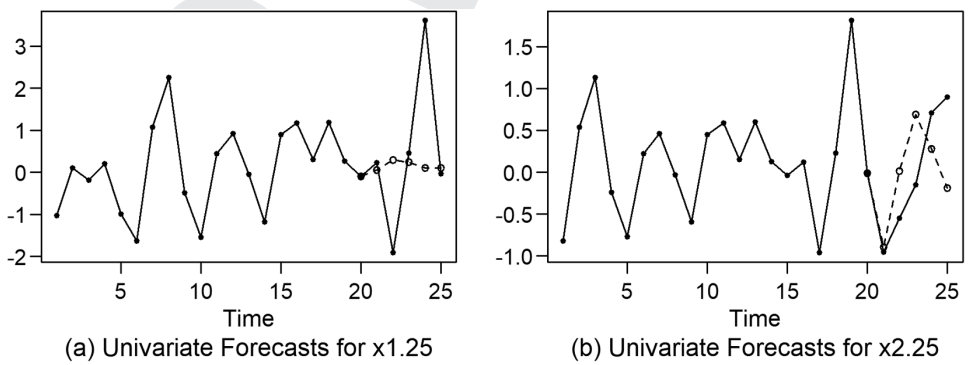


FIGURE 10.8 Univariate Forecasts for (a) `x1.25` and (b) `x2.25`.

10.3.1.2 VAR Analysis

The data in this example are designed to show (in a big way) the value of VAR over univariate forecasting. In this book, we will use the CRAN package ***vars*** for VAR analysis. The steps involved in VAR forecasting are the same as for the related univariate AR model. Recall that the data on which we will perform the VAR analysis are **x1** and **x2**, which consist of the first 20 values in each realization. We will combine these two datasets (which are stored as column vectors) into a matrix, **X**, using the command

```
X=cbind(x1,x2)
```

***1. Model Identification*:**
The first step is to identify the order p of the VAR(p) model. The function **VARselect** in the ***vars*** package is analogous in functionality to the **aic.wge** function in ***tswge***. **VARselect** will fit models up to a particular lag limit (**lag.max**) and will return AIC and other model identification criteria including BIC (referred to in the output as SC). [9]

```
# You will need to install and load the vars package from CRAN
library(vars)
VARselect(X,lag.max=6,type="const",season=NULL,exogen=NULL)
# AIC and BIC(SC) select p=5
```

The selection of a 5th-order VAR model is critical because there is a "lag 5" relationship built into the data. (Consequently, it was critical that we allowed $p = 5$ as a possible choice.)

2. Parameter Estimation:
Parameter estimation is performed using the ***vars*** command **VAR** as follows.

```
lsfit=VAR(X,p=5,type="const")
summary(lsfit) ##Note significance of 1 variable, at lag 5 ##
```

The output includes the following

```
Estimation results for equation x1:
===================================
x1=x1.l1+x2.l1+x1.l2+x2.l2+x1.l3+x2.l3+x1.l4+x2.l4+x1.l5+x2.l5+const


          Estimate     Std. Error     t value      Pr(>|t|)
x1.l1     -0.0042949    0.0077157       -0.557       0.607
x2.l1     0.0025313     0.0092102        0.275       0.797
x1.l2     0.0040630     0.0081828        0.497       0.646
x2.l2     -0.0077684    0.0103929       -0.747       0.496
x1.l3     -0.0047742    0.0077547       -0.616       0.571
x2.l3     0.0097459     0.0136252        0.715       0.514
x1.l4     0.0044834     0.0053700        0.835       0.451
x2.l4     -0.0079471    0.0140921       -0.564       0.603
x1.l5     -0.0021468    0.0053796       -0.399       0.710
x2.l5     2.0045969     0.0170018      117.905    3.1e-08 ***
const     -0.0001567    0.0055712       -0.028       0.979
```

The equation is the VAR representation of the first equation in (10.5) for a 5th-order bivariate model. The test results show that $X_{t-5,2}$ has a strong influence on X_{t1} and that there are no other significant

9 The BIC (Bayes information criterion) was derived by Gideon E. Schwarz; hence, SC (Schwarz criterion) is an alternate acronym for BIC.

relationships. A second set of tests representing the second equation in (10.5) show no significant lag relationships.

3. Forecasting:

The model fit information contained in **`lsfit`** is used to forecast, using the ***vars*** command **`predict`**. As in Figure 10.8, the forecasts for **`x1.25`** and **`x2.25`** for the last five values are shown superimposed on the actual values. Notice that in Figure 10.9(a) the forecasts for **`x1.25`** (shown as open circles) fall essentially "on top of" the actual data values (shown as smaller black circles). Obviously, **`x2.25`** is especially helpful in predicting **`x1.25`** by construction. However, the relationship between the two datasets wasn't obvious in Figure 10.7(a) but was "detected" by the VAR analysis and used to enhance the forecasts. On the other hand, Figure 10.9(b) reveals that the VAR forecasts for last five values of **`x2.25`** are no better than the corresponding univariate forecasts. This again is intuitive because, while **`x1.25`** was directly calculated from **`x2.25`**, the converse is not true.

This is an example of how the VAR model could provide excellent results without previously knowing the relationship between the variables. In a multiple regression, it would have been useful to use $X_{t-5,1}$ as an independent variable for predicting X_{t1}. However, this particular independent variable would probably only have been used if the analyst had prior "domain knowledge". In the case in which no domain knowledge was assumed to exist, the VAR model was flexible in the sense that it did not require the analyst to distinguish between independent and dependent variables. Instead, VAR methodology analyzed the relationships among all possible variables, and provided resulting forecasts for the variables. As with univariate model identification, in VAR model identification with **`VARselect`**, it is important to search for model orders over a wide sufficiently wide range of possible values. In this case, the range needed to include $p = 5$.

The forecasts and plots in Figure 10.9 can be obtained using the following code.

```
# VAR forecasting
preds=predict(lsfit,n.ahead=5)
```

The output **`preds`** contains the VAR forecasts for the last five data values. These are in the vectors **`f1.12`** and **`f2.12`** defined below.

```
f1.12=preds$fcst$x1[,1] # VAR forecasts for x1.
f2.12=preds$fcst$x2[,1] # VAR forecasts for x2

# Plotting Forecasts of x1.25
t=1:25
plot(t,x1.25,type="o",pch=20,cex=1,ylim=c(-1.75,3.75))
points(t[21:25],f1.12,type='o',cex=2,pch=1)

# Plotting Forecasts of x2.25
plot(t,x2.25,type="o",pch=20,cex=1,ylim=c(-1.75,3.75))
points(t[21:25],f2.12,type='o',cex=2,pch=1)
```

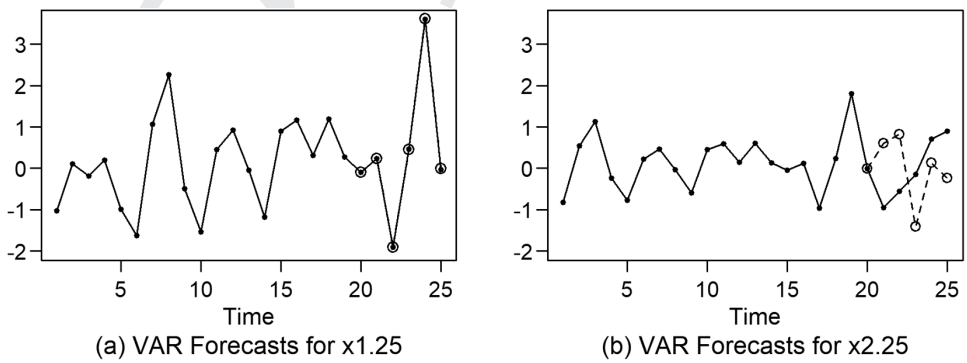


FIGURE 10.9 (a): Forecasts for x1.25 and (b) x2.25 using VAR Modeling.

Key Point: In multiple linear regression (MLR) with correlated errors, p refers to the order of the AR fit to the model residuals. In VAR modeling, p refers to the order of the model.

Example 10.3 VAR modeling using sunspot data and melanoma cases

Past research has suggested a likely relationship between sunspot numbers and melanoma cases. The relationship is time-dependent, and the speculation is that the number of melanoma cases[10] is related to the number of recorded sunspots in *previous* years (Houghton, Munster, and Viola, 1978). In particular, it was hypothesized that the lag is two years. In this example, we analyze data from 1936 to 1972 using VAR modeling. Figure 10.10(a) shows the **sunspot2.0** numbers for the years 1936–1972, Figure 10.10(b) is a plot of the melanoma cases during 1936–1972, and (c) shows the cross-correlations between sunspots and the melanoma incidences. The sunspots show the 10–11 year cycle while the melanoma cases show a rise during this time frame. The cross-correlation at lag $k = -2$, does not give much credibility to the "lag 2" hypothesis. However, we proceed as follows to see if sunspots tend to predict melanoma occurrence.

The data and the code for the cross-correlations are given below.

```
melanoma=c(1.0,0.9,0.8,1.4,1.2,1.0,1.5,1.9,1.5,1.5,1.5,1.6,1.8,2.8,2.5,2.5,
2.4,2.1,1.9,2.4,2.4,2.6,2.6,4.4,4.2,3.8,3.4,3.6,4.1,3.7,4.2,4.1,4.1,4.0,5.2,
5.3, 5.3)
sunspot=c(133,191,183,148,113,79,51,27,16,55,154,215,193,191,119,98,45,20,7,
54,201,269,262,225,159,76,53,40,15,22,67,133,150,149,148,94,98)
t=1:37
year=1935+t
plot(year,melanoma,type='o',pch=20)
plot(year,sunspot,type='o',pch=20)
ccf(sunspot,melanoma,ylim=c(-1,1))
```

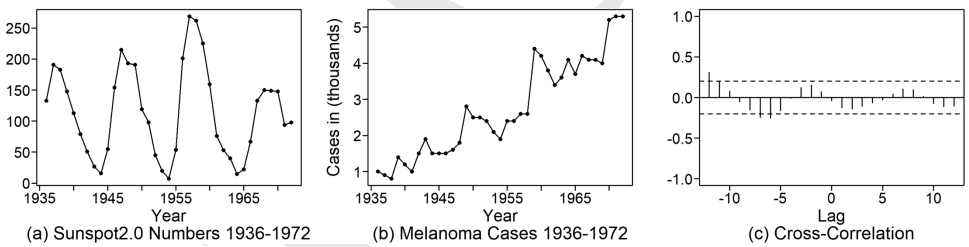


FIGURE 10.10 (a) Sunspot numbers and (b) melanoma cases between 1936 and 1972; (c) cross-correlation between melanoma cases and sunspots.

We further explore the possible relationship using VAR techniques to model the data and then use the model for forecasting. As in the simulated data in Example 10.2, we first consider forecasts for both melanoma incidents and sunspot numbers when each is modeled separately as a univariate time series.

For purposes of forecast cross-validation, we used the melanoma and sunspot data for the first 29 years (1936–1964) for model building. These (training) data are contained in the datasets **mel.64** and **sun.64** defined below.

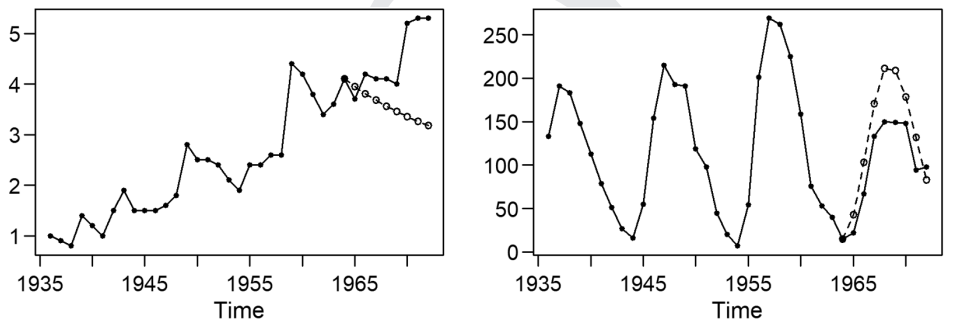
```
mel.64=melanoma[1:29]
sun.64=sunspot[1:29]
```

10 Number of incidences of melanoma skin cancer per 100,000 people in Connecticut.

We then use these models to forecast the last eight years (1965−1972). The forecasts for the melanoma and sunspot data are shown in Figures 10.11(a) and 10.11(b), respectively. Forecasts for the last eight years are shown with open circles connected by dotted lines. In each case, the univariate time series is modeled using a stationary model, and therefore the forecasts trend toward the mean. This is particularly evident in the melanoma forecasts where the resulting forecasts of melanoma incidences are quite poor. The univariate forecasts were obtained and plotted using the following code.

```
## Univariate analysis/forecasts for melanoma ##
p.mel=aic.wge(mel.64,p=0:10,q=0:0)
p.mel$p
mel.est=est.ar.wge(mel.64,p=p.mel$p)
pred_m=fore.arma.wge(mel.64,phi=mel.est$phi,n.ahead=8,lastn=FALSE,limits=
FALSE)
plot(year,melanoma,type='o',pch=20)
points(year[30:37],pred_m$f,type='o',lty=2,pch=1)

## Univariate analysis/forecasts for sunspot ##
p.sun=aic.wge(sun.64,p=0:10,q=0:0)
p.sun$p
sun.est=est.ar.wge(sun.64,p=p.sun$p)
pred_s=fore.arma.wge(sun.64,phi=sun.est$phi,n.ahead=8,lastn=FALSE,limits=
FALSE)
plot(year,sunspot,type='o',pch=20)
points(year[30:37],pred_s$f,type='o',lty=2,pch=1)
```



(a) Univariate Forecasts for Melanoma Data (b) Univariate Forecasts for Sunspot Data

FIGURE 10.11 Univariate forecasts for (a) melanoma data and (b) sunspot data for years 1965–1972.

VAR modeling follows the steps as before:

1. ***Use* `VARselect` *to pick the order p***

The following code combines the melanoma and sunspot training data into a matrix, **`X`**, and then uses ***vars*** command **`VARselect`** to select the order, allowing p to range up to **`lag.max=5`**. AIC selected five, but all the other criteria, including BIC (SC), selected order four. We will proceed using $p = 4$.

```
X=cbind(mel.64,sun.64)
VARselect(X, lag.max = 5, type = "const",season = NULL, exogen = NULL)
#AIC = 5, BIC picks 4 We go with 4
```

2. ***Use* `VAR` *to fit the VAR model on the training set***

The following command instructs VAR to fit a 4th-order model on the training set matrix, **`X`**. You can examine the VAR fit by issuing the command **`summary`**(**`VARfit`**).

```
VARfit=VAR(X,p=4,type='const') ## This fits 9 parameters ##
```

3. ***Forecast using*** **`predict`**

VAR models are complex, and our main goal is to determine whether sunspot information is a "leading indicator" of melanoma incidences. Figure 10.12 shows the VAR-based forecasts. The melanoma forecasts in Figure 10.12(a) no longer trend toward a mean, but in fact are remarkably accurate. The VAR forecasts effectively track the rapid increase in melanoma cases. The forecasts for sunspots are not improved and tend to exaggerate the height of the peak in 1968. However, using melanoma incidences to predict future sunspots does not make sense. It is physically intuitive that the one-directional relationship holds with sunspot numbers being an early indicator of later melanoma cases, but not vice versa. The VAR forecasts for melanoma incidences and sunspot numbers for the years 1965–1972 are found using the following code.

```
preds=predict(VARfit,n.ahead=8)
mel.f=preds$fcst$mel.64[,1] # VAR forecasts for mel.64
sun.f=preds$fcst$sun.64[,1] # VAR forecasts for sun.64
```

QR 10.5 VAR Modeling Sunspot and Melanoma-Example 10.3

The overlay plots of the data and the forecasts are obtained using the code:

```
t=1:37
year=t+1935
# melanoma forecasts
plot(year,melanoma,type="o",pch=20,cex=1,ylim=c(.5,6))
points(year[30:37],mel.f,type='o',cex=1,pch=1)
# sunspot forecasts
plot(year,sunspot,type="o",pch=20,cex=1)
points(year[30:37],sun.f,type='o',cex=1,pch=1)
```

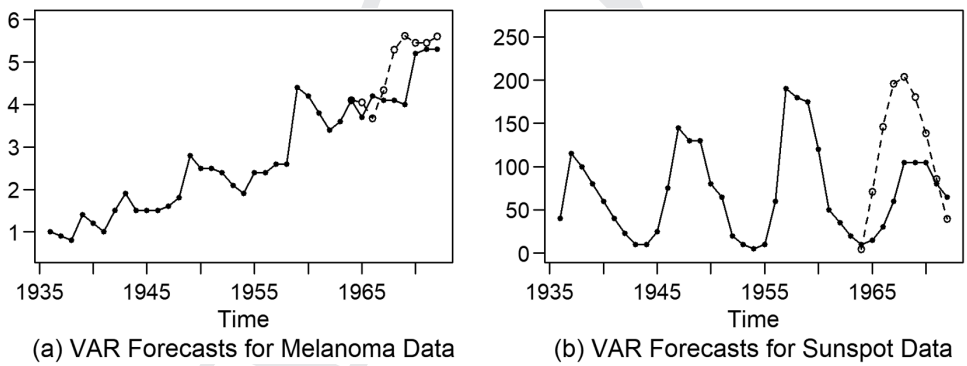


FIGURE 10.12 VAR forecasts for (a) melanoma data and (b) sunspot data for years 1965–1972.

10.3.1.3 Comparing RMSEs

Obviously, the VAR forecasts for melanoma incidences were far better than the univariate ones. To quantify the comparison, we use the RMSEs obtained using the commands

```
RMSE_AR=sqrt(mean((melanoma[30:37]-pred_m$f[1:8])^2))
RMSE_VAR=sqrt(mean((melanoma[30:37]-preds$fcst$mel.64[1:8])^2))
```

The RMSEs for univariate AR and multivariate VAR forecasts of melanoma incidences from 1965–1972 were 1.275 and .765, respectively.

10.3.1.4 Final Comments

We found that sunspot activity seemed to behave as a leading indicator of melanoma incidences. A couple of points need to be made.

(1) We haven't shown that high sunspot activity causes melanoma. We found an interesting relationship that would require much further investigation.
(2) The results we found did not point directly to a two-year lag relationship. The cross-correlations showed other lagged relationships (4–6 years) that were stronger (and negative).

10.4 RELATIONSHIP BETWEEN MLR AND VAR MODELS

In this chapter, the methodologies have been described and illustrated for multiple linear regression with correlated errors and for VAR, respectively. Partly due to the complexity of the VAR notation, it certainly seems that the two procedures are unrelated, other than the fact they both model multivariate time series. However, a very interesting (and surprising!) relationship between the two exists which may be helpful in better understanding both multivariate methods. For further detail, the reader is encouraged to see Appendix B.

10.5 A COMPREHENSIVE AND FINAL EXAMPLE: LOS ANGELES CARDIAC MORTALITY

In this example, we present a classic example (Shumway and Stoffer, 2017) in which the objective is to examine the extent to which cardiac mortality incidences can be predicted from average weekly temperature and air pollution measures. The dataset consists of weekly cardiac mortality, temperatures, and pollution measures for the years 1970−1978 and the first 40 weeks of 1979 in Los Angeles, California. Because it is suspected that cardiac mortality is related to temperature and pollution from *previous* time periods, the use of lagged variables seems appropriate. This example will illustrate the use of VAR modeling to predict cardiac mortality incidences; the modeling steps are shown in detail below. Originally obtained from the package ***astsa***, the three variables we will use in our study can be found in the ***ts*** object **cardiac** in ***tswge*** and are shown below. To familiarize yourself with the format of the dataset, consider the following code and output.

```
data(cardiac)
head(cardiac)
Time Series:
Start = c(1970, 1)
End = c(1970, 6)
Frequency = 52
         cmort  tempr part
1970.000  97.85 72.38 72.72
1970.019 104.64 67.19 49.60
1970.038  94.36 62.94 55.68
1970.058  98.05 72.49 55.16
1970.077  95.85 74.25 66.02
1970.096  95.98 67.88 44.01
```

The above output indicates that this multivariate time series dataset is composed of average weekly (f**requency=52**) cardiac mortalities (**cmort**), average weekly temperature (**tempr**) and average weekly number of particulates in the air (**part**). For each of these three variables, plots of the realization, sample autocorrelations, and Parzen spectral density estimate are shown in Figure 10.13. A decreasing trend is evident in the realization for cardiac mortalities over the ten-year period, and the realizations reveal strong evidence of an annual seasonal pattern for all three variables. The sample autocorrelation plots all show a 52-week sinusoidal cycle, and the Parzen spectral density plots all have a peak at approximately $.019 = 1/52$ in each of the spectral densities. The data in Figure 10.14 are the output from a 52^{nd}-order moving average smoother applied to the cardiac mortality data, which shows the downward trend in cardiac mortalities. Noting these behaviors will be vital to the modeling and forecasting procedures that follow.

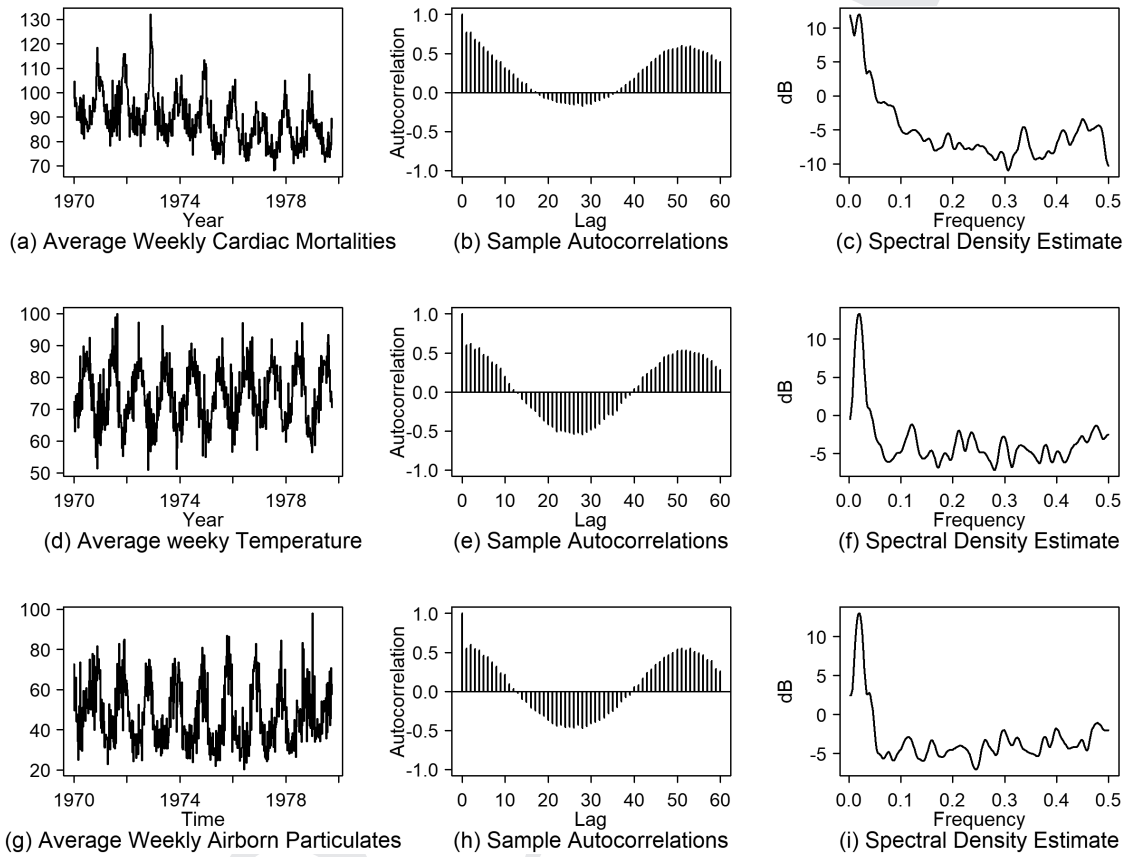


FIGURE 10.13 Realizations, sample autocorrelations, and spectral density estimates for (a)–(c) cardiac mortality data, (d)–(f) weekly temperature data, and (g)–(i) weekly airborn particulates. Note that `trunc=100` for the spectral density plots.

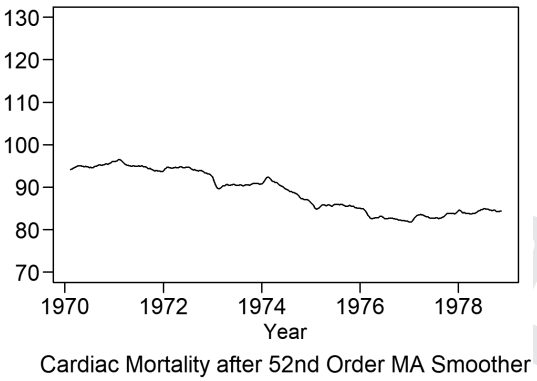


FIGURE 10.14 Cardiac mortality data after a 52nd-order centered moving average smoother.

10.5.1 Applying the VAR(*p*) to the Cardiac Mortality Data

10.5.1.1 Use `VARselect` *to Identify Candidate Model Orders*

The first step in generating a "baseline" model for forecasting the cardiac mortalities given the temperature and particulates is to use the AIC and/or BIC(SC) criterion to identify the order of the candidate VAR(*p*) model. Note that due to the previously observed evidence of a decreasing trend in the cardiac mortality series, we have included both an intercept and trend term in the model (**`type = "both"`**).

```
VARselect(cardiac, lag.max = 10, type = "both")
```

```
$selection
AIC(n)  HQ(n)   SC(n)   FPE(n)
     9      5       2        9

$criteria
               1        2        3        4        5        6        7        8        9       10
AIC(n)   11.7378 11.3019  1.2679 11.2303 11.1763 11.1527 11.1525 11.1288 11.1192 11.1202
HQ(n)    11.7876 11.3815 11.3774 11.3697 11.3456 11.3518 11.3814 11.3876 11.4078 11.4387
SC(n)    11.8646 11.5048 11.5469 11.5854 11.6076 11.6600 11.7359 11.7883 11.8547 11.9319
FPE(n)  125216.9 80972.3 78268.2 75383.7 71426.1 69758.3 69749.9 68122.4 67477.0 67556.5
```

In the output above, **`$criteria`** includes the AIC and BIC(SC) for each order, and **`$selection`** identifies the order that produced the lowest value of each criterion. The BIC(SC) has favored the VAR(2), while the AIC selects a VAR(9). As usual, the analyst is encouraged to consider domain knowledge and any other known and relevant information to make the final decision as to the value of *p* in the model. In addition, a good practice is to select a few values of *p* as candidates and evaluate the competing models by all available measures to choose the most appropriate *p*. This strategy will be implemented next by assessing a visualization and the RMSEs of the forecasts of the last 52 weeks of the series for the VAR(2) and the VAR(9). In addition, the VAR(7) will be included as a candidate model because there is evidence from the cross-correlation function that cardiac mortalities are correlated with particulates after a 7-week lag.

10.5.1.2 Use `VAR` to Fit the VAR Models to the Training Set

As in the previous example in this chapter, we will divide the data into a training set, which in this case consists of the first eight years plus the first 40 weeks of 1978. This leaves 52 weeks for the test set (the last 52 weeks of the dataset). [11] The following commands create the training and test sets.

```
cardiacTrain = window(cardiac, start = c(1970,1), end = c(1978,40))
cardiacTest = window(cardiac, start = c(1978,41), end = c(1979,40))
```

The next step is to fit the various candidate models using the following code.

```
CMortVAR2 = VAR(cardiacTrain, type = "both", p = 2)
CMortVAR9 = VAR(cardiacTrain, type = "both", p = 9)
CMortVAR7 = VAR(cardiacTrain, type = "both", p = 7)
```

An initial check of the appropriateness of the models is then conducted using the Ljung-Box test to check the residuals. The Ljung-Box commands for the default $K = 24$ are given below.

```
ljung.wge(CMortVAR2$varresult$cmort$residuals,p=2)
ljung.wge(CMortVAR9$varresult$cmort$residuals,p=9)
ljung.wge(CMortVAR7$varresult$cmort$residuals,p=7)
```

The *p*-values are .554, .215, and .069, respectively. Applying the Ljung-Box test using `K=48` yields *p*-values .779, .431, and .151, respectively. White noise seems to be a reasonable assumption for these lag ranges. (Remember that the analyst should also always plot and visually assess the residuals. (We recommend that you examine the plots.)

10.5.1.3 Use `predict` to Forecast Data Values in the Test Set

Finally, the forecasts are calculated, using a forecast horizon of 52. The code is as follows:

```
preds2=predict(CMortVAR2,n.ahead=52)
preds9=predict(CMortVAR9,n.ahead=52)
preds7=predict(CMortVAR7,n.ahead=52)
```

A visualization of forecast performance and the RMSE of the forecasts for each of the three models are obtained using the code below. The corresponding results are shown in Figure 10.15 and Table 10.5, respectively.

```
t=1:508
plot(t, cardiac[,"cmort"], type = "l", xlim = c(450,510), ylab = "Cardiac
Mortality", main = "52 Week Cardiac Mortality Forecast", xlab = "Time")
points(t[457:508], preds2$fcst$cmort[,1], type="l", lwd=2, lty=1)
points(t[457:508], preds9$fcst$cmort[,1], type="l", lwd=2, lty=2)
points(seq(457,508,1), preds7$fcst$cmort[,1], type="l", lwd=2,lty=3)

RMSE2 = sqrt(mean((cardiacTest[,"cmort"] - preds2$fcst$cmort[,1])^2))
RMSE9 = sqrt(mean((cardiacTest[,"cmort"] - preds9$fcst$cmort[,1])^2))
RMSE7 = sqrt(mean((cardiacTest[,"cmort"] - preds7$fcst$cmort[,1])^2))
```

11 Recall that the cardiac mortality data includes the years 1970–1978 and the first 40 weeks of 1979.

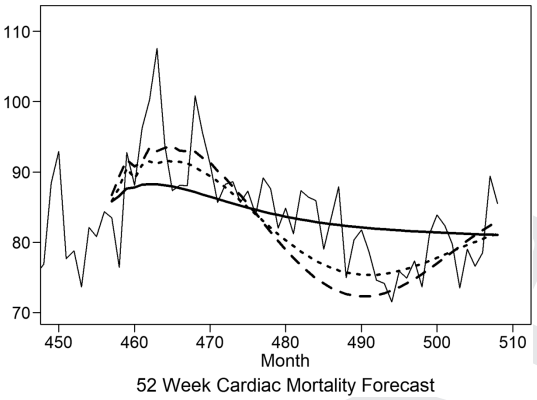


FIGURE 10.15 Visualization of the 52-week forecast of the VAR(2) (solid line), VAR(9) (dashed line), and VAR(7) (dotted line) models with trend with actual values for comparison.

TABLE 10.5 RMSEs for a Horizon of 52 Weeks for the VAR(2), VAR(9) and VAR(7) with Trend.

MODEL (WITH TREND TERM)	*RMSE*
Var(2)	5.92
Var(9)	6.08
Var(7)	**5.44**

The plot in Figure 10.15 shows that the VAR(2) seems to mostly model a decreasing trend in the cardiac mortality rates without much regard to any other behavior. On the other hand, closer inspection of the plots reveals that the VAR(7) and VAR(9) try to model the sharper increases and decreases for the first few weeks in the test set (with some success), and later weeks are predicted with a smooth curve that closely resembles the periodic and trending behavior evident in the series. Ultimately, the VAR(7) appears to most closely model the behaviors in the series which is also reflected by its superior RMSE of 5.44. It is safe to say, George Box would agree, that none of these are the "right" model, although the visualizations and RMSEs suggest that the VAR(7) is the most useful of the three.

10.5.2 The Seasonal VAR(*p*) Model

10.5.2.1 Use `VARselect` to Identify Candidate Seasonal Models

Recall that there was strong evidence of seasonality in the sample autocorrelation functions and spectral densities of each of the cardiac mortality, temperature, and particulate series, with indications of a period of 52 weeks. Analogous to how seasonality can be modeled in AR/ARMA/ARIMA models, the Base R **`VAR`** function also provides the option to model the seasonal component of a time series by use of seasonal indicator variables (also called "seasonal dummies") with the **`season`** option. Setting **`season = 52`** in this case will fit 51 indicator variables for each series to model a mean for each week using the data for the full dataset for variable selection. Below is code that will add the seasonal indicator variables (**`season= 52`**) to the **`VARselect`** call to re-identify the model order now that seasonal behavior is being considered.

```
VARselect(cardiac, lag.max = 10, season = 52, type = "both")
```

```
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     5      2      2      5

$criteria
              1        2        3        4        5        6        7        8        9       10
AIC(n)  11.1777  11.0008  10.9939  10.9560  10.9491  10.9673  10.9849  10.9964  11.0005  11.0143
HQ(n)   11.7351  11.5881  11.6111  11.6030  11.6261  11.6741  11.7216  11.7629  11.7969  11.8405
SC(n)   12.5981  12.4973  12.5666  12.6047  12.6740  12.7682  12.8619  12.9495  13.0297  13.1196
FPE(n)  71719.9  60120.7  59744.3  57552.0  57198.8  58289.0  59371.2  60108.7  60410.7  61309.4
```

When seasonality is accounted for, the AIC favors a model with $p = 5$, and the BIC (SC) favors the lag $p = 2$. Consistent with the previous analyses, a seasonal VAR model with p = 7 is also considered.

10.5.2.2 Use `VAR` to Fit the VAR Models to the Training Set

Each of three models above, seasonal models with lags two, seven, and nine, were fit by using the following code.

```
CMortVAR2 = VAR(cardiacTrain, type = "both", season = 52, p = 2)
CMortVAR9 = VAR(cardiacTrain, type = "both", season = 52, p = 9)
CMortVAR7 = VAR(cardiacTrain, type = "both", season = 52, p = 7)
```

Again, the initial check of the appropriateness of the models is conducted using the Ljung-Box test to check the residuals. The *p*-values from this test for each of these models are obtained using the code below.

```
ljung.wge(CMortVAR2$varresult$cmort$residuals,p=2)
ljung.wge(CMortVAR9$varresult$cmort$residuals,p=9)
ljung.wge(CMortVAR7$varresult$cmort$residuals,p=7)
```

The *p*-values using **`K=24`** (default) were .087, .054, and .011, respectively. Using **`K=48`**, the corresponding *p*-values were .265, .218, and .038. There are concerns about the whiteness of the noise, but we continue to assess the forecast performance of the three models.

10.5.2.3 Use `predict` to Forecast Data Values in the Test Set

The forecasts for the 52 weeks in the test ("hold out") set are calculated as usual:

```
preds2=predict(CMortVAR2,n.ahead=52)
preds9=predict(CMortVAR9,n.ahead=52)
preds7=predict(CMortVAR7,n.ahead=52)
```

Proceeding as before, the models are evaluated first by visualizing the forecasts versus the actual values in the training set and then computing the RMSEs for the last 52 weeks. The appropriate code is given below, and the corresponding output is shown in Figure 10.16 and Table 10.6, respectively.

```
plot(seq(1,508,1), cardiac[,"cmort"], type = "l", xlim = c(450,510), ylab =
"Cardiac Mortality", main = "52 Week Cardiac Mortality Forecast", xlab =
"Time")

t=1:508
points(t[457,508], preds2$fcst$cmort[,1], type="l", lwd=2, col = "red")
points(t[457,508], preds9$fcst$cmort[,1], type="l", lwd=2, col = "blue")
points(t[457,508], preds7$fcst$cmort[,1], type="l", lwd=2, col = "green")
```

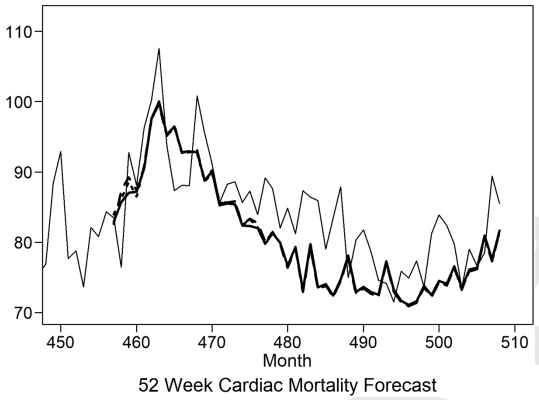


FIGURE 10.16 Visualization of the 52-week forecasts of the VAR(2), VAR(9) and VAR(7) models with trend and seasonal indicator variables, including actual values for comparison.

```
RMSE2 = sqrt(mean((cardiacTest[,"cmort"] - preds2$fcst$cmort[,1])^2))
RMSE9 = sqrt(mean((cardiacTest[,"cmort"] - preds9$fcst$cmort[,1])^2))
RMSE7 = sqrt(mean((cardiacTest[,"cmort"] - preds7$fcst$cmort[,1])^2))
```

TABLE 10.6 RMSEs for a Horizon of 52 Weeks for the VAR(2), VAR(9) and VAR(7) Models with Trend and Seasonal Indicator Variables.

MODEL (WITH TREND TERM AND SEASONAL INDICATORS)	*RMSE*
Var(2)	6.38
Var(9)	6.44
Var(7)	6.41

In this case, the VAR(2) produces a slightly smaller RMSE than the other two models, although based on the plots in Figure 10.15, the forecasts are very similar and tend to underestimate the actual cardiac mortalities (this fact will be revisited in the next chapter!). Overall, the VAR(7) without seasonality performs better with respect to BIC and RMSE, as summarized in Table 10.7 below.

TABLE 10.7 AIC, BIC and RMSEs for a Horizon of 52 Weeks for the VAR(7) with Trend and the VAR(2) with Trend and Seasonal Indicators.

MODEL	*AIC*	*BIC*	*RMSE*
VAR(7) with trend	11.15	11.73	5.44
VAR(2) with trend and seasonal indicators	11.00	12.49	6.38

The VAR(2) does have a slightly smaller AIC, which underlines the important fact that these criteria do not tell the analyst which model is "right" or even "better"; rather, these measures are tools the analyst should use along with other information or intuition in order to *make the final decision themselves*.

10.5.3 Forecasting the Future

Suppose that, based on the smaller RMSE of the VAR(7) model, an analyst decides to forecast the number of cardiac mortalities in the next 52 weeks. Using the entire dataset (**cardiac**), the forecasts and corresponding confidence intervals can be obtained using the code given below. Note that inside the **preds** object is an attribute called **$fcst**, which has a separate set of forecasts for each variable in the vector: **$cmort, $tempr** and **$part**. A summary of the data is provided by the function **head**, and the output is shown below.

```
CMortVAR7 = VAR(cardiac, type = "both", p = 7)
preds7=predict(CMortVAR7,n.ahead=52)
#cmort forecasts
head(preds7$fcst$cmort) #cardiac mortality forecasts
```

```
           fcst     lower      upper       CI
[1,]   86.80259  76.59671   97.00847  10.20588
[2,]   88.77409  77.97191   99.57627  10.80218
[3,]   87.48338  75.78664   99.18012  11.69674
[4,]   88.97024  76.86621  101.07427  12.10403
[5,]   89.56009  77.16757  101.95262  12.39253
[6,]   89.23073  76.49891  101.96254  12.73181
```

```
head(preds7$fcst$temp) # temperature forecasts
```

```
           fcst     lower      upper       CI
[1,]   71.36823  59.53554   83.20092  11.83269
[2,]   71.85304  59.82744   83.87865  12.02560
[3,]   68.87960  56.42767   81.33153  12.45193
[4,]   69.47003  56.89573   82.04433  12.57430
[5,]   67.97891  55.08042   80.87740  12.89849
[6,]   67.31557  54.02224   80.60890  13.29333
```

```
head(preds7$fcst$part) # particulates forecasts
```

```
           fcst     lower      upper       CI
[1,]   62.35471  42.21521   82.49422  20.13950
[2,]   64.26562  43.88740   84.64384  20.37822
[3,]   61.56504  40.60599   82.52409  20.95905
[4,]   62.94784  41.22299   84.67269  21.72485
[5,]   59.67438  36.89419   82.45457  22.78019
[6,]   58.65509  35.44117   81.86902  23.21392
```

Since cardiac mortality is the response of interest, we provide the following code to plot the forecast cardiac mortalities (**$cmort**) for the next 52 weeks, along with 95% prediction intervals.The output is shown in Figure 10.17.

```
plot(seq(1,508,1), cardiac[,"cmort"], type = "l",xlim = c(450,560), ylim =
c(50,112),xlab = "Time", ylab = "Cardiac Mortality", main = "52 Week Cardiac
Mortality Forecast From A VAR with p = 7")

t=1:560
points(t[509:560],preds7$fcst$cmort[,2], type = "l", lwd = 1, lty = 3)
points(t[509:560],preds7$fcst$cmort[,1] , type = "l", lwd = 1.5, lty = 1)
points(t[509:560],preds7$fcst$cmort[,3] , type = "l", lwd = 1, lty = 3)
```

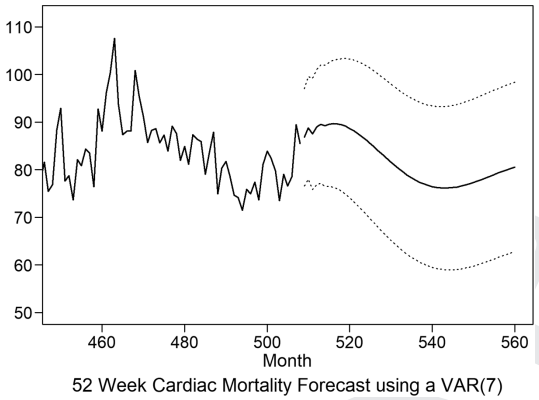


FIGURE 10.17 52-week Cardiac Mortality Forecast and 95% Prediction Limits from VAR(7) Model.

10.5.3.1 Short vs. Long Term Forecasts

The previous analyses have shown in this particular example that a VAR(7) model without seasonality (including trend) resulted in a superior RMSE based on forecasts of the last year of known data (1979). It is important to note that, similar to the behavior exhibited by an AR model with trend, this VAR(7) model with trend will also gravitate toward the trend line over time. This behavior indeed occurs in the forecasts for the next four years (1980–1984), which are shown in Figure 10.18(a). In contrast, the VAR(2) model with seasonality and trend will continue to perpetuate the periodic behavior in the forecasts into the future, as seen in Figure 10.18(b).

```
# Forecasting next 4 years with VAR(7) Model with Trend

CMortVAR7 = VAR(cardiac, type = "both", p = 7)
preds7=predict(CMortVAR7,n.ahead=208)

plot(seq(1,508,1), cardiac[,"cmort"], type = "l",xlim = c(450,716), ylim =
c(50,112),xlab = "Time", ylab = "Cardiac Mortality", main = "52 Week Cardiac
Mortality Forecast From A VAR with p = 7")
t=1:716

points(t[509:716],preds7$fcst$cmort[,2], type = "l", lwd = 1, lty = 3)
points(t[509:716],preds7$fcst$cmort[,1] , type = "l", lwd = 1.5, lty = 1)
points(t[509:716],preds7$fcst$cmort[,3] , type = "l", lwd = 1, lty = 3)

# Forecasting next 4 years with VAR(2) Seasonal and Trend Model
CMortVAR2 = VAR(cardiac,season = 52, type = "both", p = 2)
preds2=predict(CMortVAR2,n.ahead=208)

plot(t[1:508], cardiac[,"cmort"], type = "l",xlim = c(450,716), ylim =
c(50,112),xlab = "Time", ylab = "Cardiac Mortality", main = "52 Week Cardiac
Mortality Forecast From a VAR(2) and Seasonality and Trend")
points(t[509:716], preds2$fcst$cmort[,2],type="l",lwd=1,lty=4)
points(t[509:716,preds2$fcst$cmort[,1],type="l",lwd =2,lty=1)
points(t[509:716],preds2$fcst$cmort[,3],type = "l",lwd = 1,lty = 3)
```

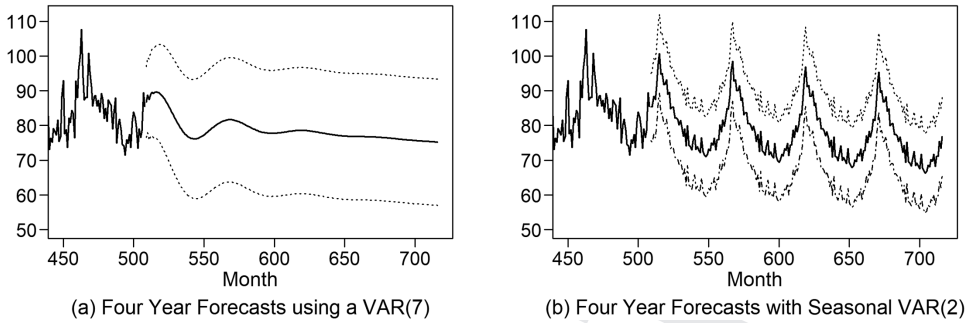


FIGURE 10.18 (a) Four-year forecasts for cardiac mortality from VAR(7) with trend model; (b) Four-year forecasts for cardiac mortality from VAR(2) model with seasonality and trend.

While there seems to be (mild) evidence that the VAR(7) model with no seasonality (trend included) provides superior forecasts for a time horizon of 52, it is not clear, especially given the strong seasonality in the cardiac mortality series, that forecasts from this model will be superior for longer horizons. (Check it out.) Recall also that white noise is questionable for the residuals in the seasonal model. Again, no models are perfect, and the seasonal forecasts are quite good.

10.6 CONCLUSION

This chapter has explored two types of multivariate models and has demonstrated how selecting useful predictor variables and assessing relevant lagged relationships results in better models and improved forecasts. Analogous to how (in the standard statistical setting with independent observations) a multiple regression model can add valuable information to a simple linear regression model, the same holds true for a multivariate time series model. The key for the analyst is to identify related variables that have a lagged relationship with the dependent variable, and then identify the lag. In certain applications such as finance and economics, many types of data have historically been collected and analyzed so that commonly known "leading" indicators are available. For example, when predicting the Gross Domestic Product (GDP), economists often use leading economic indicators which have previously been observed such as unemployment rate, income and wages, corporate profits, etc. This previous information can simplify the process, but in other more obscure or less common applications, finding such related variables may take time, effort, and creativity (and a little luck!).

APPENDIX 10A

No new ***tswge*** functions were introduced in this chapter. However, the analyses in the chapter depended heavily on several Base R functions along with functions from CRAN packages ***vars*** and ***dplyr***. We discuss these briefuly.

Base R functions: `arima` and `lm`

(a) `arima(x, order = c(0L, 0L, 0L),xreg = NULL`) is a multi-purpose function that has many more parameters than listed here. Given the parameters listed here:

(1) If **`xref=NULL`** then **`arima`** fits an ARIMA$(p,d.q)$ model to the data in **x** using ML estimates. Note that the coefficients of the moving average parameters will have opposite signs to those used in this book.
(2) If **`xreg`** specifies variables, then the function computes the ML estimates of a multiple regression with correlated errors in which the parameter order specifies the order of the model fit to the residuals from the MLR fit.

(b) **`lm(x~x1+x2+x3)`** is a function that performs a standard multiple linear regression assuming uncorrelated errors. In the above, **y** is the dependent variable and the **`x1, x2,`** and **`x3`** are independent variables.

Example 10.1:

```
x=gen.arma.wge(n=200,phi=c(1.6,-.9),theta=.8,sn=10)
arima(x,order=c(2,0,1))
```

The output is

```
Coefficients:
             ar1      ar2      ma1  intercept
          1.5657  -0.8025  -0.8932    -0.0662
s.e.      0.0436   0.0419   0.0362     0.0325

sigma^2 estimated as 0.9509: log likelihood = -279.86, aic = 569.73
```

Note that the output from the ***tswge*** command

```
est.arma.wge(x,p=2,q=1)
```

includes:

```
$phi
[1] 1.5657450 -0.8025325
$theta
[1] 0.8932299
```

Note: The ***tswge*** command **`est.arma.wge`** calls the function **`arima`** and changes the sign of the moving average parameters in the output.

Example
Example 10.1 illustrates the use of **arima** for the purpose of multiple regression with correlated errors. The following commands were used in the first multiple regression performed in Example 10.1:

```
mlrfit=lm(sales~ad_tv+ad_online+discount)
arima(sales,order=c(7,0,0),xreg=cbind(ad_tv,ad_online,discount))
```

The command **lm(sales~ad_tv+ad_online+discount)** performs a standard multiple regression with **sales** as the dependent variable and **ad_tv, ad_online,** and **discount** as the independent variables. In Example 10.1 AIC selects an AR(7) model for the residuals.

The command **arima(sales,order=c(7,0,0),xreg=cbind(ad_tv,ad_online, discount))**
produces the ML estimates of the parameters of a multiple regression with **sales** as the dependent variable, **ad_tv, ad_online**, and **discount** as the independent variables, and where the variance-covariance matrix of the residuals is based on the AR(7) model fit to the residuals from the **lm** function call.

CRAN PACKAGE *DPLYR*

dplyr is an R package designed for data manipulation. It was used in the taxicab data example in Chapter 1. In this chapter, we only use the function **lag**.

Example:

```
x=c(1,3,2,4,3,1,5,3)
x1=dplyr::lag(x)
x1
[1] NA 1 3 2 4 3 1 5
```

That is, **x1[1]=NA, x1[2]=x[1], …, x1[n]=x[n-1]**

CRAN PACKAGE *VARS*

vars is an R package that has functions that perform VAR analysis.

(a) **VARselect(y,lag.max,type,season,exogen)** provides AIC-type measures for identifying the order of the VAR(p) fit.

y in the multivariate data frame containing the multivariate time series data
lag.max is the maximum order of p allowed (default=10)
season is either **NULL** (default) if the data are not seasonal. Otherwise enter the frequency. That is, for monthly data **season=12**.
type specifies whether the model contains a constant, trend, both or none using type=**'const', 'trend', 'both',** or **'neither',** respectively.
exogen is a data frame containing any exogenous variables (default=**NULL)**

(b) **VAR(y,p,type,season,exogen)** fits an AR(p) model to the data in **y**. Other options are available. See documentation. The parameters **type, season**, and **exogen** are the same as in **VARselect**.

(c) **predict(object, n.ahead,ci,dumvar)** uses the VAR object produced by **VAR** and uses it to produce forecasts up to **n.head** steps ahead. Prediction limits are specified by **ci** (default=.95) and **dumvar** provides for dummy variables (default=**NULL**). See Sections 10.3 and 10.5.

TSWGE DATASETS INTRODUCED IN THIS CHAPTER

1. **Bsales** – a data frame containing 100 weeks of data on the variables:
   sales (in thousands of dollars)
   ad_tv – cost of TV advertising (in thousands of dollars)
   ad_online – cost of online advertising (in thousands of dollars)
   discount – discount (in percent)

2. **cardiac** – a multivariate ***ts*** object that consists of weekly cardiac mortality, temperatures, and pollution measures for the years 1970–1978 and the first 40 weeks of 1979 in Los Angeles, California. The data were obtained from the astsa package. The variables are as follows:

   cmort – average weekly cardiac mortalities
   tempr – average weekly temperatures
   part – average weekly number of particulates in the air

APPENDIX 10B

RELATIONSHIP BETWEEN MLR WITH CORRELATED ERRORS AND VAR

In Section 10.4, it was mentioned that, although not immediately apparent, there is a surprising relationship between MLR with correlated errors and VAR models. To see this, reconsider the simulated example in Example 10.2 (data are given below). Run the following code, and observe that the coefficients for the corresponding variables between the two methods match perfectly!

Three Important Points Should Be Considered:

(1) A VAR model is an MLR model, but not all MLR models are VAR models.
(2) For coefficients to exactly match among the variables for the two methods, the lag order must be the same for each of the variables for both the MLR and for the VAR.
(3) If a trend and intercept term is included for MLR, the parameter **type='both'** must be specified in VAR. If only the intercept term is included for MLR, the function **type='const'** must be specified in VAR.

```
x1 = c(-1.03,0.11,-0.18,0.20,-0.99,-1.63,1.07,2.26,-0.49,-1.54,0.45,0.92,
-0.05,-1.18,0.90,1.17,0.31,1.19,0.27,-0.09,0.23,-1.91,0.46,3.61,-0.03)

x2 = c(-0.82,0.54,1.13,-0.24,-0.77,0.22,0.46,-0.03,-0.59,0.45,0.59,0.15,
0.60,0.13,-0.04,0.12,-0.96,0.23,1.81,-0.01,-0.95,-0.55,-0.15,0.71,0.90)
```

```
XDF = data.frame(x1, x2)

x1Train = x1[1:20]
x2Train = x2[1:20]
XTrainDF = data.frame(x1Train, x2Train)

# MLR Analysis Code
# MLR Lag 2
#Manually Lag x1 and x2 at lags 1 and 2
XTrainDF$X1_l1 = dplyr::lag(XTrainDF$x1,1)
XTrainDF$X1_l2 = dplyr::lag(XTrainDF$x1,2)

XTrainDF$X2_l1 = dplyr::lag(XTrainDF$x2,1)
XTrainDF$X2_l2 = dplyr::lag(XTrainDF$x2,2)

#Show NAs
XTrainDF
```

	x1Train	x2Train	X1_l1	X1_l2	X2_l1	X2_l2
1	-1.03	-0.82	NA	NA	NA	NA
2	0.11	0.54	-1.03	NA	-0.82	NA
3	-0.18	1.13	0.11	-1.03	0.54	-0.82
4	0.20	-0.24	-0.18	0.11	1.13	0.54
5	-0.99	-0.77	0.20	-0.18	-0.24	1.13
6	-1.63	0.22	-0.99	0.20	-0.77	-0.24
7	1.07	0.46	-1.63	-0.99	0.22	-0.77
8	2.26	-0.03	1.07	-1.63	0.46	0.22
9	-0.49	-0.59	2.26	1.07	-0.03	0.46
10	-1.54	0.45	-0.49	2.26	-0.59	-0.03
11	0.45	0.59	-1.54	-0.49	0.45	-0.59
12	0.92	0.15	0.45	-1.54	0.59	0.45
13	-0.05	0.60	0.92	0.45	0.15	0.59
14	-1.18	0.13	-0.05	0.92	0.60	0.15
15	0.90	-0.04	-1.18	-0.05	0.13	0.60
16	1.17	0.12	0.90	-1.18	-0.04	0.13
17	0.31	-0.96	1.17	0.90	0.12	-0.04
18	1.19	0.23	0.31	1.17	-0.96	0.12
19	0.27	1.81	1.19	0.31	0.23	-0.96
20	-0.09	-0.01	0.27	1.19	1.81	0.23

```
# Omit NAs for lm()
XTrainDF2 = XTrainDF[3:20,]

XTrainDF2
```

	x1Train	x2Train	X1_l1	X1_l2	X2_l1	X2_l2
3	-0.18	1.13	0.11	-1.03	0.54	-0.82
4	0.20	-0.24	-0.18	0.11	1.13	0.54
5	-0.99	-0.77	0.20	-0.18	-0.24	1.13
6	-1.63	0.22	-0.99	0.20	-0.77	-0.24
7	1.07	0.46	-1.63	-0.99	0.22	-0.77
8	2.26	-0.03	1.07	-1.63	0.46	0.22
9	-0.49	-0.59	2.26	1.07	-0.03	0.46
10	-1.54	0.45	-0.49	2.26	-0.59	-0.03
11	0.45	0.59	-1.54	-0.49	0.45	-0.59

```
12       0.92      0.15     0.45     -1.54     0.59     0.45
13      -0.05      0.60     0.92      0.45     0.15     0.59
14      -1.18      0.13    -0.05      0.92     0.60     0.15
15       0.90     -0.04    -1.18     -0.05     0.13     0.60
16       1.17      0.12     0.90     -1.18    -0.04     0.13
17       0.31     -0.96     1.17      0.90     0.12    -0.04
18       1.19      0.23     0.31      1.17    -0.96     0.12
19       0.27      1.81     1.19      0.31     0.23    -0.96
20      -0.09     -0.01     0.27      1.19     1.81     0.23
```

```
#fit the MLR
fit = lm(x1Train ~ X1_l1 + X1_l2 + X2_l1 + X2_l2, data = XTrainDF2)
summary(fit)
```

```
Call:
lm(formula = x1Train ~ X1_l1 + X1_l2 + X2_l1 + X2_l2, data = XTrainDF2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.42514 -0.35651 0.03649 0.40597 1.76421

Coefficients:
             Estimate   Std. Error t value    Pr(>|t|)
(Intercept)    0.14904   0.23394     0.637     0.5351
X1_l1          0.18676   0.22461     0.831     0.4207
X1_l2         -0.58499   0.21574    -2.712     0.0178 *
X2_l1          0.09124   0.35172     0.259     0.7994
X2_l2         -0.07601   0.41576    -0.183     0.8578
---

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9243 on 13 degrees of freedom
Multiple R-squared: 0.397, Adjusted R-squared: 0.2115
F-statistic: 2.14 on 4 and 13 DF, p-value: 0.1336
```

```
# Compare with VAR Analysis
# Create ts objects
X_ts = ts(XDF)
XTrain_ts = ts(XDF[1:20,])
XTrain_ts #compare with XTrainDF
```

```
Time Series:
Start = 1
End = 20
Frequency = 1
       x1    x2
1   -1.03 -0.82
2    0.11  0.54
3   -0.18  1.13
4    0.20 -0.24
5   -0.99 -0.77
6   -1.63  0.22
7    1.07  0.46
8    2.26 -0.03
```

```
9   -0.49  -0.59
10  -1.54   0.45
11   0.45   0.59
12   0.92   0.15
13  -0.05   0.60
14  -1.18   0.13
15   0.90  -0.04
16   1.17   0.12
17   0.31  -0.96
18   1.19   0.23
19   0.27   1.81
20  -0.09  -0.01
```

```
# Fit the model with only the constant
VARfit = VAR(XTrain_ts,p=2, type='const')
VARfit$varresult$x1 #compare with MLR fit at lag 2 above

Call:
lm(formula = y ~ -1 + ., data = datamat)

Coefficients:
x1.l1      x2.l1     x1.l2      x2.l2     const
0.18676  0.09124 -0.58499   -0.07601   0.14904
```

Compare these coefficients with the results of the MLR model fit!

PROBLEMS

1. In this chapter multiple linear regression with correlated errors was used to model with **Bsales** dataset. Why does it not make sense to use VAR to model the sales in this analysis?
2. Consider the model:

   $$Sales_t = \beta_0 + \beta_1 t + \beta_2 ad\_tv_{t-1} + \beta_3 ad\_online_{t-1} + \beta_4 discount_t + Z_t$$

   (a) Use multiple linear regression (MLR) assuming independent errors (assume Z_t is iid $N(0,\hat{\sigma}_Z^2)$) to find the RMSE for last 10 sales in the dataset.
   (b) Now use multiple linear regression with correlated errors to find the RMSE for the last 10 sales.
   (c) Is there a difference? What is your conclusion?
   (d) Can you use the model in (b) to forecast sales for week 101? For week 102? Explain.
3. Use MLR with correlated errors to model the melanoma count given the sunspot count (use dataset **melanoma2.0**). Find the RMSE for the last 8 years as was done in the text for VAR. Remember to use forecast values for sunspots where appropriate. Does one model seem more useful?
4. Variable Selection: Does the addition of temperature or particulates variables make a difference?
   (a) Fit a VAR(p) model with cardiac mortality and particulates but without temperature. Find and record the AIC of the model with the lowest AIC and find and record the RMSE for the last 52 weeks.

(b) Fit a VAR(p) model with cardiac mortality, particulates and temperature. Find and record the AIC of the model with the lowest AIC and find and record the RMSE for the last 52 weeks.
(c) Does your evidence suggest that temperature is an important variable / feature?

5. Repeat the analysis in question 4 for the particulate variable.
6. Load the ***astsa*** package in CRAN and load the data **`lap`**. This is the dataset contained in the dataframe **`cardiac`** but with many more possible explanatory variables. Create a model or models using additional variables other than temperature and particulates. Can you improve on the model/s using only these two variables? Explain. Are there any relationships between cardiac mortality and other of the variables in the **`lap`** dataset? Provide a thoughtful and well explained (with plots) response.
7. *Challenge Problem*: Show that the equation from the VAR(2) model for cardiac mortality with temperature and particulates in the vector with a constant and a trend is equivalent to a multiple linear regression with the same lagged variables. Appendix 10B will be helpful in investigating this issue.