Published on *STAT 510* (https://onlinecourses.science.psu.edu/stat510)

# Lesson 11: Vector Autoregressive Models/ ARCH Models

**Assignments:**

- Read Section 5.4 of your text, with focus on Examples 5.4 and 5.5.
- Read Examples 5.10 and 5.11 on pages 304-309 of your text.
- Read through the Lesson 11 online notes that follow.
- Complete Lesson 11 Assignment.

**Overview:**

This week we'll look at two topics - models for periods of volatile variance (ARCH models) and AR models for multivariate time series.

**Learning Objectives**:

After successfully completing this lesson, you should be able to:

- Model the variance of a time series
- Identify and interpret ARCH models
- Simultaneously model multiple variables in terms of past lags of themselves and one another

# 11.1: ARCH/GARCH Models

An ARCH (autoregressive conditionally heteroscedastic) model is a model for the variance of a time series. ARCH models are used to describe a changing, possibly volatile variance. Although an ARCH model could possibly be used to describe a gradually increasing variance over time, most often it is used in situations in which there may be short periods of increased variation. (Gradually increasing variance connected to a gradually increasing mean level might be better handled by transforming the variable.)

ARCH models were created in the context of econometric and finance problems having to do with the amount that investments or stocks increase (or decrease) per time period, so there's a tendency to describe them as models for that type of variable. For that reason, the authors of our text suggest that the variable of interest in these problems might either be $y_t = (x_t - x_{t-1})/x_{t-1}$, the proportion gained or lost since the last time, or $\log(x_t/x_{t-1}) = \log(x_t) - \log(x_{t-1})$, the logarithm of the ratio of this time's value to last time's value. It's not necessary that one of these be the primary variable of interest. An ARCH model could be used for any series that has periods of increased or decreased variance. This might, for example, be a property of residuals after an ARIMA model has been fit to the data.

**The ARCH(1) Variance Model**

Suppose that we are modeling the variance of a series $y_t$. The ARCH(1) model for the variance of model $y_t$ is that conditional on $y_{t-1}$, the variance at time t is

$$Var(y_t|y_{t-1}) = \sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 \qquad (1)$$

We impose the constraints $\alpha_0 \geq 0$ and $\alpha_1 \geq 0$ to avoid negative variance.

Note that the variance at time *t* is connected to the value of the series at time *t* – *1*. A relatively large value of $y^2_{t-1}$ gives a relatively large value of the variance at time *t*. This means that the value of $y_t$ is less predictable at time *t* −*1* than at times after a relatively small value of $y^2_{t-1}$.

If we assume that the series has mean = 0 (this can always be done by centering), the ARCH model could be written as

(2) $$y_t = \sigma_t \epsilon_t, \text{ with } \sigma_t = \sqrt{\alpha_0 + \alpha_1 y^2_{t-1}}, \text{ and } \epsilon_t \sim iid(0,1)$$

For inference (and maximum likelihood estimation) we would also assume that the $\varepsilon_t$ are normally distributed.

**Possibly Useful Results**

Two potentially useful properties of the useful theoretical property of the ARCH(1) model as written in equation line (2) above are the following:

- $y^2_t (y_t \text{ squared }) \text{ has the AR}(1) \text{ model } y^2_t = \alpha_0 + \alpha_1 y^2_{t-1} + error.$

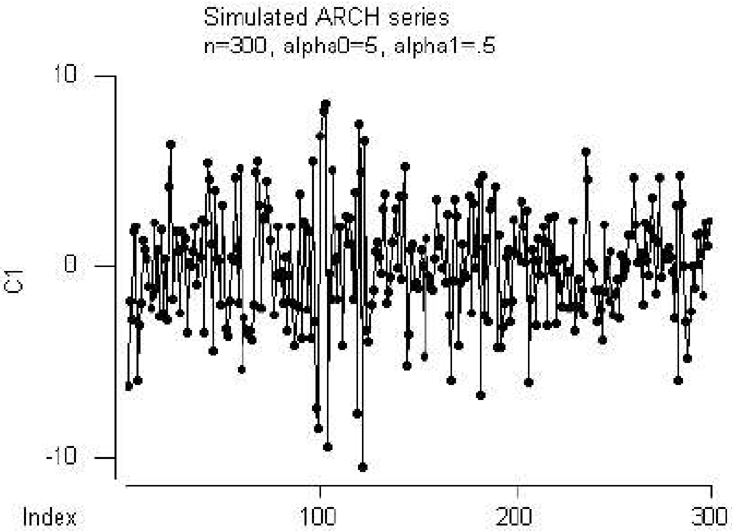
  This model will be causal, meaning it can be converted to a legitimate infinite order MA only when $\alpha^2_1 < \frac{1}{3}$

- $y_t$ is white noise when $0 \leq \alpha_1 \leq 1$.

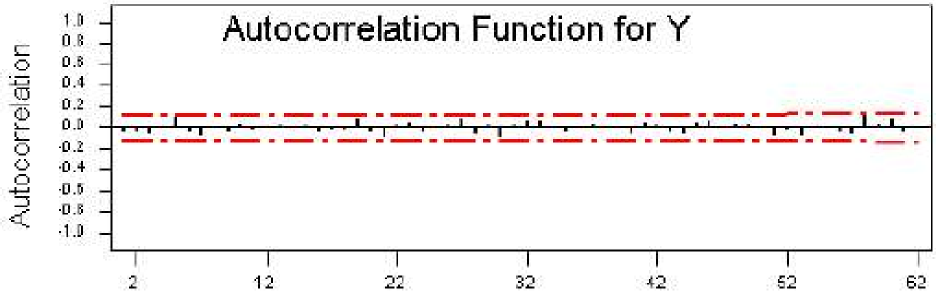
**Example**: The following plot is a time series plot of a simulated series (n = 300) for the ARCH model

$$Var(y_t|y_{t-1}) = \sigma^2_t = 5 + 0.5y^2_{t-1}.$$

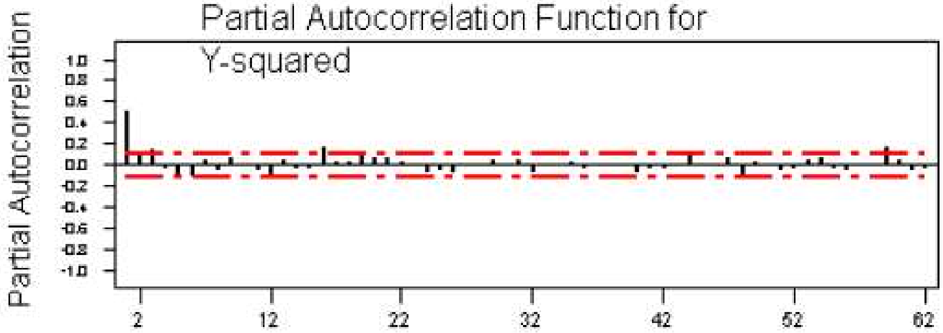
Note that there are a few periods of increased variation, most notably around *t* = 100.



The ACF of this series just plotted follows. Note that no correlations are significant, so the series looks to be white noise.



The PACF (following) of the squared values has a single spike at lag 1 suggesting an AR(1) model for the squared series.



### Generalizations

An ARCH(m) process is one for which the variance at time *t* is conditional on observations at the previous *m* times, and the relationship is

$$Var(y_t|y_{t-1},\ldots,y_{t-m}) = \sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \cdots + \alpha_m y_{t-m}^2.$$

With certain constraints imposed on the coefficients, the $y_t$ series squared will theoretically be AR(m).

A GARCH (generalized autoregressive conditionally heteroscedastic) model uses values of the past squared observations and past variances to model the variance at time *t*. As an example, a GARCH(1,1) is

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

In the GARCH notation, the first subscript refers to the order of the $y^2$ terms on the right side, and the second subscript refers to the order of the $\sigma^2$ terms.

### Identifying an ARCH/GARCH Model in Practice

The best identification tool may be a time series plot of the series. It's usually easy to spot periods of increased variation sprinkled through the series. It can be fruitful to look at the ACF and PACF of both $y_t$ and $y_t^2$. For instance, if $y_t$ appears to be white noise and $y_t^2$ appears to be AR(1), then an ARCH(1) model for the variance is suggested. If the PACF of the $y_t^2$ suggests AR(m), then ARCH(m) may work. GARCH models may be suggested by an ARMA type look to the ACF and PACF of $y_t^2$. In practice, things won't always fall into place as nicely as they did for the simulated example in this lesson. You might have to experiment with various ARCH and GARCH structures after spotting the need in the time series plot of the series.

In the book, read Example 5.4 (p. 283-middle of p. 285), and Example 5.5 (p. 286-p.287). R code for similar examples will be given in the homework for this week.

# 11.2: Vector Autoregressive models VAR(p) models

VAR models (vector autoregressive models) are used for multivariate time series. The structure is that each variable is a linear function of past lags of itself and past lags of the other variables.

As an example suppose that we measure three different time series variables, denoted by $x_{t,1}$, $x_{t,2}$, and $x_{t,3}$.

The vector autoregressive model of order 1, denoted as VAR(1), is as follows:

$$x_{t,1} = \alpha_1 + \phi_{11}x_{t-1,1} + \phi_{12}x_{t-1,2} + \phi_{13}x_{t-1,3} + w_{t,1}$$

$$x_{t,2} = \alpha_2 + \phi_{21}x_{t-1,1} + \phi_{22}x_{t-1,2} + \phi_{23}x_{t-1,3} + w_{t,2}$$

$$x_{t,3} = \alpha_3 + \phi_{31}x_{t-1,1} + \phi_{32}x_{t-1,2} + \phi_{33}x_{t-1,3} + w_{t,3}$$

Each variable is a linear function of the lag 1 values for all variables in the set.

In a VAR(2) model, the lag 2 values for all variables are added to the right sides of the equations, In the case of three x-variables (or time series) there would be six predictors on the right side of each equation, three lag 1 terms and three lag 2 terms.

In general, for a VAR(p) model, the first p lags of each variable in the system would be used as regression predictors for each variable.

VAR models are a specific case of more general VARMA models. VARMA models for multivariate time series include the VAR structure above along with moving average terms for each variable. More generally yet, these are special cases of ARMAX models that allow for the addition of other predictors that are outside the multivariate set of principal interest.

Here, as in Section 5.8 of the text, we'll focus on VAR models.

On page 304, the authors fit the model of the form

$$\mathbf{x}_t = \Gamma\mathbf{u}_t + \phi\mathbf{x}_{t-1} + \mathbf{w}_t$$

where $\mathbf{u}_t = (1, t)'$ includes terms to simultaneously fit the constant and trend. It arose from macroeconomic data where large changes in the data permanently affect the level of the series.

There is a not so subtle difference here from previous lessons in that we now are fitting a model to data that need not be stationary. In previous versions of the text, the authors separately de-trended each series using a linear regression with t, the index of time, as the predictor variable. The de-trended values for each of the three series are the residuals from this linear regression on *t*. The de-trending is useful conceptually because it takes away the common steering force that time may have on each series and created stationarity as we have seen in past lessons. This approach results in similar coefficients, though slightly different as we are now simultaneously fitting the intercept and trend together in a multivariate OLS model.

The R vars library authored by Bernhard Pfaff has the capability to fit this model with trend. Let's look at 2 examples: a difference-stationary model and a trend-stationary model.

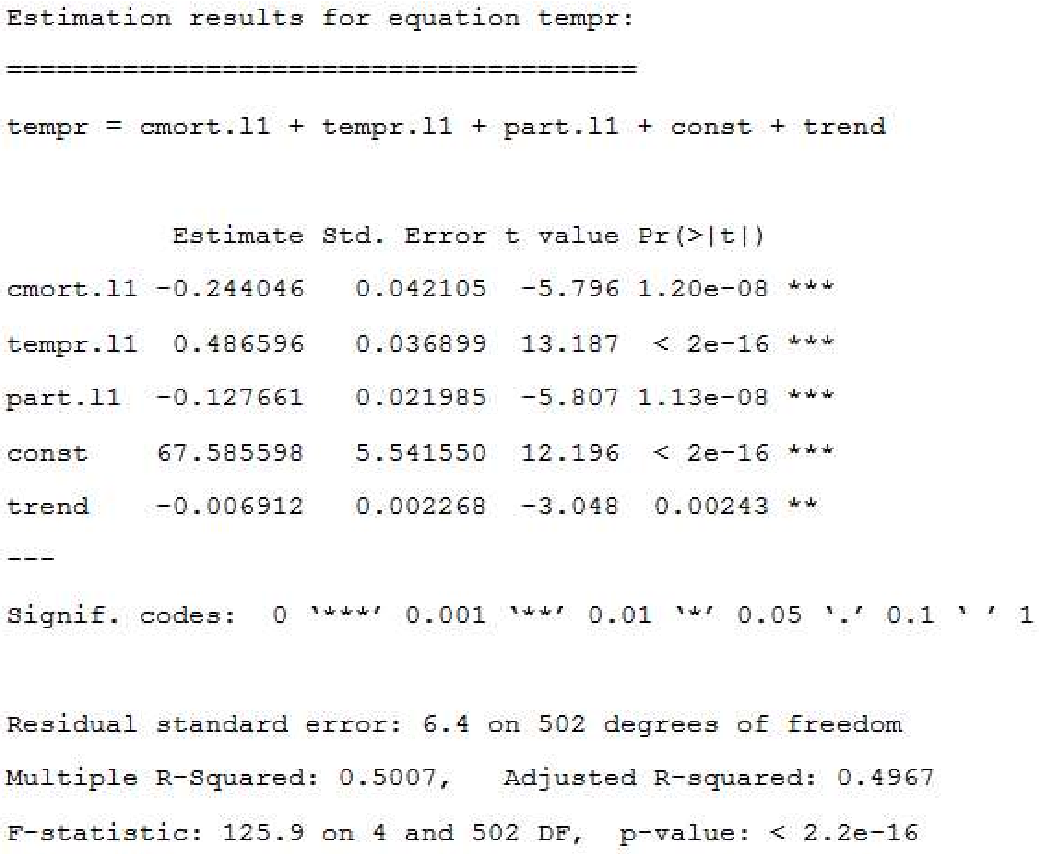
## Difference-Stationary Model

Example 5.10 from the text is a difference-stationary model in that first differences are stationary. Let's examine the code and example from the text by fitting the model above:

```
install.packages("vars") #If not already installed
install.packages("astsa") #If not already installed
library(vars)
library(astsa)
x = cbind(cmort, tempr, part)
plot.ts(x , main = "", xlab = "")
summary(VAR(x, p=1, type="both"))
```

- The first two commands load the necessary commands from the vars library and the necessary data from our text's library.
- The cbind command creates a vector of response variables (a necessary step for multivariate responses).
- The VAR command does estimation of AR models using ordinary least squares while simultaneously fitting the trend, intercept, and ARIMA model. The p = 1 argument requests an AR(1) structure and "both" fits constant and trend. With the vector of responses, it's actually a VAR(1).

Following is the output from the VAR command for the variable tempr (the text provides the output for cmort):



```
Estimation results for equation tempr:
=========================================
tempr = cmort.l1 + tempr.l1 + part.l1 + const + trend

          Estimate Std. Error t value Pr(>|t|)
cmort.l1 -0.244046   0.042105  -5.796 1.20e-08 ***
tempr.l1  0.486596   0.036899  13.187  < 2e-16 ***
part.l1  -0.127661   0.021985  -5.807 1.13e-08 ***
const    67.585598   5.541550  12.196  < 2e-16 ***
trend    -0.006912   0.002268  -3.048  0.00243 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 6.4 on 502 degrees of freedom
Multiple R-Squared: 0.5007,   Adjusted R-squared: 0.4967
F-statistic: 125.9 on 4 and 502 DF,  p-value: < 2.2e-16
```

The coefficients for a variable are listed in the Estimate column. The .l1 attached to each variable name indicates that they are lag 1 variables.

Using notation T = temperature, t=time (collected weekly), M = mortality rate, and P = pollution, the equation for temperature is

$$\hat{T}_t = 67.586 - .007t - 0.244M_{t-1} + 0.487T_{t-1} - 0.128P_{t-1}$$

The equation for mortality rate is

$$\hat{M}_t = 73.227 - 0.014t + 0.465M_{t-1} - 0.361T_{t-1} + 0.099P_{t-1}$$

The equation for pollution is

$$\hat{P}_t = 67.464 - .005t - 0.125M_{t-1} - 0.477T_{t-1} + 0.581P_{t-1}.$$

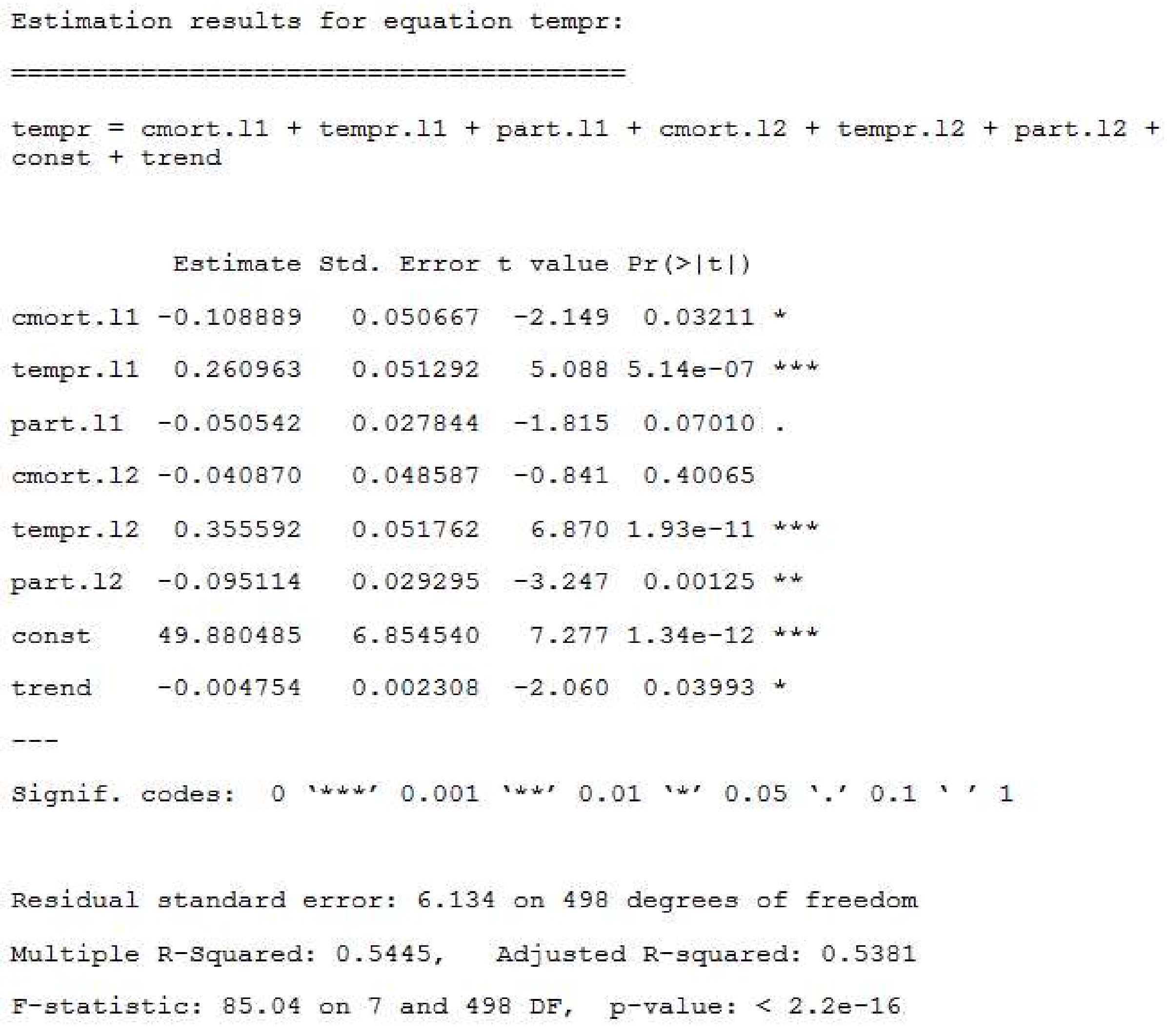
The covariance matrix of the residuals from the VAR(1) for the three variables is printed below the estimation results. The variances are down the diagonal and could possibly be used to compare this model to higher order VARs. The determinant of that matrix is used in the calculation of the BIC statistic that can be used to compare the fit of the model to the fit of other models (see formulas 5.89 and 5.90 of the text).

For further references on this technique see Analysis of integrated and co-integrated time series with R by Pfaff and also Campbell and Perron [1991].

In Example 5.11 on page 307, the authors give results for a VAR(2) model for the mortality rate data. In R, you may fit the VAR(2) model with the command

```
summary(VAR(x, p=2, type="both"))
```

The output, as displayed by the VAR command is as follows:



```
Estimation results for equation tempr:

=========================================

tempr = cmort.l1 + tempr.l1 + part.l1 + cmort.l2 + tempr.l2 + part.l2 +
const + trend


          Estimate Std. Error t value Pr(>|t|)

cmort.l1 -0.108889   0.050667  -2.149  0.03211 *

tempr.l1  0.260963   0.051292   5.088 5.14e-07 ***

part.l1  -0.050542   0.027844  -1.815  0.07010 .

cmort.l2 -0.040870   0.048587  -0.841  0.40065

tempr.l2  0.355592   0.051762   6.870 1.93e-11 ***

part.l2  -0.095114   0.029295  -3.247  0.00125 **

const    49.880485   6.854540   7.277 1.34e-12 ***

trend    -0.004754   0.002308  -2.060  0.03993 *

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 6.134 on 498 degrees of freedom

Multiple R-Squared: 0.5445,   Adjusted R-squared: 0.5381

F-statistic: 85.04 on 7 and 498 DF,  p-value: < 2.2e-16
```

Again, the coefficients for a particular variable are listed in the Estimate column. As an example, the estimated equation for temperature is

$$\hat{T}_t = 49.88 - .005t - 0.109M_{t-1} + 0.261T_{t-1} - 0.051P_{t-1} - 0.041M_{t-2} + 0.356T_{t-2} - 0.095P_{t-2}$$

We will discuss information criterion statistics to compare VAR models of different orders in the homework.

Residuals are also available for analysis. For example, if we assign the VAR command to an object titled fitvar2 in our program,

```
fitvar2 = VAR(x, p=2, type="both")
```

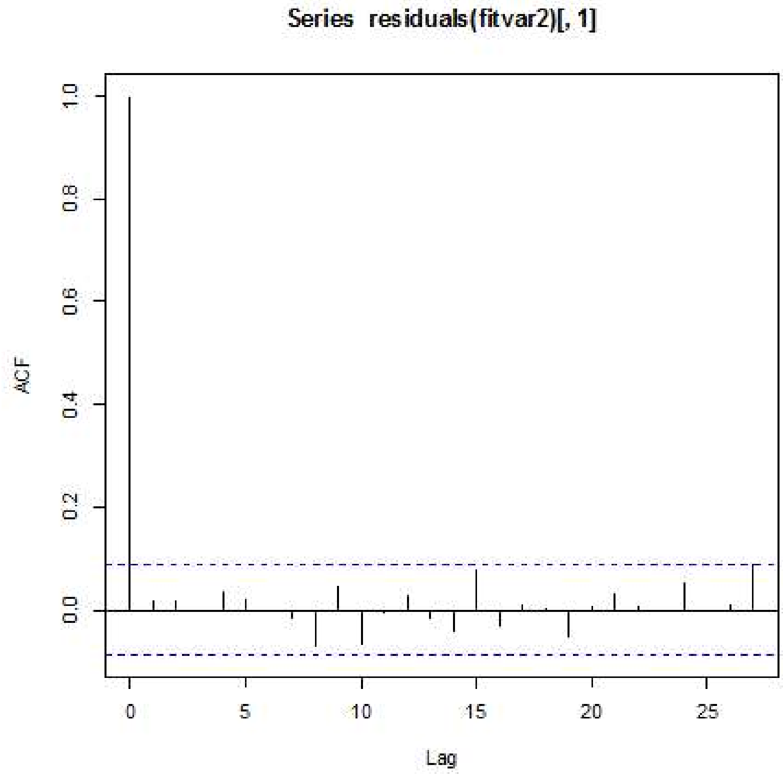
then we have access to the matrix residuals(fitvar2). This matrix will have three columns, one column of residuals for each variable.

For example, we might use

```
acf(residuals(fitvar2)[,1])
```

to see the ACF of the residuals for mortality rate after fitting the VAR(2) model.

Following is the ACF that resulted from the command just described. It looks good for a residual ACF. (The big spike at the beginning is the unimportant lag 0 correlation.)



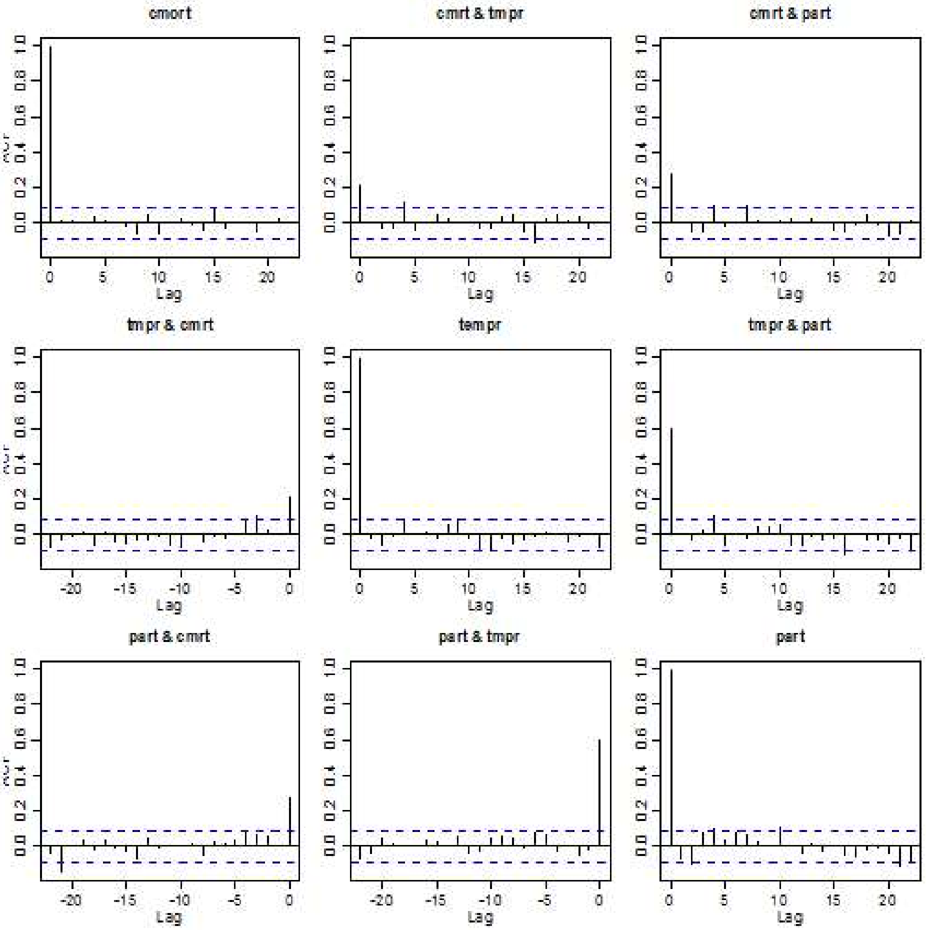
The following two commands will create ACFs for the residuals for the other two variables.

```
acf(residuals(fitvar2)[,2])
```

```
acf(residuals(fitvar2)[,3])
```

They also resemble white noise.

We may also examine these plots in the cross-correlation matrix provided by `acf(residuals(fitvar2))`:



The plots along the diagonal are the individual ACFs for each model's residuals that we just discussed above. In addition, we now see the cross-correlation plots of each set of residuals. Ideally, these would also resemble white noise, however we do see remaining cross-correlations, especially between temperature and pollution. As our authors note, this model does not adequately capture the complete association between these variables in time.

## Trend-Stationary Model

Lets explore an example where the original data are stationary and examine the VAR code by fitting the model above with both a constant and trend. Using R, we simulated *n* = 500 sample values using the VAR(2) model

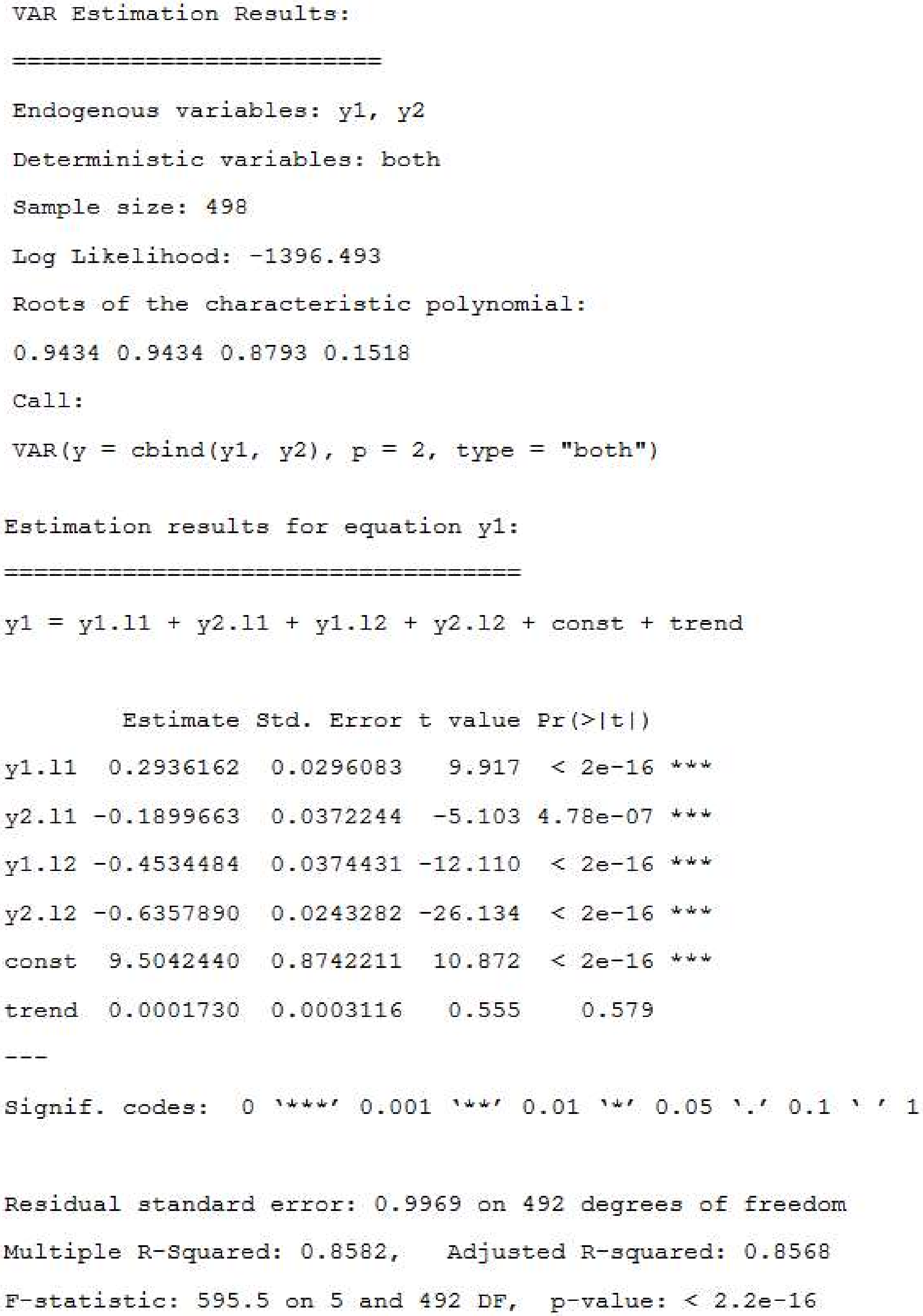
$$y_{t,1} = 10 + .25y_{t-1,1} - .20y_{t-1,2} - .40y_{t-2,1} - .65y_{t-2,2}$$

$$y_{t,2} = 20 + .60y_{t-1,1} - .45y_{t-1,2} + .50y_{t-2,1} + .35y_{t-2,2}$$

Using the VAR command explained above:

```
y1=scan("var2daty1.dat")
y2=scan("var2daty2.dat")
summary(VAR(cbind(y1,y2), p=2, type="both"))
```

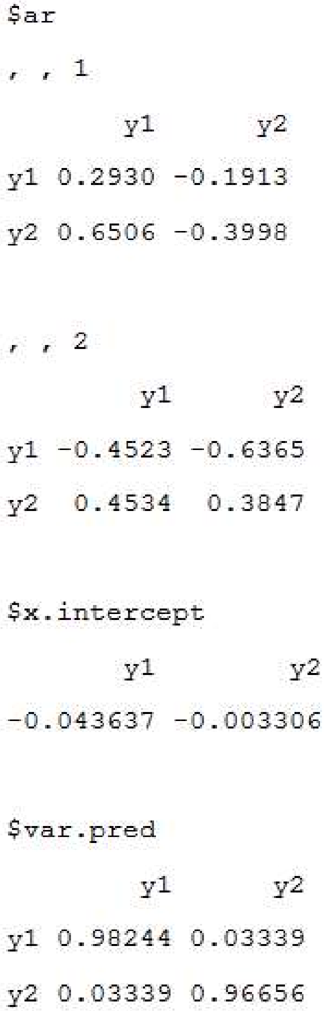
We obtain the following output:



```
VAR Estimation Results:
=========================
Endogenous variables: y1, y2
Deterministic variables: both
Sample size: 498
Log Likelihood: -1396.493
Roots of the characteristic polynomial:
0.9434 0.9434 0.8793 0.1518
Call:
VAR(y = cbind(y1, y2), p = 2, type = "both")

Estimation results for equation y1:
===================================
y1 = y1.l1 + y2.l1 + y1.l2 + y2.l2 + const + trend


        Estimate Std. Error t value Pr(>|t|)
y1.l1  0.2936162  0.0296083   9.917  < 2e-16 ***
y2.l1 -0.1899663  0.0372244  -5.103 4.78e-07 ***
y1.l2 -0.4534484  0.0374431 -12.110  < 2e-16 ***
y2.l2 -0.6357890  0.0243282 -26.134  < 2e-16 ***
const  9.5042440  0.8742211  10.872  < 2e-16 ***
trend  0.0001730  0.0003116   0.555    0.579
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 0.9969 on 492 degrees of freedom
Multiple R-Squared: 0.8582,   Adjusted R-squared: 0.8568
F-statistic: 595.5 on 5 and 492 DF,  p-value: < 2.2e-16
```

The estimates are very close to the simulated coefficients and the trend is not significant, as expected. For stationary data, when detrending is unnecessary, you may also use the ar.ols command to fit a VAR model:

```
fitvar2 = ar.ols(cbind(y1, y2), order=2)
```

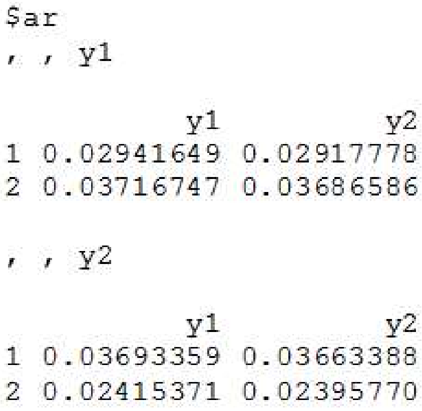


```
$ar

, , 1

       y1      y2

y1 0.2930 -0.1913

y2 0.6506 -0.3998


, , 2

        y1      y2

y1 -0.4523 -0.6365

y2  0.4534  0.3847


$x.intercept

      y1        y2

-0.043637 -0.003306


$var.pred

        y1      y2

y1 0.98244 0.03339

y2 0.03339 0.96656
```

In the first matrix given, read across a row to get the coefficients for a variable. The preceding commas followed by 1 or 2 indicate whether the coefficients are lag 1 or lag 2 variables respectively. The intercepts of the equations are given under `$x.intercept` – one intercept per variable.

The matrix under `$var.pred` gives the variance-covariance matrix of the residuals from the VAR(2) for the two variables. The variances are down the diagonal and could possibly be used to compare this model to higher order VARs as noted above.

The standard errors of the AR coefficients are given by the `fitvar2$asy.se.coef` command. The output is



```
$ar
, , y1

          y1         y2
1 0.02941649 0.02917778
2 0.03716747 0.03686586

, , y2

          y1         y2
1 0.03693359 0.03663388
2 0.02415371 0.02395770
```

As with the coefficients, read across rows. The first row gives the standard errors of the coefficients for the lag 1 variables that predict y1. The second row gives the standard errors for the coefficients that predict y2.

You may note that the coefficients are close to the VAR command except the intercept. This is because ar.ols estimates the model for x-mean(x). To match the intercept provided by the `summary(VAR(cbind(y1,y2), p=2, type="const"))` command, you must calculate the intercept as follows:

$$(y_{t,1} - \hat{\mu}_1) = \alpha_1 + \phi_{11}(y_{t-1,1} - \hat{\mu}_1) + \phi_{12}(y_{t-1,2} - \hat{\mu}_2) + \phi_{21}(yt - 2, 1 - \hat{\mu}_1) + \phi_{22}(y_{t-2,2} - \hat{\mu}_2) + w_{t,1}$$

$$y_{t,1} = \alpha_1 + \hat{\mu}_1(1 - \phi_{11} - \phi_{21}) - \hat{\mu}_2(\phi_{12} + \phi_{22}) + \phi_{11}y_{t-1,1} + \phi_{12}y_{t-1,2} + \phi_{21}y_{t-2,1} + \phi_{22}y_{t-2,2} + w_{t,1}$$

In our example, the intercept for the simulated model for $y_{t,1}$ equals

```
   -0.043637 -2.733607*(1-0.2930+0.4523) - 15.45479*(-0.1913-0.6365) = 9.580768,
```

and the estimated equation for $y_{t,1}$

$$y_{t,1} = 9.58 + .29y_{t-1,1} - .19y_{t-1,2} - .45y_{t-2,1} - .64y_{t-2,2}$$

## Estimation with Minitab

For Minitab users, here’s the general flow of what to do.

- Read the data into columns.
- Use Time Series > Lag to create the necessary lagged columns of the stationary values.
- Use Stat > ANOVA > General MANOVA.
- Enter the list of “present time” variables as the response variables.
- Enter the lagged x variables as covariates (and as the model).
- Click Results and select “Univariate Analysis” (to see the estimated regression coefficients for each equation).
- If desired, click “Storage” and select Residuals and/or Fits.

**Source URL:** https://onlinecourses.science.psu.edu/stat510/node/55