Ordinary least squares estimation and time series data

One of the assumptions underlying ordinary least squares (OLS) estimation is that the errors be uncorrelated. Of course, this assumption can easily be violated for time series data, since it is quite reasonable to think that a prediction that is (say) too high in June could also be too high in May and July. That kind of cyclical effect is indicative of positive autocorrelation, and it is quite common in time series data. But say we ignore this fact; why is it a problem to use OLS if the errors are autocorrelated?

The following two tables can help to answer that. Consider a simple regression problem, and let ρ be the first order autocorrelation of the errors (i.e., $\rho = corr(\varepsilon_i, \varepsilon_{i+1})$) and λ be the first order autocorrelation of the predicting variable x (it's likely that this, too, would exhibit autocorrelation; this is not a violation of any assumptions, but it can affect the properties of OLS estimators if there is also autocorrelation of errors). Further, assume the particular autocorrelation structure known as a *first order autoregressive model* (we'll talk more about this a little later).

The first problem with using OLS estimates in the context of autocorrelated errors is that they are *inefficient*; that is, they have higher variability (as estimates of the true parameters) than they should. The following table gives the efficiency of the OLS estimator of β_1 compared to the best possible estimator (the efficiency is simply the ratio of variances):

	ρ								
λ	–0.9	–0.8	–0.5	–0.2	0	0.2	0.5	0.8	0.9
0.0	10.5	22.0	60.0	92.3	100.0	92.3	60.0	22.0	10.5
0.2	12.6	25.4	63.2	92.9	100.0	92.3	58.4	19.8	9.1
0.5	18.5	34.3	71.4	94.6	100.0	93.5	60.0	18.4	7.9
0.8	35.9	56.2	85.4	97.5	100.0	96.6	71.4	22.0	8.4
0.9	52.8	71.8	92.0	98.7	100.0	98.1	81.3	29.3	10.5

It is apparent that if errors are autocorrelated, the OLS estimator can be seriously inefficient. For example, if $\rho = \lambda = .9$, the variance of the OLS estimator is 10 times

that of the best estimator. For positively autocorrelated errors, the inefficiency is fairly insensitive to the autocorrelation of the predictor, but for negatively autocorrelated errors, a positively autocorrelated predictor can actually help (it's fairly unlikely, however, that the signs of the autocorrelations of the predictor and of the errors would be different). Note, by the way, that results for negatively autocorrelated predictors mimic those above, except that the role of the sign of the autocorrelation of the errors is reversed (a negatively autocorrelated predictor is more trouble for negatively autocorrelated errors).

This is not good, but an even bigger problem also exists. The standard error of $\hat{\beta}_1$ that is output by the computer is no longer correct; it is estimating the wrong thing. The following table gives the percentage bias in estimating $var(\hat{\beta}_1)$ if the OLS computer output is used:

					ρ				
λ	–0.9	–0.8	–0.5	–0.2	0	0.2	0.5	0.8	0.9
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.2	43.9	38.1	22.2	8.3	0.0	–7.7	–18.2	–27.5	–30.5
0.5	163.6	133.3	40.0	22.2	0.0	–18.2	–40.0	–57.1	–62.1
0.8	514.3	355.6	133.3	38.1	0.0	–27.6	–57.1	–78.0	–83.7
0.9	852.6	514.3	163.6	43.9	0.0	–30.5	–62.1	–83.7	–89.5

It is apparent that using the usual measures of fit can lead to very misleading inferences. For example, if $\rho = \lambda = .9$, the estimated variance of $\hat{\beta}_1$ is about 10% of its true value. This implies that the t–statistic for β_1 is about 3.1 times too large (a similar inflation of F and R^2 values also occurs). Thus, if left uncorrected, an insignificant relationship (say $t = 1.5$) can be mistakenly viewed as highly significant (apparent $t = 4.65$). It often happens that a regression on time series data with $R^2 = .8$ has the R^2 drop down to .3 or .4 when autocorrelation is addressed. Note also that if λ and ρ are of opposite sign, the apparent strength of the regression is too low, rather than too high.

Identifying autocorrelation

Since it is autocorrelation of the errors that is a violation of regression assumptions, it shouldn't be surprising that it is the (standardized) residuals that are used to identify possible autocorrelation. We've already talked about one way that this is done — whenever there is a time ordering to the data, a time series plot of the residuals should be constructed and examined for possible evidence of cyclical behavior. Note that the observations in all time series data **must** be ordered in correct chronological order (earliest to latest), rather than reverse order, or else tests and estimation methods are incorrect.

Consider a hypothetical data set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p, \mathbf{y}\}$, and a hypothesized linear regression model

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + \varepsilon_i.$$

In addition to a time series plot of the residuals, there are several formal tests that can be used to identify the presence of autocorrelation in the residuals.

The *Durbin–Watson test* is a highly parametric test for autocorrelation. It is assumed that under the null **all** of the usual assumptions for regression hold:

$$H_0 : \varepsilon_i \sim N(0, \sigma^2) \text{ for all } i, \text{ corr}(\varepsilon_i, \varepsilon_j) = 0 \text{ for } i \neq j.$$

Further, the form of possible autocorrelation is also specified as being from an autoregressive model of order 1 [AR(1)]:

$$H_a : \varepsilon \sim \text{AR}(1),\ \rho \neq 0$$

The AR(1) model says that autocorrelation is due to the following structure for the errors:

$$\varepsilon_i = \rho \varepsilon_{i-1} + z_i, \quad i = 2, \ldots, n,$$

where $|\rho| < 1$ and the z_i are independent $N(0, \sigma^2)$ random variables. This implies that $\rho_s = \rho^s$, where ρ_s is the s^{th} order autocorrelation [$\text{corr}(\varepsilon_i, \varepsilon_{i+s})$]. So, for example, if $\rho = .7$, then $\rho_1 = .7, \rho_2 = .49, \rho_3 = .34$, and so on. That is, the autocorrelation in the errors goes down geometrically as the distance between them goes up.

The Durbin–Watson test is simply

$$DW = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n} e_i^2},$$

where e_i is the i^{th} residual. Small values of DW indicate positive autocorrelation, while large values indicate negative autocorrelation.

The enclosed tables give critical values for the test. The Durbin-Watson statistic has the unfortunate property that it is not *pivotal*; that is, its distribution under the null hypothesis is a function of the actual data values. This means that there is no single critical value (given the number of observations and predictors) for which you can determine statistical significance, but rather only a range of critical values (see the attached tables). For large samples ($n \geq 100$, say), an approximate z–statistic for the Durbin–Watson test is

$$z = \left(\frac{DW}{2} - 1\right)\sqrt{n},$$

which can be compared to a Gaussian critical value. Note that a positive value of z indicates negative autocorrelation, while a negative value indicates positive autocorrelation.

Since the Durbin–Watson test has so many assumptions, it is important to check them if you're going to use it. That is, you must look at residual plots to check homoscedasticity and normality, and you can look at an autocorrelation function (ACF) plot of the standardized residuals to see if the observed autocorrelations appear to be consistent with the AR(1) model; this plot gives correlations of a variable with itself shifted by one time period, two time periods, etc.). If the autocorrelations decay roughly at the geometric rate consistent with an AR(1) process, that supports the use of the Durbin–Watson test. Note that a complete lack of evidence of any autocorrelation in an ACF plot **is** consistent with an AR(1) process (one with $\rho = 0$). The ACF plot also can be used to identify other types of autocorrelation, such as seasonality and nonstationarity, which we will discuss later.

The individual residual autocorrelations that are used to construct the autocorrelation function also can be used to test for autocorrelation in the errors, since they should be close to zero if there is no autocorrelation in the errors. For each order autocorrelation, the hypotheses being tested are

$$H_0 : \rho_j = 0$$

versus

$$H_a : \rho_j \neq 0.$$

The ACF plot gives $\alpha = .05$ limits for each estimated autocorrelation, corresponding to a t–test for the significance of the estimated coefficient relative to the true coefficient equaling zero. This test requires the usual assumptions on the errors for small samples, but normality is not needed for large samples (as the Central Limit Theorem applies).

The *runs test*, being nonparametric in nature, requires virtually no assumptions about the data. The hypotheses being tested are very general:

$$H_0 : \text{there is no autocorrelation in the errors}$$

versus

$$H_a : \text{there is autocorrelation.}$$

Let n_+ be the number of positive residuals, and n_- be the number of negative ones in a sample of size n. A "run" is defined as a set of consecutive observations where the residuals have the same sign. In the presence of positive autocorrelation you would expect positive and negative residuals to tend to occur together, resulting in fewer than expected runs; in the presence of negative autocorrelation you would expect positive residuals to tend to be followed by negative ones (and vice versa), resulting in more runs than expected. It can be shown that under H_0 the expected number of runs is $\mu = \frac{2n_+n_-}{n} + 1 \approx \frac{n}{2} + 1$, while the variance of the number of runs is $\sigma^2 = \frac{2n_+n_-(2n_+n_- - n)}{n^2(n-1)} \approx \frac{n^2-2n}{4(n-1)}$. The runs test is a z–test, comparing the observed number of runs u to the expected number:

$$z = \frac{|u - \mu| - \frac{1}{2}}{\sigma}$$

(the "$-\frac{1}{2}$" is a continuity correction). For large enough n, z is approximately normally distributed, and a tail probability can be obtained.

Note, by the way, that neither the runs test nor ACF plot can be produced in `Minitab` if there are missing values in the residuals (other than at the very beginning or very end of the series). Note also that even if the assumptions of all three tests are satisfied that doesn't guarantee that they will agree on statistical significance. It might be that the runs

test will not be statistically significant when the Durbin-Watson test is because of the lower power of the runs test. It also might be that there isn't enough evidence to reject that the autocorrelation at any particular lag is 0, based on the ACF, but taken together as a whole they are significantly different from all being equal to 0 according the runs test. The point is that not rejecting the null is not the same as accepting it, these tests should be viewed as diagnostics designed to raise potential red flags, not hurdles that must be crossed before considering the possibility of autocorrelation.

Addressing autocorrelation

Say we've identified autocorrelation in the residuals from a regression. What should we do? There are many different possibilities, ranging from relatively simple approaches to quite complicated ones. Note, by the way, that addressing autocorrelation is not the full story — this is still a regression problem, and all of the usual checks (scatter plots, residual plots, diagnostics, etc.) are still essential.

Detrending and deseasonalizing

The structure in time series data is often greatly simplified if broad trends and seasonal effects are removed. If a time series plot of a variable shows steadily increasing (or decreasing) values over time, the variable can be *detrended* by running a regression on a time index variable (that is, the case number), and then using the residuals as the detrended series. If the series has natural seasonal effects, these too can be handled using regression. For example, say the variable being examined is quarterly sales of ice cream. We wouldn't be surprised to see seasonal effects in such a variable. It can be *deseasonalized* by running a regression on three indicator variables that identify first, second, and third quarter observations, respectively, and then using the residuals as the deseasonalized series. Seasonal effects are often apparent in the ACF plot of the residuals as a large autocorrelation at the lag corresponding to the period of the seasonality (lag 4 for quarterly data, lag 12 for monthly data, etc.). So, for example, unemployment rates are usually reported in deseasonalized form, which roughly corresponds to taking the residuals from a regression on the quarter effects and adding the overall average unemployment rate back.

Usually, we are not interested in creating as a final product detrended or deseasonalized variables; rather, we would just like to include trend and/or seasonal effects as part of a time series regression model. In this situation, all that is required is to add a time index and/or seasonal indicator variables as additional predictors in the regression model (that is, you typically don't need to detrend or deseasonalize each variable in the model separately). Of course, this implies that you should look at a time series plot of the target variable (that is, a plot of the target variable versus the time index), and side–by–side boxplots of the target variable separated by season to see if detrending or deseasonalizing seem to be worth considering. Indeed, you should look at these plots routinely in any regression on time series data (the boxplots for a seasonal effect only if it is meaningful in your context, of course), since trend and seasonal effects are so common.

Sometimes trend effects are actually reflecting more fundamental properties of the variables that should be addressed. For example, variables that are related to population will increase over time simply because of increasing population. This is not what we're typically interested in, so such effects should be removed before analyzing the data by converting to per capita measures. Similarly, variables measured in current dollars will increase over time because of inflation; such variables should be converted to constant dollars using a price deflator like the consumer price index.

Lagging and differencing

Autocorrelation can sometimes be handled by using values of the target variable from the previous time period(s) as predictors in the model. So, for example, this quarter's sales might be regressed on last quarter's sales. Such variables are called *lagged* variables. A plot of the target versus the lagged target will often show a strong relationship between the two. Even more importantly, in a regression , autocorrelation has pretty much disappeared (an important point, however: the assumed distribution for the Durbin–Watson statistic is not valid if a lagged version of the target is used as a predictor, so the observed statistic should not be evaluated for statistical significance). Obviously, if you are considering using the lagged response as a predictor, you need to look at a scatter plot of the response versus the lagged response, since you need to look at scatter plots of the response versus **all** potential

predictors.

In a multiple regression, if you decide to include a lagged target variable as a potential predictor, it is just that — a potential predictor — and should be treated as such. So, for example, performing a best subsets regression based on all of your available predictors is appropriate, since the presence of the lagged target might change the usefulness of other variables in the model (that is, your previously chosen predictors might no longer be the appropriate choices).

A transformation related to lagging is *differencing* the data. This operation is appropriate when the **change** in a variable from one time period to the next is hypothesized to be uncorrelated with the changes at other time points. Data that follow a random walk, and more generally series that are *nonstationary*, satisfy this condition, and benefit from differencing, which helps explain why stock returns are much more interesting than stock prices (indeed, stock returns are almost **always** the correct thing to study, rather than stock prices). Differencing the target variable is often particularly useful when an ACF plot of the residuals indicates slowly decaying autocorrelations. If you are differencing a logged variable you should use natural (base e) logs rather than common (base 10) logs, as then the differenced value corresponds approximately to a proportional difference. So, for example, differencing the natural log of stock price yields the so-called *log return*, which is roughly equal to the return. Of course, in that situation you could just as easily use the actual return as the predictor or response variable.

It is important to remember that once a target variable is differenced, the problem has changed. The goal is no longer to try to build a model that predicts the (original) target, but rather now the *change* in the target. The close connection between differencing a target variable and lagging one can be seen from the regression model form after differencing:

$$y_i - y_{i-1} = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \varepsilon_i$$

is equivalent to a regression including a lagged value of y as a predictor,

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_p x_{pi} + \beta_{p+1} y_{i-1} + \varepsilon_i,$$

with the coefficient of the lagged y (β_{p+1}) equaling one.

If the target variable y follows a random walk, this relationship is particularly clear. A random walk is characterized by the change in y being random Gaussian noise; that is, $\beta_1 = \cdots = \beta_p = 0$ in the models above. A regression of y on lagged y will be highly significant (with coefficient for lagged y close to one), while one with differenced y as the target will be insignificant, yet **the two models are reflecting exactly the same relationship**. This can be seen if the standard errors of the estimate for the two models are compared; since both models have the same error structure, the estimates of σ will be very close, even though one model has a high R^2 and the other has a low one. Stock prices are often modeled as following a geometric random walk. What that implies is that logged stock price yesterday is a very good predictor of logged stock price today. What it also means is that the change in logged stock price (the return) is pretty much unrelated to anything else. The latter model, with its low R^2 and F statistics, is the correct representation of the relationship, even though it "doesn't look as good."

It is important to note, however, that differencing data can sometimes result in what is called *overdifferencing*, where the resultant series exhibits significant negative autocorrelation. In this circumstance, using a lagged version of the target as the predictor can be considerably more effective than differencing the data, which would be reflected in the estimated coefficient for the lagged target variable being significantly different from 1.

Using a lagged response variable as a predictor can also help address seasonality. For example, in quarterly data, using the response variable lagged by four periods (that is, the response from the period one year earlier) can account for quarterly seasonal effects in an effective way.

Predicting variables also can be included in a regression in lagged or differenced form. It is important to recognize, however, that this is only sensible if there is a good reason to believe that such variables have predictive power. So, for example, if you believed that higher interest rates might cause higher unemployment, but only after a three month lag, it would be sensible to use interest rates from three months earlier as a predictor.

The Cochrane–Orcutt procedure

A more complicated approach to autocorrelation is to use *generalized least squares*. The principle is to transform the problem from one that exhibits autocorrelation to one

that doesn't, and then analyze on the data that don't have autocorrelation. Assume the autocorrelation structure is AR(1), as described above. Consider for simplicity a simple regression model, although the discussion here generalizes in a straightforward way to multiple regression. Thus, we have

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i.$$

Now, consider the transformation to

$$y_i^* = y_i - \rho y_{i-1}.$$

Substituting into the regression model above we get

$$\begin{aligned} y_i^* &= y_i - \rho y_{i-1} \\ &= \beta_0 + \beta_1 x_i + \varepsilon_i - \rho(\beta_0 + \beta_1 x_{i-1} + \varepsilon_{i-1}) \\ &= \beta_0(1-\rho) + \beta_1(x_i - \rho x_{i-1}) + \varepsilon_i - \rho\varepsilon_{i-1} \\ &= \beta_0(1-\rho) + \beta_1(x_i - \rho x_{i-1}) + z_i \\ &\equiv \beta_0^* + \beta_1 x_i^* + z_i, \end{aligned}$$

where $\beta_0^* = \beta_0(1-\rho)$ and $x_i^* = x_i - \rho x_{i-1}$. Thus, the regression of y_i^* on x_i^* provides estimates of β_0^* and β_1 that are appropriate, since the error z_i are independent and normally distributed. We aren't really interested in β_0^*, but that's okay; we just convert back to an estimate to an estimate of β_0 by dividing by $1-\rho$.

So, the **Cochrane–Orcutt procedure** is as follows:

(1) Determine an estimate of ρ. A good one is the entry for lag 1 in the ACF plot (call it $\hat{\rho}$).

(2) Form the transformed variables $y_i^* = y_i - \hat{\rho} y_{i-1}$ and $x_i^* = x_i - \hat{\rho} x_{i-1}$ (do this for **each** of the predicting variables).

(3) Do the regression of y_i^* on the x_i^*'s. The slope estimates are left alone; the constant term estimate is adjusted by $\hat{\beta}_0 = \hat{\beta}_0^*/(1-\hat{\rho})$. A rough prediction interval is $\hat{y} \pm 2\hat{\sigma}/\sqrt{1-\hat{\rho}^2}$, where $\hat{\sigma}$ is the standard error of the estimate from the Cochrane–Orcutt fit.

It is very important to remember that the Cochrane–Orcutt procedure is merely a computational trick that allows a generalized least squares analysis using ordinary least squares programs. There is **no** physical meaning to y^* or x^*; they are merely tools that are used to get the GLS fit. However, since the Cochrane–Orcutt regression mimics that GLS fit, the usual measures of fit (R^2, F, t), residual plots, and regression diagnostics from the Cochrane–Orcutt fit can be interpreted in the usual way, since they are the appropriate ones from a GLS fit. One important note: the Cochrane–Orcutt procedure is **not** appropriate if a lagged version of the response variable is being used as a predictor.

A variation on the Cochrane-Orcutt procedure is the Prais-Winsten procedure, which replaces the x_1^* and y_1^* (which are missing when using Cochrane-Orcutt) with $x_1\sqrt{1-\hat{\rho}^2}$ and $y_1\sqrt{1-\hat{\rho}^2}$, respectively. Typically the results of the two approaches are very similar. In addition, each procedure can be iterated, by successively substituting the new estimates of $\boldsymbol{\beta}$ into the appropriate formulas.

The GLS formulation being approximated using Cochrane-Orcutt or Prais-Winston is straightforward in matrix notation. The model is

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

with $E(\boldsymbol{\varepsilon}) = \mathbf{0}$ and $Var(\boldsymbol{\varepsilon}) = V\sigma^2$, where V is the $n \times n$ matrix of autocorrelations in the errors. Matrix manipulations then give the following:

$$\begin{aligned} \Rightarrow V^{-1/2}\mathbf{y} &= V^{-1/2}X\boldsymbol{\beta} + V^{-1/2}\boldsymbol{\varepsilon} \\ \Leftrightarrow \mathbf{y}^* &= Z\boldsymbol{\beta} + \boldsymbol{\delta}. \end{aligned}$$

Thus,

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (Z'Z)^{-1}Z'\mathbf{y}^* \\ &= (X'V^{-1}X)^{-1}X'V^{-1}\mathbf{y}. \end{aligned}$$

The Cochrane–Orcutt procedure does this fitting for the special case of AR(1) errors; in that case, V has the particularly simple form $v_{ij} \equiv \text{corr}(\varepsilon_i, \varepsilon_j) = \rho^{|i-j|}$.

Minitab commands

The Durbin–Watson test is obtained in `Minitab` by clicking in the `Durbin-Watson statistic` under `Results` when fitting a regression. An ACF plot is obtained by clicking on `Stat` → `Time Series` → `Autocorrelation`, and entering the variable name under `Series:`. `Minitab` does not give the runs test as a regression option, but it can be calculated by first saving the residuals into a column and then going to `Stat` → `Nonparametrics` → `Runs Test`.

To create a time index variable (so that a time trend can be fit in a regression model), click on `Calc` → `Make patterned data` → `Simple set of numbers`. Enter the variable name (*Time*, for example) under `Store patterned data in:`, the value *1* under `From first value:`, and the sample size (n) under `To last value:`. To create indicator variables that define a categorical variable (so that a seasonal effect can be fit in a regression model), click on `Calc` → `Make Indicator Variables`. Under `Indicator variables for` enter the categorical variable. The distinct categories will be listed below, along with names for the constructed indicator variables (which can be changed).

Lagged variables are formed using the `LAG()` function in `Calc` → `Calculator`. Note that the data must be in chronological order (**not** reverse chronological order) in order for the lagging and differencing operations to be implemented correctly.

