Lecture 13
Auto/cross-correlation

1

Generalized Regression Model

• The generalized regression model's assumptions:

(**A1**) DGP: $\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is correctly specified.

(**A2**) $E[\boldsymbol{\varepsilon}|\mathbf{X}] = 0$

(**A3'**) $Var[\boldsymbol{\varepsilon}|\mathbf{X}] = \boldsymbol{\Sigma} = \sigma^2\boldsymbol{\Omega}$.

(**A4**) **X** has full column rank – $rank(\mathbf{X}) = k$ –, where $T \geq k$.

• We assume that the ε's in the sample are not longer generated independently of each other. Ignoring hetersocedasticity, we have a new $\boldsymbol{\Sigma}$:

$$E[\varepsilon_i\, \varepsilon_j|\mathbf{X}] = \sigma^2 \quad \text{if } i = j$$
$$= \sigma_{ij} \quad \text{if } i \neq j$$

Auto-correlation

• In general, we find autocorrelation (or serial correlation) in time series, shocks are persistent over time: It takes time to absorb a shock.

• The shocks can also be correlated over the cross-section, causing cross-correlation. For example, if an unexpected new tax is imposed on the technology sector, all the companies in the sector are going to share this shock.

• Usually, we model autocorrelation using two model: autoregressive (AR) and moving averages (MA).

• In an AR model, the errors, ε_t, show a correlation over time. In an MA model, the errors, ε_t, are a function (similar to a weighted average) of previous errors, now denoted u_t's.

3

Auto-correlation

Examples:

- First-order autoregressive autocorrelation: AR(1)

$$\varepsilon_t = \rho_1 \varepsilon_{t-1} + u_t$$

- p^{th}-order autoregressive autocorrelation: AR(p)

$$\varepsilon_t = \rho_1 \varepsilon_{t-1} + \rho_2 \varepsilon_{t-2} + \cdots + \rho_p \varepsilon_{t-p} + u_t$$

- Third-order moving average autocorrelation: MA(3)

$$\varepsilon_t = u_t + \lambda_1 u_{t-1} + \lambda_2 u_{t-2} + \lambda_3 u_{t-3}$$

Note: The last example is described as third-order moving average autocorrelation, denoted MA(3), because it depends on the three previous innovations as well as the current one.

4

Auto-correlation – Visual Check

• Plot data, usually residuals from a regression, to see if there is a pattern:

- Positive autocorrelation: A positive (negative) observation tends to be followed by a positive (negative) observation. We tend to see continuation in the series.

- Negative autocorrelation: A positive (negative) observation tends to be followed by a negative (positive) observation. We tend to see reversals.

- No autocorrelation: A positive (negative) observation has the same probability of being followed by a negative or positive (positive or negative) observation. We tend to no pattern.

5

Auto-correlation – Visual Check

Example: I simulate a y_t series:

$$y_t = \rho_1 y_{t-1} + u_t, \qquad \text{with } u_t \sim \text{WN}(0, \sigma^2=1)$$

Three cases:
(1) Positive autocorrelation: $\rho_1 = .70$
(2) Negative autocorrelation: $\rho_1 = -.70$
(3) No correlation: $\rho_1 = -0$

• R code for simulation:

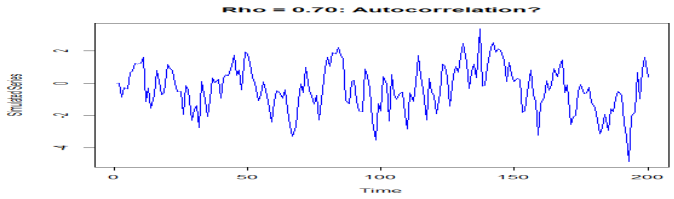
```
T_sim <- 200
u <- rnorm(200)                          # Draw T_sim normally distributed errors
y_sim <- matrix(0,T_sim,1)
rho <- .7                                # Change to create different correlation patterns
a <- 2                                   # Time index for observations
while (a <= T_sim) {
        y_sim[a] = rho * y_sim[a-1] + u[a]   # y_sim simulated autocorrelated values
a <- a + 1
}
plot(y_sim, type="l", col="blue", ylab ="Simulated Series", xlab ="Time")
title("Visual Test: Autocorrelation?")
```

6

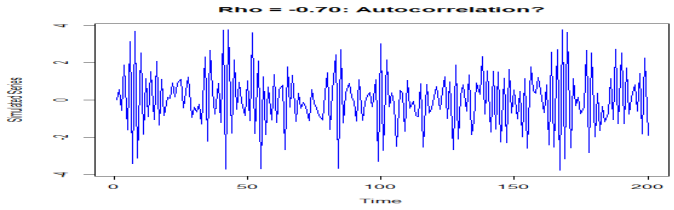
Auto-correlation – Visual Check

Example (continuation):

(1) Positive autocorrelation $\rho_1 = .70$



(2) Negative autocorrelation $\rho_1 = -.70$

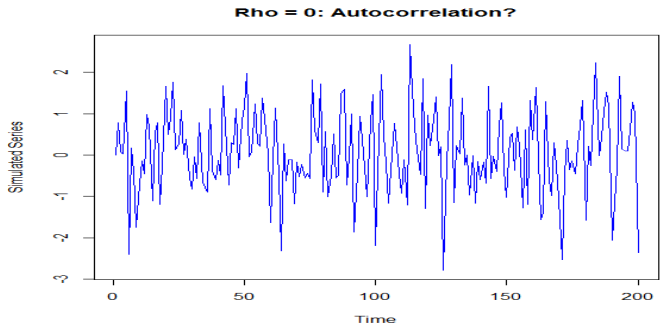


7

Auto-correlation – Visual Check

Example (continuation):

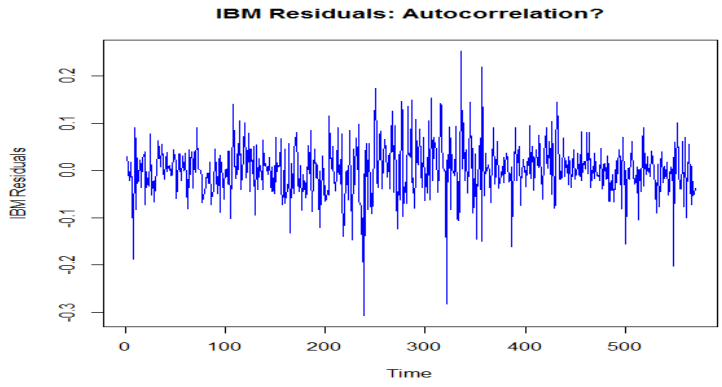
(3) No autocorrelation: $\rho_1 = 0$



8

Auto-correlation – Visual Check: IBM

Example: Residual plot for the 3 factor F-F model for **IBM returns**:

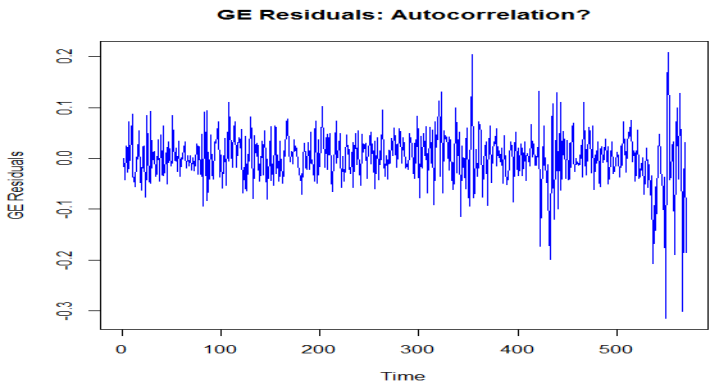


- It looks like a small ρ_1, but not very clear pattern from the graphs.

9

Auto-correlation – Visual Check: GE

Example: Residual plot for the 3 factor F-F model for **GE returns**:

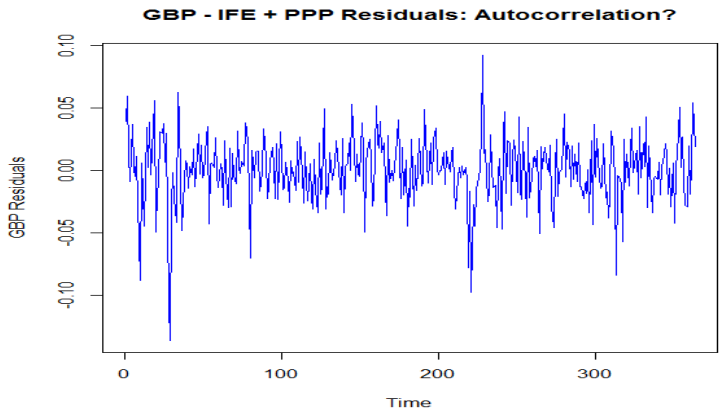


- It looks like a small ρ_1, but not very clear pattern from the graphs.

10

Auto-correlation – Visual Check: GBP

Example: Residual plot for the encompassing model (IFE + PPP) for changes in the **USD/GBP**:



• Again, it looks like a small ρ_1, but not very clear pattern.

11

Implications for OLS

• Similar to the heteroscedasticity results:

- OLS is unbiased, consistent (we need additional assumptions), asymptotic normality (we need additional assumptions and definitions), but inefficient.
- OLS standard errors are incorrect, often biased downwards.

• A very important exception: The lagged dependent variable

$$y_t = \alpha + \beta\, x_t + \gamma\, y_{t-1} + \varepsilon_t, \qquad \varepsilon_t = \rho\, \varepsilon_{t-1} + u_t.$$

Now, $\text{Cov}[y_{t-1}, \varepsilon_t] \neq 0$ $\Rightarrow$ IV Estimation

• Useful strategy: OLS estimates with the Newey-West (NW) SE. Recall NW's HAC estimator of $\mathbf{Q}^*$:

$$\mathbf{S}_0 = (1/T) \sum_{i=1}^{T} e_i^2 \boldsymbol{x}_i \boldsymbol{x}_i' \quad \text{– the White estimator.}$$

$$\mathbf{S}_T = \mathbf{S}_0 + (1/T) \sum_{l=1}^{L} k(l) \sum_{t=l+1}^{T} (\mathbf{x}_{t-l} e_{t-l} e_t \mathbf{x}_t' + \mathbf{x}_t e_t e_{t-l} \mathbf{x}_{t-l}')$$

Implications for OLS: Relative Efficiency

• We define relative efficiency of GLS against OLS as:

$$[RE(\beta)]_i = \frac{\left((X'\Omega^{-1}X)^{-1}\right)_i}{\left((X'X)^{-1}X'\Omega X(X'X)^{-1}\right)_i}$$

• Let $y_t = \alpha + \beta x_t + \gamma y_{t-1} + \varepsilon_t$, $\varepsilon_t = \rho \varepsilon_{t-1} + u_t$, $u_t \sim$ WN
Also, let x_t also follow an AR(1) process: $x_t = \theta x_{t-1} + \xi_t$, $\xi_t \sim$ WN

Then, when T is large, it can be shown that

$$RE(\beta) = \frac{Var(\hat{\beta}_{GLS})}{Var(b_{OLS})} \approx \frac{(1-\rho\theta)(1-\rho^2)}{(1+\rho\theta)(1+\rho^2-2\rho\theta)}$$

• The relative efficiency can be very poor for large ρ for any given θ.

Example: Let ρ = θ = 0.7 ⇒ RE ≈ 0.3423.
Suppose SE[β_{GLS}]= 1 ⇒ SE[**b**] = **1.71** (= sqrt[1/0.3423])
⇒ OLS SE is about **71%** > GLS SE.

Implications for OLS: Relative Efficiency

$$RE(\beta) = \frac{Var(\hat{\beta}_{GLS})}{Var(b_{OLS})} \approx \frac{(1-\rho\theta)(1-\rho^2)}{(1+\rho\theta)(1+\rho^2-2\rho\theta)}$$

• The OLS estimators can be quite reasonable for a low degree of autocorrelation for any given θ, for example, when ρ = .3 and θ=.9, then RE ≈ 0.9510.

• The inefficiency of OLS is difficult to generalize. We tend to see increase inefficiency with increasing values of the disturbance variances.

• In practice, it is worst in *low frequency* -i.e., long period (year)- slowly evolving data. Can be extremely bad. GLS vs. OLS, the efficiency ratios can be 3 or more.

• Given the potential efficiency gain, it makes sense to test for autocorrelation.

Newey-West estimator

• The performance of NW estimators depends on the choice of the kernel function –i.e., k_l– and truncation lag (L). These choices affect the resulting test statistics and render testing results fragile.

• NW SEs perform poorly in Monte Carlo simulations: the finite-sample performance of tests using NW SE is not well approximated by the asymptotic theory (big size problems), especially when show $\boldsymbol{x}_t e_t$ moderate or high persistence:

- The kernel weighting scheme yields negative bias –i.e., NW SEs are downward biased–, which could be big in finite samples.

- The tests based on the NW SE usually over-reject H_0.

- A relatively small L is needed to minimize MSE, which leads to considerable bias of the $\mathbf{Q}^*$ estimator (&, then, distorts the size the tests). Minimizing size distortions needs a larger L.

Newey-West estimator: Implementation

• To implement the HAC estimator, we need to determine: lag order – i.e., truncation lag (L) or bandwidth–, and kernel choice ($k_l(L)$).

(1) Truncation lag (L)

No optimal formula; though selecting L to minimize MSE is popular.

To determine L, we use:

- Trial and error, informed guess.
- Rules of thumb. For example: $L = 0.75T^{1/3} - 1$.
- Automatic selection rules, following Andrews (1991), Newey and West (1994) or Sun *et al.* (2008).

The choice of L matters. In general, for ARMA models we have:

- Shorter lags: Larger Bias, Smaller Variance
- Longer lags: Smaller Bias, Larger Variance

Newey-West estimator: Implementation

• Usual practical advise regarding L : Choose L (lags) a little longer than you might otherwise.

• Sun *et al.* (2008) give some intuition for a longer L than the optimal MSE L, by expanding the probability of a test. Simple example:

Let $z \sim N(0, \sigma^2)$, s^2 is an estimator of σ^2 (assume independent of z).

$$\Pr\left(\frac{z^2}{s^2} < c\right) = \Pr\left(z^2 < s^2 c\right) = E\left(\mathrm{I}[z^2 < s^2 c]\right) = E\left(\mathrm{g}(s^2)\right)$$

$$\approx E\left(\mathrm{g}(\sigma^2)\right) + E\left(s^2 - \sigma^2\right) \times \mathrm{g}'(\sigma^2) + \frac{1}{2} E\left((s^2 - \sigma^2)^2\right) \times \mathrm{g}''(\sigma^2)$$

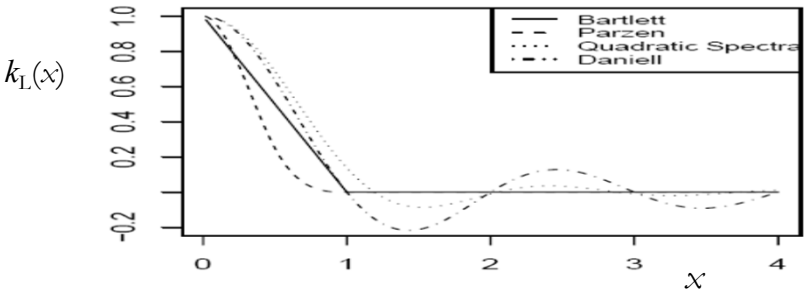
$$= F_{\chi_1^2}(c) + \mathrm{Bias}(s^2) \times \mathrm{g}'(\sigma^2) + \frac{1}{2} \mathrm{MSE}\left(s^2\right) \times \mathrm{g}''(\sigma^2)$$

$\Rightarrow$ Equal weight MSE + Bias. Long L minimizes the Bias; better size!

Newey-West estimator: Implementation

(2) Kernel Choice

- In theory, the kernel choice matters.
- In practice, at least for psd kernels, it does not seem to matter.



• Based on the work of Andrews (1991), where he finds a HAC that minimizes the AMSE of the LRV, the QS kernel tends to be the default kernel in computations of HAC SE.

NW Estimator: Improvements

• Other than finding a good kernel and a (long) L, the performance of HAC estimators may be improved by:

(1) Pre-whitening the data -Andrews and Monahan (1992). Regress $\mathbf{x}_i e_t$ on its lagged values. Some arbitrary choice in the selection of the lag order to do the regression.

(2) Use forecast errors –Kuan and Hsieh (2006). Computing sample autocovariances based on forecast errors, instead of OLS residuals. Replace e_t with one-step-ahead forecast errors: f$e_t = y_t - \mathbf{X}_t' \, \boldsymbol{b}_{t-1}$, where $\boldsymbol{b}_{t-1}$ is the recursive OLS estimators based on the subsample of the first $t-1$ observations.

NW Estimator: Improvements - Example

Example: We compute different NW SE for the 3 factor F-F model for IBM returns, with bandwidth selected as in Andrews (1991):

```
> library(sandwich)
> reg <- lm(y ~ x -1)
> reg$coefficients
      x          xx1                      xx2          xx3
-0.2331470817  0.0101872239  0.0009802843 -0.0044459013          ⟹ OLS b

> sqrt(diag(kernHAC(reg, prewhite = 0, bw = bwAndrews, kernel = "Quadratic Spectral", verbose = TRUE)))
x         xx1         xx2         xx3
0.020959375 0.002848645 0.003983330 0.005310548                  ⟹ & Bandwidth chosen: 3.035697

> sqrt(diag(kernHAC(reg, prewhite = 0, bw = bwAndrews, kernel = "Bartlett", verbose = TRUE)))
x         xx1         xx2         xx3
0.020344074 0.002828663 0.003995942 0.005177482                  ⟹ & Bandwidth chosen: 3.507051

> sqrt(diag(kernHAC(reg, prewhite = 0, bw = bwAndrews, kernel = "Parzen", verbose = TRUE)))
x         xx1         xx2         xx3
0.022849506 0.002839034 0.003954436 0.005427730                  ⟹ & Bandwidth chosen: 6.110888
```

NW Estimator: Improvements - Example

Example: Now, we also pre-white the data (**prewhite** = **1**):

```
> sqrt(diag(kernHAC(reg, prewhite = 1, bw = bwAndrews, kernel = "Quadratic Spectral", verbose = TRUE)))
x        xx1        xx2        xx3
0.043339699 0.002908898 0.004029606 0.005783013          ⇒ & Bandwidth chosen: 0.8118876

> sqrt(diag(kernHAC(reg, prewhite = 1, bw = bwAndrews, kernel = "Bartlett", verbose = TRUE)))
x        xx1        xx2        xx3
0.042943572 0.002912273 0.004022336 0.005786720          ⇒ & Bandwidth chosen: 0.516233

> sqrt(diag(kernHAC(reg, prewhite = 1, bw = bwAndrews, kernel = "Parzen", verbose = TRUE)))
x        xx1        xx2        xx3
0.040963950 0.002912789 0.004006919 0.005767432          ⇒ & Bandwidth chosen: 1.634337
```

• Note: Pre-whitening tends to increase the standard errors (& decrease the bandwidth). Nice result, given that the usual NW SEs tend to be downward biased.

Newey-West estimator: Inconsistency

• Recall that a key assumption in establishing consistency for $\mathbf{S}_T$ is that $L \to \infty$ as $T \to \infty$, but $L/T \to 0$.

• In practice, L/T is never equal to 0, but some positive fraction, b ($b \in (0,1]$). Under this situation, the NW estimator is no longer consistent.

• Thus, t- and F-tests no longer converge in distribution to Normal and χ^2 RVs, but they do converge in distribution to a RV that have non-standard distribution; which do not depend on the unknown value of $\mathbf{\Omega}$. Tests are still possible.

• To get asymptotic distributions (& critical values) we use "*fixed-b*" asymptotics. Under fixed-b asymptotics, the truncation parameter, L, is treated as proportional to T, so $L = bT$, where b is fixed –see, Kiefer, Vogelsang & Bunzell (KVB, 2000), Kiefer & Vogelsang (2002, 2005).

Newey-West estimator: Inconsistency

• Under fixed-b asymptotics, typically $\mathbf{S}_T \rightarrow \mathbf{Q}^{*1/2}\ \Xi\ \mathbf{Q}^{*1/2}$, where Ξ is a RV with $E[\Xi]=\mathbf{I}_p$. Ξ has a non-standard distribution.

• Kiefer and Vogelsang (2005) derive limiting distribution for $\mathbf{S}_T$, which is complicated, but the 95% critical values (CV) for t-tests can be constructed using the following polynomial ($b = L/T$):

$$CV\ (L/T) = 1.96 + 2.9694\ b + 0.416\ b^2 - .05324\ b^3.$$

Note: As $b \rightarrow 0$, the standard t critical values apply.

• Since non-standard distributions are not popular, work has been devoted to find simple and intuitive estimators of $\mathbf{Q}^*$ that can be used in tests with traditional distributions (say, N(0, 1) and χ^2).

Newey-West estimator: Inconsistency

• When the frequency domain kernel weights are equal and truncated after the first $B/2$ *periodogram ordinates* (an estimator of the spectrum at frequency $(2\pi j/T)$), the limiting fixed-b distribution of $\mathbf{S}_T$ is a χ^2_B/B.

• This corresponds to the equal-weighted periodogram estimator of $\mathbf{Q}^*$ (the Daniell window):

$$S_T^{EWP} = \frac{2\pi}{B}\sum_{j=1}^{B/2} I_{xeex}(2\pi j/T) = \frac{1}{B}\sum_{j=1}^{B/2}\left(\frac{1}{\sqrt{T}}\sum_{t=1}^{T}(x_t e_t)e^{-i\omega t}\right)\left(\frac{1}{\sqrt{T}}\sum_{t=1}^{T}(x_t e_t)e^{-i\omega t}\right)^T$$

Now, the usual t-test, $t^{EWP} = \sqrt{T}(\hat{\beta}-\beta_0)/\sqrt{S_T^{EWP}}$, has a t_B asymptotic distribution under H_0.

• The EWP estimator has the nice property that fixed-b asymptotic inference can be conducted using standard t and F distributions.

Newey-West estimator: Inconsistency

• In addition, the EWP estimator is psd with probability 1.

• Müller (2007), and Sun (2013) note that other estimators of $\mathbf{Q}^*$ can be derived by replacing the Fourier functions in S^{EWP}_T by other basis functions of a general orthonormal set of basis function for $L^2[0,1]$.

• Then, we can see S^{EWP}_T as a especial case of:

$$S_T^{BF} = \frac{1}{B}\sum_{j=1}^{B} S_j, \quad \text{where} \quad S_j = \hat{\Lambda}_j' \hat{\Lambda}_j \quad \& \quad \hat{\Lambda}_j = \frac{1}{\sqrt{T}}\sum_{t=1}^{T}(x_t e_t)\phi_j(t/T)$$

• Different ϕ_j basis functions (say, cosine), different estimators.

Note: Since S^{BF}_T is computed using an outer product, it is psd.

Newey-West estimator: KVB

• The (kernel) HAC estimation requires the choices of the kernel function and L. Such choices are somewhat arbitrary in practice.

• To avoid these difficulties, Kiefer, Vogelsang, and Bunzel (2000), KVB, proposed an approach that yields an asymptotically pivotal test without consistent estimation of the asymptotic covariance matrix.

• Idea: Use a normalizing matrix to eliminate the nuisance parameters in $\mathbf{Q}^{*1/2}$, the matrix square root of $\mathbf{Q}^*_T$ & impose no truncation (b=1). Let

$$\boldsymbol{\varphi}_j = \frac{1}{\sqrt{T}}\sum_{j=1}^{T} \boldsymbol{x}_j e_j \qquad \text{-a } k\text{x1 vector}$$

Normalizing matrix:

$$\mathbf{C}_T = \frac{1}{T}\sum_{j=1}^{T} \boldsymbol{\varphi}_j \boldsymbol{\varphi}_j' = \frac{1}{T^2}\sum_{j=1}^{T}\left(\sum_{j=1}^{T} \boldsymbol{x}_j e_j\right)\left(\sum_{j=1}^{T} \boldsymbol{x}_j e_j\right)'$$

Newey-West estimator: KVB

• Normalizing matrix:

$$\mathbf{C}_T = \frac{1}{T}\sum_{j=1}^{T} \boldsymbol{\varphi}_j \, \boldsymbol{\varphi}_j' = \frac{1}{T^2} \sum_{j=1}^{T} \left(\sum_{j=1}^{T} \boldsymbol{x}_j e_j\right) \left(\sum_{j=1}^{T} \boldsymbol{x}_j e_j\right)'$$

This normalizing matrix is inconsistent for $\mathbf{Q}^*_T$ but it is free from the choice of kernel and L. (Note: There is no truncation, $L=T \Rightarrow$ Good for size of test!)

• We use this $\mathbf{C}_T$ matrix to calculate tests. For example, to test J restrictions H_0: ($\mathbf{R}\,\boldsymbol{\beta} - \mathbf{q}=\mathbf{0}$), we have the following statistic

$$\boldsymbol{W}_T^{\dagger} = T\,(\mathbf{R}\,\mathbf{b}_T - \mathbf{q})'[\mathbf{R}\,(\mathbf{X'X})^{-1}\,\mathbf{C}_T\,(\mathbf{X'X})^{-1}\,\mathbf{R}]^{-1}\,(\mathbf{R}\,\mathbf{b}_T - \mathbf{q}).$$

Although the asymptotic distribution of W^+_T is non-standard, it can be simulated -Lobato (2001).

Newey-West estimator: KVB

• KV (2002) showed that $2\mathbf{C}_T$ is algebraically equivalent to $\boldsymbol{Q}_T^{*,B}$ (where B stands for Bartlett kernel) without truncation (b=1) - i.e., $L(T)=T$. Then, usual W based on $\boldsymbol{Q}_T^{*,B}$ without truncation is the same as $\boldsymbol{W}_T^{\dagger}/2$.

• KVB derive the (non-standard) asymptotic distribution of the conventional t-test of H_0: $\beta_i = \beta_i^0$; but using their robust version, $t^{\dagger}$:

$$t^{\dagger} = \frac{\sqrt{T}(b_{i,T} - \beta_i^0)}{\sqrt{\hat{\delta}_i}} \xrightarrow{d} \frac{W(1)}{\int_0^1 B(r)^2 dr]^2}.$$

where δ_i is the i-th diagonal element of $(\mathbf{X'X})^{-1}\,\mathbf{C}_T\,(\mathbf{X'X})^{-1}$, W is a standard Wiener process, and $B(r)$ is a Brownian Bridge –i.e., $B(r) = W(r) - r\,W(1)$, $0 \leq r \leq 1$. This distribution is symmetric, but more disperse than the N(0,1).

Newey-West estimator: KVB

• KVB report the quantiles of the asymptotic distribution of the usual t-test, using $\mathbf{C}_T$ and using the NW SE, without truncation. (Notation: $\boldsymbol{Q}_T^{*,kernel} = \boldsymbol{\Sigma}_T^{kernel}$)

Table 1: The quantiles of the $t^\dagger$ and t tests based on $\widehat{\Sigma}_T^\kappa$ without truncation.

prob.	90%	95%	97.5%	99%
$t^\dagger$ with $\widehat{C}_T$	3.890	5.374	6.811	8.544
t with $\widehat{\Sigma}_T^B$	2.740	3.764	4.771	6.090
t with $\widehat{\Sigma}_T^P$	2.840	4.228	5.671	8.112
t with $\widehat{\Sigma}_T^{QS}$	5.188	8.283	12.374	20.380
t with $\widehat{\Sigma}_T^D$	4.822	7.711	11.573	19.180

Remark: KV (2002) shows that under certain assumptions the *t-test* with NW's SE without truncation are also asymptotically pivotal.

Newey-West estimator: KVB - Remarks

• An advantage of testing with KVB's $\mathbf{C}_T$ matrix is that its asymptotic distribution usually provides good approximation to its finite-sample counterpart. That is, the empirical size is close to the nominal size (α).

• This is not the case for the NW HAC SE: in finite samples, they are downward biased. Tests are usually over-sized –i.e., not conservative.

• KV (2002b) show that, for $\boldsymbol{Q}_T^{*,k}$ with the truncation lag equal to sample size, T, $\boldsymbol{Q}_T^{*,B}$ compares favorably with $\boldsymbol{Q}_T^{*,QS}$ in terms of power. This is in contrast with the result in HAC estimation, where the latter is usually preferred to other kernels.

• Reference: Kiefer, N. M., T. J. Vogelsang and H. Bunzel (2000). "Simple robust testing of regression hypothesis," *Econometrica*, **68**, 695–714.

Testing for Autocorrelation: LM Test

• There are several autocorrelation tests. Under the null hypothesis of no autocorrelation of order p, we have H_0: $\rho_1 = ... = \rho_p = 0$.
Under H_1, we consider: $\varepsilon_t = \rho_1\varepsilon_{t-1} + \rho_2\varepsilon_{t-2} + \cdots + \rho_p\varepsilon_{t-p} + u_t$

Under H_0, we can use OLS residuals.

• Breusch–Godfrey (1978) LM test. Similar to the BP test:

– **Step 1**. (Same as BP's Step 1). Run OLS on DGP:

$\mathbf{y} = \mathbf{X}\,\beta + \boldsymbol{\varepsilon}$. - Keep residuals, e_t.

– **Step 2**. (Auxiliary Regression). Run the regression of e_t on all the explanatory variables, **X**:

$e_t = \boldsymbol{x}_t'\,\boldsymbol{\gamma} + \alpha_1\, e_{t-1} + ... + \alpha_p\, e_{t-p} + v_t$ - Keep R^2 (R_e^2)

– **Step 3**. Keep R_e^2. Then, calculate:

LM = $(T\text{-}p) * R_e^2 \xrightarrow{d} \chi_p^2$.

31

Testing for Autocorrelation: LM Test

Example: LM-AR Test for the 3 factor F-F model for IBM returns (p=12 lags):

```
fit_ibm<- lm(ibm_x ~ Mkt_RF + SMB + HML)        # OLS regression
e <- fit_ibm$residuals                          # OLS residuals
p_lag <- 12                                     # Select # of lags for test (set p)
e_lag <- matrix(0,T-p_lag,p_lag)                # Matrix to collect lagged residuals
a <- 1
while (a<=p_lag) {                              # Do loop creates matrix (e_lag) with lagged e
  za <- e[a:(T-p_lag+a-1)]
   e_lag[,a] <- za
a <- a+1
}

Mkt_RF_p <- Mkt_RF[(p_lag+1):T]                 # Adjust for new sample size: T – p_lag
SMB_p <- SMB[(p_lag+1):T]
HML_p <- HML[(p_lag+1):T]
fit1 <- lm(e[(p_lag+1):T] ~ e_lag + Mkt_RF_p + SMB_p + HML_p)   # Auxiliary Regression
r2_e1 <- summary(fit1)$r.squared                # get R^2 from Auxiliary Regression
lm_t <- (T-p_lag )* r2_e1                       # LM-test wih p lags
```

32

Testing for Autocorrelation: LM Test

Example (continuation):

```
lm_t                                  # print lm_t
df <- ncol(e_lag)                     # degrees of freedom of test
1 - pchisq(lm_t,df)                   # p-value of lm_t
> r2_e1 <- summary(fit1)$r.squared
> r2_e1
[1] 0.0303721
> (T-p_lag)
[1] 557
> lm_t <- (T - p_lag) * r2_e1
> lm_t
[1] 16.91726
> df <- ncol(e_lag)                   # degrees of freedom for the LM Test
> 1-pchisq(lm_t,df)
[1] 0.1560063
```

LM-AR(12) Test: **16.91726** ⇒ cannot reject H_0 at 5% level (*p-value* > .05).

If I run the test with p=4 lags, I get

LM-AR(4) Test: **2.9747** (*p-value* = 0.56) ⇒ cannot reject H_0 at 5% level (*p-value* > .05).

33

Testing for Autocorrelation: LM Test

Example (continuation):

The package *lmtest,* performs this test, *bgtest,* (and many others, used in this class, encompassing, jtest, waldtest, etc). You need to install it first: install.packages("lmtest"), then call the library(lmtest).

```
> library(lmtest)
> bgtest(ibm_x ~ Mkt_RF + SMB + HML, order=12)

        Breusch-Godfrey test for serial correlation of order up to
        12

data:  lr_ibm ~ Mkt_RF + SMB + HML
LM test = 16.259, df = 12, p-value = 0.1797
```

(minor difference with the previous test, likely due to multiplication by *T*. Results do not change much)

Note: If you do not include in the Auxiliary Regression the original regressors (Mkt_RF, SMB, HML) the test do not change much. You get LM-AR(12) Test: **16.83253** ⇒ very similar. Not entirely correct, but it works well

34

Testing for Autocorrelation: LM Test

Example (continuation):

Autocorrelation is very common. If I run the test for Disney, CNP, or GE, instead, we get significant test results. For DIS:

```
lr_dis <- log(x_dis[-1]/x_dis[-T])
dis_x <- lr_dis – RF

> bgtest(dis_x ~ Mkt_RF + SMB + HML, order=4)
      Breusch-Godfrey test for serial correlation of order up to  4

data:  dis_x ~ Mkt_RF + SMB + HML
LM test = 8.6382, df = 4, p-value = 0.07081
```

⇒ cannot reject H_0 at 5% level (*p-value* >.05)

```
> bgtest(dis_x ~ Mkt_RF + SMB + HML, order=12)
      Breusch-Godfrey test for serial correlation of order up to  12

data:  dis_x ~ Mkt_RF + SMB + HML
LM test = 30.068, df = 12, p-value = 0.002728
```

⇒ reject H_0 at 5% level (*p-value* < .05)

35

Testing for Autocorrelation: LM Test

Example (continuation):

LM tests for autocorrelation (with 12 lags) for GE and CNP again show significant test results:

```
lr_ge <- log(x_ge[-1]/x_ge[-T]); ge_x <- lr_ge – RF
lr_cnp <- log(x_cnp[-1]/x_cnp[-T]); cnp_x <- lr_cnp – RF

> bgtest(ge_x ~ Mkt_RF + SMB + HML, order=4)
      Breusch-Godfrey test for serial correlation of order up to  4

data:  ge_x ~ Mkt_RF + SMB + HML
LM test = 28.257, df = 4, p-value = 0.005073
```

⇒ cannot reject H_0 at 5% level (*p-value* >.05)

```
> bgtest(cnp_x ~ Mkt_RF + SMB + HML, order=12)
      Breusch-Godfrey test for serial correlation of order up to  12

data:  cnp_x ~ Mkt_RF + SMB + HML
LM test = 31.718, df = 12, p-value = 0.00153
```

⇒ reject H_0 at 5% level (*p-value* < .05)

36

Testing for Autocorrelation: LM Test

• Q: How many lags are needed in the test? In general, enough to make sure there is no auto-correlation left in the residuals. Using some criteria for optimal ("*automatic*") selection is possible.

• There are some popular rule of thumbs: for daily data, 5 or 20 lags; for weekly, 4 or 12 lags; for monthly data, 12 lags; for quarterly data, 4 lags.

37

Testing for Autocorrelation: Durbin-Watson

• The Durbin-Watson (1950) (DW) test for AR(1) autocorrelation: H_0: $\rho_1 = 0$ against H_1: $\rho_1 \neq 0$. Based on simple correlations of $\boldsymbol{e}$.

$$d = \frac{\sum_{t=2}^{T}(e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2}$$

• It is easy to show that when $T \rightarrow \infty$, $d \approx 2(1 - \rho_1)$.

• ρ_1 is estimated by the sample correlation r.

• Under H_0, $\rho_1=0$. Then, d should be distributed randomly around 2.

• Small values of d lead to rejection of H_0. The distribution depends on **X**. Durbin-Watson derived bounds for the test.

• In the presence of lagged dependent variables, Durbin's (1970) h test should be used: $h = r\,\text{sqrt}\{T/(1 - T\,s^2)\}$

Testing for Autocorrelation: DW Test

Example: DW Test for the 3 factor F-F model for IBM returns

```
fit_dw <- lm(ibm_x ~ Mkt_RF + SMB + HML)        # OLS regression
e <- fit_dw$residuals                           # OLS residuals
> RSS <- t(e)%*%e                               # RSS
> DW <- sum((e[1:(T-1)]-e[2:T])^2)/RSS          # DW stat
> DW
[1] 2.042728                  ⇒ DW statistic ≈2 ⇒ No evidence for autocorrelation of order 1.
> 2*(1-cor(e[1:(T-1)],e[2:T]))                  # approximate DW stat
[1] 2.048281
```

- Similar finding for Disney returns:

```
> DW
      [,1]
[1,] 2.1609          ⇒ DW statistic ≈2 ⇒ But, DIS suffers from autocorrelation!
```

⇒ This is why DW are not that informative. They only test for AR(1) in residuals.

Note: The package *lmtest* performs this test too, *dwtest*:

```
> dwtest(y ~ Mkt_RF + SMB + HML)
DW = 2.0427, p-value = 0.7087
```

39

Testing for Autocorrelation: DW Test

Example: DW Test for the residuals of the encompassing model (IFE + PPP) for changes in USD/GBP:

```
fit_gbp <- lm(lr_usdgbp ~ inf_dif + int_dif)
e_gbp <- fit_gbp$residuals
> dwtest(fit_gbp)

        Durbin-Watson test

data:  fit_gbp
DW = 1.8588, p-value = 0.08037          ⇒ not significant at 5% level.
alternative hypothesis: true autocorrelation is greater than 0
```

40

Testing for Autocorrelation: Portmanteu tests

• Portmanteu tests are tests with a well-defined H_0, but not specific H_1. We will present two: Box-Pierce Q test and the Ljung-Box test.

• Box-Pierce (1970) test (Q test).

It tests H_0: $\rho_1 = ... = \rho_p = 0$ using the sample correlation, $r_j = \frac{\hat{\gamma}_j}{\hat{\gamma}_0}$

where (using time series notation)

$\hat{\gamma}_j$ = Sample covariance between y_t & y_{t-j} = $\frac{\sum_{t=j+1}^{T}(y_t - \bar{y})(y_{t-j} - \bar{y})}{T-j}$

$\hat{\gamma}_0$ = Sample variance.

Then, under H_0:

$$Q = T \sum_{j=1}^{p} r_j^2 \xrightarrow{d} \chi_p^2.$$

41

Testing for Autocorrelation: Portmanteu tests

• Ljung-Box (1978) test (LB test).

A variation of the Box-Pierce test. It has a small sample correction.

$$LB = T * (T + 2) * \sum_{j=1}^{p} \frac{r_j^2}{T-j} \xrightarrow{d} \chi_p^2.$$

• The asymptotic distribution of both tests is based on the fact that, under the null of independent data, $\sqrt{T}\,\boldsymbol{r} \xrightarrow{d} N(0, \mathbf{I})$.

Note: When analyzing residuals, e_t, of a regression we compute r_j as:

$$r_j = \frac{\hat{\gamma}_j}{\hat{\gamma}_0} = \frac{\sum_{j=1}^{T-j} e_t\, e_{t-j}}{\sum_{j=1}^{T} e_j^2}$$

• The LB statistic is widely used. But, the BG (1978) LM tests conditions on **X**. Thus, it is more powerful.

42

Testing for Autocorrelation: Portmanteu tests

Example: Q and LB tests with $p = 12$ lags for the residuals in the 3-factor FF model for **IBM excess returns**:

```
RSS <- sum(e_ibm^2)
r_sum <- 0
lb_sum <- 0
p_lag <- 12
a <- 1
while (a <= p_lag) {
      za <- as.numeric(t(e_ibm[(p_lag+1):T]) %*% e_ibm[a:(T-p_lag+a-1)])
 r_sum <- r_sum + (za/RSS)^2                    #sum cor(e[(p_lag+1):T], e[a:(T-p_lag+a-1)])^2
 lb_sum <- lb_sum + (za/RSS)^2/(T-a)
a <- a + 1
}
Q <- T*r_sum
LB <- T*(T+2)*lb_sum
> Q
[1] 16.39559        (p-value = 0.1737815)        ⇒ cannot reject H0 at 5% level.
> LB
[1] 16.46854        (p-value = 0.1707059)        ⇒ cannot reject H0 at 5% level.
```

43

Testing for Autocorrelation: Portmanteu tests

Example (continuation): The *Box.test* function computes Q & LB:

- Q test

```
> Box.test(e_ibm, lag = 12, type="Box-Pierce")

        Box-Pierce test

data:  e
X-squared = 16.304, df = 12, p-value = 0.1777
```

- LB test

```
> Box.test(e_ibm, lag = 12, type="Ljung-Box")

        Box-Ljung test

data:  e
X-squared = 16.61, df = 12, p-value = 0.1649
```

Note: There is a minor difference between the previous code and the code in Box.test. They are based on how the correlations of e are computed (centered around the mean, or assumed zero mean).

44

Testing for Autocorrelation: Portmanteu tests

Example (continuation): Same tests (p = 12 lags) & same model:

• For **DIS** (dis_x), we get:

> Q

[1] **28.76842** (*p-value* = 0.004264043) ⇒ reject H_0 at 5% level.

> LB

[1] **29.05072** (*p-value* = 0.003872236) ⇒ reject H_0 at 5% level.

• For GE (ge_x), we get

> Q

[1] **24.20958** (*p-value* = 0.01904602) ⇒ reject H_0 at 5% level.

> LB

[1] **24.33922** (*p-value* = 0.01828389) ⇒ reject H_0 at 5% level.

• Autocorrelation in financial asset returns is a usual finding in monthly, weekly and daily data.

45

Testing for Autocorrelation: Portmanteu tests

• Q & LB tests are widely use, but they have two main limitations:

(1) The test was developed under the independence assumption.

If y_t shows dependence, such as heteroscedasticity, the asymptotic variance of $\sqrt{T}\,\boldsymbol{r}$ is no longer **I**, but a non-diagonal matrix.

There are several proposals to "*robustify*" both Q & LB tests, see Diebold (1986), Robinson (1991), Lobato et al. (2001). The "robustified" Portmanteau statistic uses $\tilde{r}_j$ instead of r_j:

$$\tilde{r}_j = \frac{\hat{\gamma}_j^2}{\tau_j} = \frac{\sum_{t=j+1}^{T}(y_t - \bar{y})(y_{t-j} - \bar{y})}{\sum_{t=j+1}^{T}(y_t - \bar{y})^2\ (y_{t-j} - \bar{y})^2}$$

Thus, for Q we have:

$$Q^* = T \sum_{j=1}^{p} \tilde{r}_j^2 \xrightarrow{d} \chi_p^2.$$

46

Testing for Autocorrelation: Portmanteu tests

(2) The selection of the number of autocorrelations p is arbitrary.

The traditional approach is to try different p values, say 3, 6 & 12. Another popular approach is to let the data "select" p, for example, using AIC or BIC, an approach sometimes referred as "*automatic selection*."

Escanciano and Lobato (2009) propose combining BIC's and AIC's penalties to select p in Q* (BIC for small ρ and AIC for bigger ρ).

- It is common to reach different conclusion from Q and Q*.

47

Testing for Autocorrelation: Portmanteu tests

Example: Q* tests with automatic selection of **p** for the residuals in the 3-factor FF model for **IBM & DIS excess returns.** We use Auto.Q funcition in R package *vrtest*.

- For **IBM** (e_ibm), we get:

```
> library(vrtest)
> Auto.Q(e_ibm, 12)            #Maximum potential lag = 12
> $Stat
[1] 0.2781782

$Pvalue
[1] 0.5978978
```

- For **DIS** (e_dis), we get:

```
> Auto.Q(e_dis, 12)
$Stat
[1] 2.649553

$Pvalue
[1] 0.103579
```

⇒ Reversal for DIS

48

GLS: The AR(1) Model

• (**A1**) holds: $\boldsymbol{y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}$

But, $\boldsymbol{\varepsilon}$ is no longer white noise:

$\varepsilon_t = \rho\,\varepsilon_{t-1} + u_t$, $|\rho| < 1$. $\quad u_t \sim$ WN $(0, \sigma_u^2)$

Note: This characterizes the disturbances, not the regressors.

Notation: Let L be the lag operator, such that $L^q\, z_t = z_{t-q}$. Then,

$$(1 - \rho L)\,\varepsilon_t = u_t.$$

• We will get Var$[\varepsilon_t]$ in a different way. After some algebra, we get

$$\varepsilon_t = u_t + \rho\, u_{t-1} + \rho^2\, u_{t-2} + \rho^3\, u_{t-3} + \ldots$$
$$= \sum_{j=0}^{T} \rho^j\, u_{t-j} = \sum_{j=0}^{T} (\rho L)^j\, u_t \quad \text{(a moving average)}$$

• $\text{Var}[\varepsilon_t] = \sum_{j=0}^{T} \rho^{2j}\, Var[u_{t-j}] = \sigma_u^2/(1-\rho^2)$

GLS: AR(1) Case – Autocorrelation Matrix Σ

• Now, we get (**A3'**) $\boldsymbol{\Sigma} = \sigma^2\,\boldsymbol{\Omega}$.

$$(\textbf{A3'}) \quad \sigma^2\Omega = \left(\frac{\sigma_u^2}{1-\rho^2}\right)\begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{T-1} \\ \rho & 1 & \rho & \cdots & \rho^{T-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{T-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{T-1} & \rho^{T-2} & \rho^{T-3} & \cdots & 1 \end{bmatrix}$$

1. Then, we can get the transformation matrix $\mathbf{P} = \boldsymbol{\Omega}^{-1/2}$:

$$\Omega^{-1/2} = \begin{bmatrix} \sqrt{1-\rho^2} & 0 & 0 & \ldots & 0 \\ -\rho & 1 & 0 & \ldots & 0 \\ 0 & -\rho & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & -\rho & 0 \end{bmatrix}$$

GLS: AR(1) Case – Transformed y & X: y^* & X*

2. With $\mathbf{P} = \mathbf{\Omega}^{-1/2}$, we transform the data to do GLS.

$$\mathbf{P} = \mathbf{\Omega}^{-1/2} = \begin{bmatrix} \sqrt{1-\rho^2} & 0 & 0 & \dots & 0 \\ -\rho & 1 & 0 & \dots & 0 \\ 0 & -\rho & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & -\rho & 0 \end{bmatrix}$$

$$\boldsymbol{y}^* = \mathbf{P}\,\boldsymbol{y} = \begin{pmatrix} \left(\sqrt{1-\rho^2}\right) y_1 \\ y_2 - \rho y_1 \\ y_3 - \rho y_2 \\ \dots \\ y_T - \rho_{T-1} \end{pmatrix} \quad \Rightarrow \text{GLS: Transformed } \mathbf{y}^*.$$

GLS: AR(1) Case – Transformed y & X: y* & X*

2. Transformed $\mathbf{x}_k$ column (independent variable k) of matrix $\mathbf{X}$ is:

$$\mathbf{P} = \mathbf{\Omega}^{-1/2} = \begin{bmatrix} \sqrt{1-\rho^2} & 0 & 0 & \dots & 0 \\ -\rho & 1 & 0 & \dots & 0 \\ 0 & -\rho & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & -\rho & 0 \end{bmatrix}$$

$$\boldsymbol{x}_k^* = \mathbf{P}\,\boldsymbol{x}_k = \begin{pmatrix} \left(\sqrt{1-\rho^2}\right) x_{k1} \\ x_{k2} - \rho\, x_{k1} \\ x_{k3} - \rho\, x_{k2} \\ \dots \\ x_T - \rho\, x_{T-1} \end{pmatrix} \quad \Rightarrow \text{GLS: Transformed } \mathbf{X}^*.$$

3. GLS is done with transformed data. In (**A3'**) we assume ρ known.

GLS: The Autoregressive Transformation

• With AR models, sometimes it is easier to transform the data by taking *pseudo differences*.

• For the AR(1) model, we multiply the DGP by ρ and subtract it from it. That is,

$$y_t = \boldsymbol{x}_t'\boldsymbol{\beta} + \varepsilon_t, \qquad \varepsilon_t = \rho\varepsilon_{t-1} + u_t$$
$$\rho y_{t-1} = \rho\boldsymbol{x}_{t-1}'\boldsymbol{\beta} + \rho\varepsilon_{t-1}$$
$$y_t - \rho y_{t-1} = (\boldsymbol{x}_t - \rho\boldsymbol{x}_{t-1})'\boldsymbol{\beta} + (\varepsilon_t - \rho\varepsilon_{t-1})$$
$$y_t^* = \boldsymbol{x}_t^{*\prime}\boldsymbol{\beta} + u_t$$

Now, the errors, u_t, which are uncorrelated. We can do OLS with the pseudo differences.

Note: $y_t^* = y_t - \rho y_{t-1}$ & $\boldsymbol{x}_t^* = \boldsymbol{x}_t - \rho\boldsymbol{x}_{t-1}$ are *pseudo differences*.

FGLS: Unknown Ω

• The problem with GLS is that $\boldsymbol{\Omega}$ is unknown. For example, in the AR(1) case, ρ is unknown.

• Solution: Estimate $\boldsymbol{\Omega}$. ⇒ *Feasible GLS* (FGLS).

• In general, there are two approaches for GLS

(1) Two-step, or *Feasible estimation*: - First, estimate $\boldsymbol{\Omega}$ first.
- Second, do GLS.

Similar logic to HAC procedures: We do not need to estimate $\boldsymbol{\Omega}$, difficult with T observations. We estimate $(1/T)\mathbf{X}'\boldsymbol{\Omega}^{-1}\mathbf{X}$.

– Nice asymptotic properties for FGLS estimator. Not longer BLUE

(2) ML estimation of $\boldsymbol{\beta}$, σ^2, and $\boldsymbol{\Omega}$ at the same time (joint estimation of all parameters). With some exceptions, rare in practice.

FGLS: Specification of Ω

• $\boldsymbol{\Omega}$ must be specified first.

• $\boldsymbol{\Omega}$ is generally specified (modeled) in terms of a few parameters. Thus, $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{\theta})$ for some small parameter vector $\boldsymbol{\theta}$. Then, we need to estimate $\boldsymbol{\theta}$.

Example: ε_i with AR(1) process. We have already derived $\sigma^2 \boldsymbol{\Omega}$ as a function of ρ.

Technical note: To achieve full efficiency, we do not need an *efficient* estimate of the parameters in $\boldsymbol{\Omega}$, only a consistent one.

• For the AR(1) case, there is a simple estimation technique, the Cochrane-Orcutt method.

FGLS Estimation: Cochrane-Orcutt

• $y_t - \rho y_{t-1} = (\boldsymbol{X}_t - \rho \boldsymbol{X}_{t-1})' \beta + \varepsilon_t - \rho \varepsilon_{t-1}$

$\Rightarrow y_t = \rho y_{t-1} + \boldsymbol{X}_t' \beta - \boldsymbol{X}_{t-1}' \rho\beta + u_t$

• We have a linear model, but it is nonlinear in parameters. This is not a problem: Non-linear estimation is possible.

• Before today's computer power, Cochrane–Orcutt's (1949) iterative procedure was an ingenious way to do NLLS. Steps:

(1) Do OLS. Get residuals, $\boldsymbol{e}$. Then estimate ρ with a regression of $\boldsymbol{e}_t$ against $\boldsymbol{e}_{t-1}$. We use *r* to denote the estimator of ρ.

(2) FGLS Step. Using *r* transform the model to get $\boldsymbol{y}^*$ and $\mathbf{X}^*$. Do OLS $\Rightarrow$ get $\mathbf{b}$ to estimate β. Get residuals, $\boldsymbol{e}^*$. Go back to (1).

(3) Iterate until convergence.

FGLS Estimation: Cochrane-Orcutt in R

Example: Cochrane-Orcutt in R

```
# C.O. funcition requires Y, X (with constant), OLS b.
c.o.proc <- function(Y,X,b_0,tol){
 T <- length(Y)
 e <- Y - X%*%b_0                    # OLS residuals
rss <- sum(e^2)                      # Initial RSS of model, RSS_0
rss_1 <- rss                         # RSS_1 will be used to reset RSS after each iteration
d_rss = rss                          # initialize d_rss: difference between RSS_i & RSS_i-1
 e2 <- e[-1]                         # adjust sample size for e_t
 e3 <- e[-T]                         # adjust sample size for e_t-1
 ols_e0 <- lm(e2 ~ e3 - 1)           # OLS to estimate rho
 rho <- ols_e0$coeff[1]              # initial value for rho, ρ_0
i<-1
while (d_rss > tol) {                # tolerance of do loop. Stop when diff in RSS < tol
rss <- rss_1                         # RSS at iter (i-1)
  YY <- Y[2:T] - rho * Y[1:(T-1)]    # pseudo-diff Y
  XX <- X[2:T, ] - rho * X[1:(T-1), ]  # pseudo-diff X
ols_yx <- lm(YY ~ XX - 1)            # adjust if constant included in X
```

FGLS Estimation: Cochrane-Orcutt in R

Example (continuation):

```
b <- ols_yx$coef                     # updated OLS b at iteration i
#  b[1] <- b[1]/(1-rho)              # If constant not pseudo-differenced remove tag #
e1 <- Y - X%*%b                      # updated residuals at iteration i
   e2 <- e1[-1]                      # adjust sample size for updated e_t
   e3 <- e1[-T]                      # adjust sample size for updated e_t-1 (lagged e_t)
   ols_e1 <- lm(e2~e3-1)             # updated regression to value for rho at iteration i
   rho <- ols_e1$coeff[1]            # updated value of rho at iteration i, ρ_i
 rss_1 <- sum(e1^2)                  # updated value of RSS at iteration i, RSS_i
 d_rss <- abs(rss_1 - rss)           # diff in RSS (RSS_i - RSS_i-1)
i <- i+1
   }

 result <-list()
 result$Cochrane-Orc.Proc <- summary(ols_yx)
 result$rho.regression <- summary(ols_e1)
 #  result$Corrected.b_1 <- b[1]
 result$Iterations < -i-1
 return(result)
}
```

FGLS Estimation: Cochrane-Orcutt – i_{MX}

Example: In the model for Mexican interest rates (i_{MX}), we suspect an AR(1) in the residuals:

$$i_{MX,t} = \beta_0 + \beta_1 i_{US,t} + \beta_2 e_t + \beta_3 mx\_I_t + \beta_4 mx\_y_t + \varepsilon_t$$
$$\varepsilon_t = \rho\varepsilon_{t-1} + u_t$$

• Cochrane-Orcutt estimation.

```
y <- mx_i_1
T_mx <- length(mx_i_1)
xx_i <- cbind(us_i_1, e_mx, mx_I, mx_y)
x0 <- matrix(1,T_mx,1)
X <- cbind(x0,xx_i)                                  # X matrix
fit_i <- lm(mx_i_1 ~ us_i_1 + e_mx + mx_I + mx_y)
b_i <-fit_i$coefficients                             # extract coefficients from lm
> summary(fit_i)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.04022    0.01506   2.671  0.00834 **
us_i_1       0.85886    0.31211   2.752  0.00661 **
e_mx        -0.01064    0.02130  -0.499  0.61812
mx_I         3.34581    0.19439  17.212  < 2e-16 ***
mx_y        -0.49851    0.73717  -0.676  0.49985
```

FGLS Estimation: Cochrane-Orcutt – i_{MX}

Example (continuation):

```
> c.o.proc(y,X,b,.0001)
$Cochrane.Orcutt.Proc

Call:
lm(formula = YY ~ XX - 1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.69251 -0.02118 -0.01099  0.00538  0.49403

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
XX           0.16639    0.07289   2.283   0.0238 *
XXus_i_1     1.23038    0.76520   1.608   0.1098        ⇒ not longer significant at 5% level.
XXe_mx      -0.00535    0.01073  -0.499   0.6187
XXmx_I       0.41608    0.27260   1.526   0.1289        ⇒ not longer significant at 5% level.
XXmx_y      -0.44990    0.53096  -0.847   0.3981
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

FGLS Estimation: Cochrane-Orcutt – i_{MX}

Example (continuation):

Residual standard error: 0.09678 on 160 degrees of freedom
Multiple R-squared: 0.1082, Adjusted R-squared: 0.08038
F-statistic: 3.884 on 5 and 160 DF, p-value: 0.002381

$rho
e3
0.8830857 ⇒ very high autocorrelation.

$Corrected.b_1
XX
0.1663884 ⇒ Constant corrected if X does not include a constant

$Number.Iteractions
[1] 10 ⇒ algorithm converged in 10 iterations.

FGLS Estimation: Cochrane-Orcutt

• SE[$\mathbf{b}_{CO}$] and SE[r_{CO}] are obtained from the regression in the last iteration. If the constant is not pseudo-differentiatted, the estimated $b_{CO,0}$ has to be adjusted by (1- r_{CO}). Similar correction for SE[$b_{CO,0}$].

• If we do not want to lose the first observation, we can use the Prais-Winsten (1945) transformation of the first observation:

$$\text{sqrt}\{1 - \rho^2\}\ \mathbf{y}_1 \quad \& \quad \text{sqrt}\{1 - \rho^2\}\ \mathbf{X}_1$$

• A grid search around ρ can speed up the algorithm considerably. This is the Hildreth-Lu (1960) procedure.

• The iterative two-step estimation procedure can be easily extended to AR(p) models.

FGLS Estimation: Cochrane-Orcutt

• Note: Cochrane-Orcutt works well if the specified AR(p) structure is correct. Otherwise, we are in the presence of a missspecified model.

Example: For the 3 FF factor model for IBM returns we run C-O with an AR(1) process for ε_t: $\varepsilon_t = \rho\varepsilon_{t-1} + u_t$.

Then, after the final run, we do an LM-AR(3) test on the residuals, u_t. We do this by adding in the C-O procedure (& add to the list the last line: result$LM.AR3.test_u <- lm_t_u:

```
## lm_t for AR(3) in u
ols_u <- lm(u[4:T] ~ u[1:(T-3)] + u[2:(T-2)] + u[3:(T-1)])
r2_u <- summary(ols_u)$r.squared
lm_t_u <- (T-3)*r2_u

$LM.AR3.test_u
[1] 56.29834
```

⇒ Very significant. We need to use a higher AR(p) model.

FGLS & MLE Estimation

• We need to estimate $\mathbf{\Omega}$ ⇒ We need a model for $\mathbf{\Omega} = \mathbf{\Omega}(\boldsymbol{\theta})$.
⇒ In the AR(1) model, we had $\mathbf{\Omega} = \mathbf{\Omega}(\rho)$.

- FGLS estimation is done using Cochrane-Orcutt or NLLS.
- MLE can also be done, say assuming a normal distribution for u_t, to estimate ρ and β simultaneously. For the AR(1) problem, the MLE algorithm works like the Cochrane-Orcutt algorithm.

• For an AR(2) model, Beach-Mackinnon (1978) propose an MLE algorithm that is very fast to converge.

• For an AR(p) models, with p > 3, MLE becomes complicated. Two-step estimation is usually done.

MLE Estimation: Example in R

• Log likelihood of ARMA(1,1)-GARCH(1,1) Model:

```
log_lik_garch11 <- function(theta, data) {
 mu <- theta[1]; delta <- theta[2]; gamma <- theta[3]; alpha0 <- abs(theta[4]);
alpha1 <- abs(theta[5]); beta1 <- abs(theta[6]);  chk0 <- (1 - alpha1 - beta1)
 r <- ts(data)
 n <- length(r)

 u <- vector(length=n);
 u <- ts(u)
 u[1] <- r[1] - mu                                  # set initial value for u[t] series
 for (t in 2:n)
 {u[t] = r[t] - mu - delta*r[t-1] - gamma*u[t-1]}

 h <- vector(length=n);  h <- ts(h)
 h[1] = alpha0/chk0                                 # set initial value for h[t] series
if (chk0==0) {h[1]=.00001}                          #check to avoid dividing by 0
 for (t in 2:n)
 {h[t] = abs(alpha0 + alpha1*(u[t-1]^2)+ beta1*h[t-1])
if (h[t]==0) {h[t]=.00001} }                        #check to avoid log(0)

return(-sum(-0.5*log(2*pi) - 0.5*log(abs(h[2:n])) - 0.5*(u[2:n]^2)/abs(h[2:n])))
}
```

MLE Estimation: Example in R

• To maximize the likelihood we use optim (mln can also be used):

```
dat_xy <- read.csv(" http://www.bauer.uh.edu/rsusmel/phd/datastream-K-DIS.csv",head='TRUE,sep=",")
summary(dat_xy)
names(dat_xy)

z <- dat_xy$SP500                                   # S&P 500 90-2016 monthly data

theta0 = c(0.01,  -0.1, 0.01, -0.001, 0.2,  0.7)    # initial values
ml_2 <- optim(theta0,  log_lik_garch11, data=z, method="BFGS", hessian='TRUE)

ml_2$par                                            # estimated parameters

I_Var_m2 <- ml_2$hessian
eigen(I_Var_m2)                                     #check if Hessian is pd.
sqrt(diag(solve(I_Var_m2)))                         # parameters SE
```

Autocorrelation as a Common Factor

• From the first-order autocorrelated model

$$y_t = \rho y_{t-1} + \boldsymbol{X}_t' \beta - \boldsymbol{X}_{t-1}' \rho\beta + u_t \qquad (*)$$

• We can generalize (*) using the lag operator L –i.e., L $y_t = y_{t-1}$:

$$(1 - \rho L)\, y_t = (1 - \rho L)\, \boldsymbol{X}_t' \beta + u_t$$

Then, dividing by (1- ρL):

$$y_t = \boldsymbol{X}_t' \beta + u_t/(1 - \rho L) = \boldsymbol{X}_t' \beta + \varepsilon_t$$

• We can think of a model with autocorrelation as a misspecified model. The *common factor* (1 – ρL) is omitted. See Mizon (1977).

• We can generalize (*) even more by introducing more common lags:

(1 – B(L)) y_t = (1 – B(L)) $\boldsymbol{X}_t' \beta + u_t$ B(L): function of L, L^2,..., L^q; ρ

Common Factor Test

• From the AR(1) model:

(R) $$y_t = \rho y_{t-1} + \boldsymbol{X}_t' \beta - \boldsymbol{X}_{t-1}' \rho\beta + u_t \quad (*)$$

• We can think of (*) as a special case of a more general specification:

(U) $$y_t = \lambda_1 y_{t-1} + \boldsymbol{X}_t'\lambda_2 + \boldsymbol{X}_{t-1}' \lambda_3 + u_t$$

Restrictions needed to get (*): $\lambda_3 = -\lambda_1 \lambda_2$

• Hendry and Mizon (1980) propose testing the validity of the restrictions using a LR test, which has an asymptotic χ^2 distribution, with degrees of freedom equal to the number of restrictions .

$$LR = T\, log\, [RSS_R/RSS_U]$$

• The test is known as the *common factor* (COMFAC) *test*.

Common Factor Test

• We can use an *F-test* or Wald tests. See Mizon (1995) and McGuirk and Spanos (2004).

Note: Since the H_0 and H_1 models involve lagged y_t's, the test statistics do not follow the asymptotic distribution. Bootstraps are a good idea.

Common Factor Test - Example

• Common Factor Test for 3 FF factor model for IBM returns:

(U) Fit the unrestricted model: $y_t = \lambda_1 y_{t-1} + \boldsymbol{X}_t'\lambda_2 + \boldsymbol{X}_{t-1}'\lambda_3 + u_t$

```
> x <- cbind(x0,x1,x2,x3)
> x_l <- cbind(x1,x2,x3)
> reg_u <- lm(y[2:T]~y[1:T-1]+x[2:T,]+x_l[1:(T-1),] -1)
> sum(residuals(reg_u)^2)
[1] 2.92264
```

(R) Fit the restricted model: $y_t = \rho y_{t-1} + \boldsymbol{X}_t'\beta - \boldsymbol{X}_{t-1}'\rho\beta + u_t$

```
sum2 <- function(theta, x,y) {
rho1 <- theta[1];  mu <- theta[2]; beta <- theta[3:5]; lambda3 <- (-1)*rho1%*%beta
 r <- ts(y)
 T <- length(r)
T1 <- T-1
 u <- vector(length='T1);
u = r[2:T]- rho1*r[1:(T-1)] - x[2:T]*mu - x[2:T,2:4]%*%beta - x[1:(T-1),2:4]%*%t(lambda3)
return(sum(u^2))
}
```

Common Factor Test - Example

```
> theta0 = c(0.5, -.02, 0.01, -0.005,  -0.003)# initial values
> cf_r <- optim(theta0, sum2, x=x, y=y, method="BFGS", hessian=TRUE)

> cf_r$par
[1]  0.875011230 -0.027804863  0.009997961 -0.002767329 -0.003927199
> sum2(cf_r$par, x,y)
[1] 2.927888
> T*log(sum2(cf_r$par, x,y)/sum(residuals(reg_u)^2)                    # LR COMFAC TEST
[1] 0.5561482
```

- *F-test* = [(**2.927888** - **2.92264**)/3]/[**2.92264**/311] = 0.1861477
  $\Rightarrow$ cannot reject H_0 at 5% level.

Note: The restricted model seems OK. But, we need to check that the model is well specified. In this case, does the AR(1) structure is enough to remove the autocorrelation in the errors?

Common Factor Test - Example

- We do an LM-AR(5) test to check the errors in the U Model:

```
> fit_u <- lm(e_u[(5+1):T]~e_u[1:(T-5)]+e_u[2:(T-4)]+e_u[3:(T-3)]+e_u[4:(T-2)]+e_u[5:(T-1)])
> r2_e_u <- summary(fit_u)$r.squared
> lm_t_u <- (T-4)*r2_e_u
> lm_t_u
[1] 70.75767
```

$\Rightarrow$ Very significant (p-value: 1.6e-14). An AR(1) structure is not sufficient to remove AR in errors.

In general, if we allow for more dynamics in the U Model we do better. For example, we use 4 lags in y_t and 2 lags in X_t:

```
> reg_u4 <- lm(y[5:T]~y[1:(T-4)]+y[2:(T-3)]+y[3:(T-2)]+y[4:(T-1)] +x[5:T,]+x_l[4:(T-1),]+x_l[3:(T-2),] -1)
> e_u4 <- residuals(reg_u4)
> fit_u5 <- lm(e_u4[(5+1):T]~e_u4[1:(T-5)]+e_u4[2:(T-4)]+e_u4[3:(T-3)]+e_u4[4:(T-2)]+e_u4[5:(T-1)])
> r2_e_u5 <- summary(fit_u5)$r.squared
> lm_t_u5 <- (T-5)*r2_e_u5
> lm_t_u5
[1] 6.938392
```

$\Rightarrow$ Not significant (p-value: .139).

Building the Model

- Old (pre-LSE school) view: A feature of the data
  - "Account" for autocorrelation in the data.
  - Different models, different estimators

- Contemporary view: Why is there autocorrelation?
  - What is missing from the model?
  - Build in appropriate dynamic structures
  - Autocorrelation should be "built out" of the model
  - Use robust procedures (OLS with Newey-West SE) instead of elaborated models specifically constructed for the AR errors.