Financial Econometrics Using Stata

SIMONA BOFFELLI
University of Bergamo (Italy) and Centre for Econometric Analysis, Cass Business School, City University London (UK)

GIOVANNI URGA
Centre for Econometric Analysis, Cass Business School, City University London (UK) and University of Bergamo (Italy)

A Stata Press Publication
StataCorp LP
College Station, Texas

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in LaTeX 2ε
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

Print ISBN-10: 1-59718-214-1
Print ISBN-13: 978-1-59718-214-0
ePub ISBN-10: 1-59718-215-X
ePub ISBN-13: 978-1-59718-215-7
Mobi ISBN-10: 1-59718-216-8
Mobi ISBN-13: 978-1-59718-216-4

Library of Congress Control Number: 2016955738

Contents

(Pages omitted)

Preface

In this book, we illustrate how to use Stata to perform intermediate and advanced analyses in financial econometrics. The book is mainly for graduate students and practitioners who have an average econometric background. We provide a comprehensive overview of ARMA modeling, as well as univariate and multivariate GARCH models. Our approach consists of presenting a brief but rigorous summary of the theoretical framework, which we then implement using many examples. In particular, we report several empirical applications using real financial markets data to illustrate how to model conditional mean and conditional variance of typical financial time series. Users can easily replicate all the applications, executed using Stata 14, with the datasets and do-files we provide to get familiar with the techniques and Stata commands.

Throughout the book, we use acronyms extensively. For your convenience, we have included a glossary of acronyms at the end of the book.

The book is organized as follows. Chapter 1 provides an introduction to the following: the main features of financial time series, commands for obtaining descriptive statistics, analyzing normality, conducting stationarity tests, autocorrelation, heteroskedasticity, and model selection criteria. Chapter 2 provides a detailed description of the univariate ARMA framework to model the conditional mean of financial time series, with a specific focus on the S&P 500 returns time series.

Chapter 3 introduces the notion of conditional volatility and the popular family of GARCH models, specifically designed to capture the autoregressive nature of the volatility of asset returns. Brief descriptions of GARCH-M, asymmetric GARCH (SAARCH, TGARCH, GJR, APARCH) models, and nonlinear GARCH (PARCH, NGARCH, NGARCHK) models are followed by empirical implementations considering the S&P 500. Chapter 4 extends the univariate GARCH models to the multivariate framework, to account for not only volatility but also correlations between assets. Seminal multivariate GARCH models, such as vech and BEKK models, are described mainly to highlight the curse of dimensional issues; the chapter largely focuses on the CCC and DCC models widely used in the profession. Extensive empirical applications are conducted using four stock indices to stress the empirical validity of the MGARCH framework.

The last two chapters focus on risk management and contagion analyses, two leading research themes among academics and practitioners in the field of financial econometrics. In particular, chapter 5 introduces the concept of risk, risk measures, and their properties, concluding with an overview on some unilevel VaR and multilevel VaR backtesting procedures proposed in the literature. The empirical applications reported illustrate the methods and the way to implement them. Chapter 6 focuses on contagion analysis, where alternative methodologies are presented to evaluate the presence of a contagion. The techniques are illustrated by empirical applications examining the presence of a contagion among the United States, the United Kingdom, Germany, and Japan.

We acknowledge several people to whom we are in debt. First, we are grateful to David Drukker for having sponsored and encouraged us to pursue this project. His support was vital throughout the long gestation of the book, and he read and commented on several drafts of it. Second, we thank Elisabetta Pellini, who carefully read the complete version of the book and provided detailed and constructive feedback at various stages of the project on both the completion of the final document and the empirical applications. Third, we thank Jan Novotny for providing us with useful comments to a preliminary version of the book. Finally, we thank Lisa Gilmore and Deirdre Skaggs for production, LaTeX, and editorial assistance. Any mistakes within the book are ours.

(Pages omitted)

The Shapiro–Francia test is implemented by the `sfrancia` command:

```
. sfrancia return
                 Shapiro-Francia W´ test for normal data
    Variable │       Obs       W´           V´         z       Prob>z
─────────────┼──────────────────────────────────────────────────────
      return │    16,102    0.89665    902.543    18.495    0.00001
Note: The normal approximation to the sampling distribution of W´
      is valid for 10<=n<=5000 under the log transformation.
```

The Shapiro–Francia test also rejects the hypothesis of Gaussian distribution for the returns. Similarly to the Shapiro–Wilk test, we are provided with a synthetic index for the degree of departure from normality, `V'`, whose 95% confidence interval for accepting the null hypothesis of normality is [2.0, 2.8]. In our case, `V'` equals 902.54, clearly lying outside the confidence bounds.

Note that the Shapiro–Wilk test is accurate only when the number of observations lies between 4 and 2,000; Shapiro–Francia is accurate for 5 to 10,000 observations.

To summarize, in this section, we have presented several alternatives to test for normality. Each of the alternatives supplied evidence of departure from normality when applied to the S&P 500 returns series.

1.4 Stationarity

Financial econometricians generally work with returns rather than prices. In general, returns are characterized by time-invariant distribution, meaning that returns follow a stationary process.

Definition 1.4. A time series $\{r\}_t$ is strictly stationary if the joint distribution of $(r_{t_1}, \ldots, r_{t_k})$ is identical to that of $(r_{t_1+\tau}, \ldots, r_{t_k+\tau})$ for all positive integers τ.

Strict stationarity requires that the joint distribution of the subsequence $(r_{t_1}, \ldots, r_{t_k})$ does not change when it is shifted by an arbitrary amount τ. If we consider that stationarity requires that all moments of the joint distribution are invariant to time shifts, we can easily understand that the distributions that generate most financial time series are not strictly stationary.

Thus, we use a weaker definition of stationarity.

Definition 1.5. A time series $\{r\}_t$ is said to be weakly or covariance stationary if the following conditions hold true:

1. $E(r_t) = \mu$: the mean of the process is constant through time and equal to a constant μ;
2. $\mathrm{Var}(r_t) = \gamma_0$: the variance of the process is time invariant and equal to a finite constant $\gamma_0 < \infty$;

3. $\text{Cov}(r_t, r_{t+l}) = \gamma_l$, $|\gamma_l| < \infty$: the covariance of the process should not be time dependent, but it can be affected just by the distance between the two time ticks considered, equal to l.

Therefore, the weak stationarity imposes constraints on just the first two moments of the distribution, while the strict stationarity checks that the entire distribution is time invariant. Thus, weak stationarity does not imply strict stationarity, because the weak stationarity does not impose conditions on moments higher than the second. Nor does strict stationarity imply weak stationarity, because the definition of strict stationarity does not require the variance to be finite. However, under the Gaussian assumption, weak stationarity always implies strict stationarity, because the Gaussian distribution is entirely characterized by its first two moments.

A well-known stationary process is the white-noise process.

Definition 1.6. A return time series $\{r\}_t$ is said to follow an independent white-noise process if it satisfies the following conditions:

1. $E(r_t) = 0$
2. $E(r_t^2) = \sigma^2 < \infty$
3. $E(r_t, r_{t-j}) = 0 \;\; \forall j \neq 0$

A white-noise process has finite mean and variance, and it does not show any time pattern, meaning that the current realizations of a process cannot help in predicting its future realizations. Therefore, because independence implies absence of autocorrelation, a white-noise process is characterized by almost flat autocorrelation function (ACF) and partial autocorrelation function (PACF), with no correlation statistically different from 0. Returns can usually be ascribed to the class of white-noise process, coherently with the assumption of efficient market hypothesis.

We now simulate a Gaussian white-noise process. Note that normality is not a general requirement for this process. We start by setting the length of our simulation period equal to 1,000 by using the `set obs` command, and we generate a time index (`index`) of the same length. In addition, we set the seed (the starting point for any random sequence) to ensure we get the same sequence of random numbers every time the simulation is run—which is important when we are ready to replicate the simulation. Finally, we extract simulated numbers from a standard normal distribution by using the `rnormal()` function, taking as an argument the mean and the standard deviation that, in our case, we respectively set equal to 0 and to 1.

```
. clear
. set obs 1000
number of observations (_N) was 0, now 1,000
. generate index = _n
. * fix seed
. set seed 1
```

```
. generate wn1 = rnormal(0,1)
. generate wn2 = rnormal(0,1)
. generate wn3 = rnormal(0,1)
. tsset index
        time variable:  index, 1 to 1000
                delta:  1 unit
```

We then use the `tsline` command to graph the results; see figure 1.8.

```
. tsline wn1 wn2 wn3
```

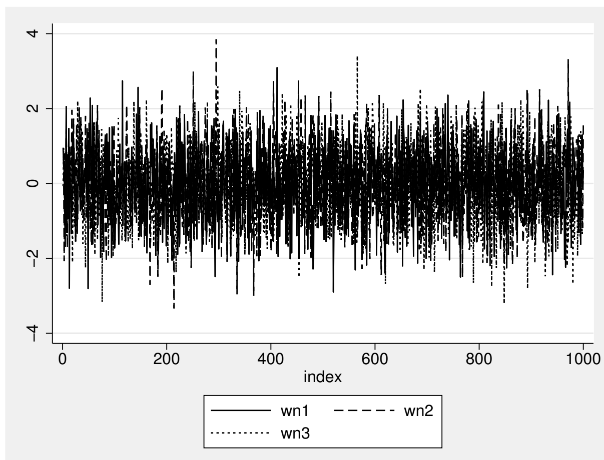


Figure 1.8. Simulated white-noise processes

Although the three processes are almost not distinguishable, they all move around the zero line, suggesting that they are stationary.

A common nonstationary process is the random walk.

Definition 1.7. A time series $\{p_t\}$ is a random walk if it satisfies

$$p_t = p_{t-1} + \varepsilon_t \tag{1.1}$$

where ε_t is a white-noise process.

A random walk is the typical process that is able to describe the behavior of stock prices.

A generalization of (1.1) is the random walk with drift:

$$p_t = \mu + p_{t-1} + \varepsilon_t$$

where μ, commonly called drift, represents the time trend of the log price.

We can obtain a random-walk process (see figure 1.9) as the cumulative sum of the white-noise processes just simulated above.

```
. generate rw1 = sum(wn1)
. generate rw2 = sum(wn2)
. generate rw3 = sum(wn3)
. tsline rw1 rw2 rw3
```

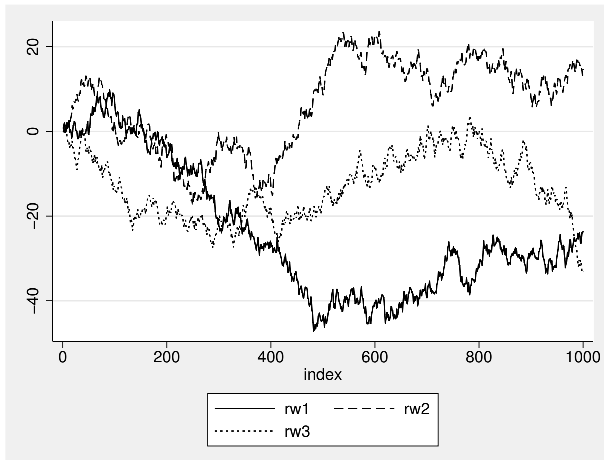


Figure 1.9. Simulated random-walk processes

All three simulated processes show a trend suggesting that they are not stationary.

1.4.1 Stationarity tests

With the purpose of establishing whether a time series is stationary or nonstationary, we can use the unit-root test. A process with a unit root has time-dependent variance, thus violating the condition of weak stationarity, $\text{Var}(r_t) = \gamma_0$.

Stata can test for the presence of a unit root by using two main testing procedures: the augmented Dickey–Fuller (ADF, 1979) test and the Phillips and Perron (PP, 1988) test.

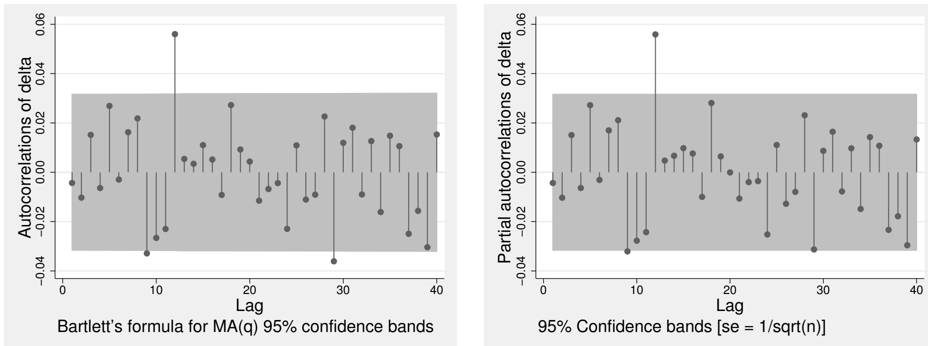
Given a time series $\{y_t\}$, the ADF test is based on the regression

$$\Delta y_t = \alpha + \beta t + \theta y_{t-1} + \delta_1 \Delta y_{t-1} + \cdots + \delta_{p-1} \Delta y_{t-p+1} + \varepsilon_t \tag{1.2}$$

where α is a constant, t is the time trend, and p is the order of the autoregressive process.

The null hypothesis under which the ADF test is distributed is that the time series is not stationary, corresponding to $\theta = 0$, against the alternative that it is stationary,

(Pages omitted)



(a) ACF (b) PACF

Figure 2.11. ACF and PACF for EUR–USD daily returns

Neither the ACF nor the PACF show any evidence of a time dependence structure in the data, with almost no peak being statistically significantly different from 0. This result is because of the high liquidity of the foreign market, making it extremely efficient.

2.4.1 Model estimation

We can use the `arima` command to fit ARMA models.

When checking the estimates, remember that Stata reports the intercept as the unconditional mean. For instance, given an ARMA(p, q) model,

$$r_t = \delta + \phi_1 r_{t-1} + \cdots + \phi_p r_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

the intercept shown in the output is actually $\delta/1 - \phi_1 - \cdots - \phi_p$.

Before starting a full empirical implementation of an ARMA model, we briefly describe the estimation technique implemented in Stata. `arima` implements the conditional and the unconditional ML estimators. The conditional ML estimator drops the observations lost to lagged values of the dependent variable or lagged errors. The unconditional ML estimator uses the structure of the model to identify values to fill in for these missing values. The unconditional estimator can be more efficient and is frequently preferred. All the details can be found in [TS] **arima**.

As decided when we checked the ACF and the PACF, we now fit an ARMA(2,1) model on the S&P 500 daily returns:

```
. use http://www.stata-press.com/data/feus/spdaily, clear
. tsset newdate
        time variable:  newdate, 03jan1950 to 31dec2013
                delta:  1 day
```

```
. arima return, ar(1/2) ma(1)
(setting optimization to BHHH)
Iteration 0:   log likelihood =  51721.001
Iteration 1:   log likelihood =  51721.421
Iteration 2:   log likelihood =  51721.438
Iteration 3:   log likelihood =  51721.448
Iteration 4:   log likelihood =  51721.453
(switching optimization to BFGS)
Iteration 5:   log likelihood =  51721.457
Iteration 6:   log likelihood =  51721.465
Iteration 7:   log likelihood =  51721.467
Iteration 8:   log likelihood =  51721.469
Iteration 9:   log likelihood =  51721.469
Iteration 10:  log likelihood =  51721.469
Iteration 11:  log likelihood =  51721.469
ARIMA regression
Sample:  04jan1950 - 31dec2013                  Number of obs     =      16102
                                                Wald chi2(3)      =     277.12
Log likelihood =  51721.47                      Prob > chi2       =     0.0000
```

return	Coef.	OPG Std. Err.	z	P>|z|	[95% Conf.	Interval]
return						
_cons	.0002924	.0000799	3.66	0.000	.0001358	.0004491
ARMA						
ar						
L1.	-.068257	.0899525	-0.76	0.448	-.2445607	.1080466
L2.	-.0400998	.0041099	-9.76	0.000	-.0481552	-.0320445
ma						
L1.	.0981323	.0898873	1.09	0.275	-.0780435	.2743081
/sigma	.0097445	.0000152	640.54	0.000	.0097147	.0097743

```
Note: The test of the variance against zero is one sided, and the two-sided
      confidence interval is truncated at zero.
. estimates store ARMA21
```

We have fit a model with two lags for the AR part, `ar(2)`, and one lag for the MA part, `ma(1)`. Alternatively, we could have typed **`arima returns, arima(2,0,1)`**; here the first number indicates that we want to add two lags for the AR part, the second number indicates that we want to add the order of integration (here equal to 0), and the third number indicates that we want to add one lag to the MA part.

In the first part of the output, we find some information about the optimization procedure, with the iterations of the algorithm aimed at maximizing the log-likelihood function. The convergence is achieved in 11 steps, and it stops at the log-likelihood value of 51,721.47. We find this value just above the table. In addition, we are informed that the estimation sample consists of 16,102 observations and that the model is overall statistically significant, as suggested by the Wald test. The table provides parameters and standard errors, the t test for the statistical significance of parameters `z` and `P>|z|`, and the 95% confidence interval. `OPG Std. Err.` reminds us that Stata is using the

(Pages omitted)

An interesting relationship exists between the resampling frequency and the β parameter accounting for persistence in the GARCH model. To illustrate this point, we now fit a GARCH(1,1) model on monthly returns using a resampled time series of our daily S&P 500 returns, under the Gaussianity assumption and with no mean equation.

```
. use http://www.stata-press.com/data/feus/spmonthly, clear
. arch return, arch(1) garch(1) nolog
ARCH family regression
Sample: 2 - 769                                 Number of obs    =        768
Distribution: Gaussian                          Wald chi2(.)     =          .
Log likelihood =  1372.229                      Prob > chi2      =          .

------------------------------------------------------------------------------
             |                 OPG
      return |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
return       |
       _cons |   .0065792    .001475     4.46   0.000     .0036883    .0094702
-------------+----------------------------------------------------------------
ARCH         |
        arch |
         L1. |   .1130881   .0245265     4.61   0.000     .0650171    .1611591
             |
       garch |
         L1. |   .8405205   .0279828    30.04   0.000     .7856753    .8953657
             |
       _cons |    .000092   .0000315     2.92   0.004     .0000303    .0001538
------------------------------------------------------------------------------
```

The β parameter is equal to 0.84, while it equaled 0.91 when we fit the model on daily data. Therefore, as expected, we confirm that the variance process is more persistent when measured on higher frequency data.

A peculiar case of GARCH models is the integrated GARCH (IGARCH) model, which is characterized by the presence of a unit root in the autoregressive dynamic of squared residuals, corresponding to setting $\alpha + \beta = 1$ in (3.9). The IGARCH(1,1) model takes the following form:

$$h_t = \omega + \alpha\varepsilon_{t-1}^2 + (1-\alpha)h_{t-1}$$

Given that the IGARCH model is nonstationary, this process is useful when the conditional variance is highly serially correlated (long-memory process), for instance, when working with intraday data.

An example of the IGARCH model is the risk metrics model. In this case, the values of the ARCH and GARCH parameters are fixed: $\alpha = (1-\lambda)$ and $\beta = \lambda$.

$$h_t = (1-\lambda)\varepsilon_{t-1}^2 + \lambda h_{t-1}$$

where $\omega = 0$, $\lambda = 0.94$ for daily data, and $\lambda = 0.97$ for weekly data.

3.4.2 GARCH in mean

The conditional variance can even enter the equation for the conditional mean. In that case, we have a GARCH-in-mean (GARCH-M) model. The GARCH-M model was proposed to allow us to account for the widely studied relationship between risk and return: as the volatility of an asset raises, so does the expected risk premium. We can represent the GARCH-M as follows:

$$r_t = \omega + \beta x_t + \theta h_t + \varepsilon_t \tag{3.10}$$

where h_t follows a GARCH process, θ is the risk aversion parameter, and x_t is the vector of exogenous variables at time t.

Instead of the linear form in (3.10), we can insert the conditional variance h_t in the equation for the conditional mean by adopting a nonlinear function $g(\cdot)$:

$$r_t = \omega + \beta x_t + \theta_0 g\left(h_t\right) + \theta_1 g\left(h_{t-1}\right) + \theta_2 g\left(h_{t-2}\right) + \cdots + \varepsilon_t$$

We can fit an ARCH-in-mean (ARCH-M) model by specifying the `archm` option in the usual `arch` command. Then, the `archmlags(`*numlist*`)` option specifies the number of lags for the conditional variance that we want to add in the conditional mean equation. For instance, by specifying `archmlags(0)`, we add just the contemporaneous conditional variance h_t; by specifying `archmlags(1)`, we are adding the once-lagged variance h_{t-1}.

```
. use http://www.stata-press.com/data/feus/spdaily
. tsset newdate
        time variable:  newdate, 03jan1950 to 31dec2013
                delta:  1 day
. arch return, arch(1) garch(1) archm archmlags(1) nolog
ARCH family regression
Sample: 04jan1950 - 31dec2013                   Number of obs   =     16,102
Distribution: Gaussian                          Wald chi2(2)    =      11.95
Log likelihood =   54792.1                      Prob > chi2     =     0.0025

------------------------------------------------------------------------------
             |                 OPG
      return |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
return       |
       _cons |   .0003063   .0000786     3.90   0.000     .0001522    .0004605
-------------+----------------------------------------------------------------
ARCHM        |
      sigma2 |
         --. |   17.18942   6.664707     2.58   0.010     4.126831      30.252
         L1. |  -13.97885   6.509351    -2.15   0.032    -26.73695   -1.220761
-------------+----------------------------------------------------------------
ARCH         |
        arch |
         L1. |    .081337   .0016464    49.40   0.000       .07811    .0845639
             |
       garch |
         L1. |   .9122545   .0022107   412.65   0.000     .9079216    .9165874
             |
       _cons |   8.03e-07   6.76e-08    11.89   0.000     6.71e-07    9.35e-07
------------------------------------------------------------------------------
```

In the output reported above, we can see the two extra parameters for the ARCH-M part as well as the `archmlags(1)` option. These two parameters correspond to coefficients in (3.10), loading h_t and h_{t-1}, respectively, and they are both statistically significant.

3.4.3 Forecasting

On the basis of a GARCH(1,1), we can obtain a volatility forecast at time $t+1$ as

$$E\left(h_{t+1}|I_t\right) = E\left(\widehat{\omega} + \widehat{\alpha}\varepsilon_t^2 + \widehat{\beta}h_t|I_t\right)$$
$$E\left(h_{t+1}|I_t\right) = \widehat{\omega} + \widehat{\alpha}\varepsilon_t^2 + \widehat{\beta}h_t$$

where we are exploiting the fact that at time t, given the information set I_t, we know both quantities ε_t and h_t. When moving to forecasts at the next time $t+k$ with $k \geq 2$, it is necessary to distinguish between dynamic and static forecasts.

In the case of dynamic forecast, the informative set remains the same through time, and equal to I_t, which is where the time series stops. For instance, at time $t+2$, the dynamic forecast for the conditional volatility is