

RATS Handbook for Panel and Grouped Data

Thomas A. Doan
Estima

Final Version
January 18, 2026

Copyright © 2026 by Thomas A. Doan

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Contents

Preface	v
1 Introduction	1
1.1 Heterogeneous vs Homogeneous parameters	2
1.2 Panel vs Grouped Data	2
1.3 Balanced vs Unbalanced Panels	3
1.4 PANEL date scheme	3
2 Preparing Data	6
2.1 Combining Multiple Time Series.	6
2.2 Changing the Blocking	7
2.3 Mixed Time-Varying and Time-Invariant Data	8
3 Computational Tools for Panel Data	9
3.1 SET, SSTATS and related instructions	9
3.2 PANEL	9
3.3 SWEEP	10
3.4 PSTATS	11
3.5 BOOT and Bootstrapping	11
3.6 LWINDOW=PANEL and Clustered Standard Errors.	12
4 Fixed Effects	15
4.1 A More Realistic Example.	17
4.2 Balanced vs Unbalanced Samples	18
4.3 Implementing with RATS	19
4.4 Testing Fixed Effects	23
4.5 METHOD=BETWEEN	25
4.1 Fixed Effects	25

5 Random Effects	28
5.1 The Random Effects Estimator	29
5.2 Estimating the Component Variances	30
5.3 Using RATS	32
5.4 Hausman Test	34
5.5 Direct Calculation of Component Variances	35
5.6 Random Effects Transformations	36
5.1 Random Effects	38
6 Two-Way Effects	40
6.1 Balanced vs Unbalanced	41
6.2 Using RATS	42
6.3 Hausman Tests	44
6.4 PANEL instruction	44
6.1 PREGRESS with Two-Way Effects	45
7 Dynamic Panels	46
7.1 The Bias in Fixed Effects Estimators	46
7.2 The Examples	48
7.3 First Difference Instrumental Variables Estimators	49
7.4 Expanded Instrument Sets	51
7.5 Bias Correction.	53
7.1 Dynamic Panel-Instrumental Variables Estimation	56
7.2 Dynamic Panel-Biased Correction Estimation	58
8 Non-Linear Models	59
8.1 Non-linear Least Squares	59
8.2 Probit and Logit Models.	64
8.2.1 Conditional (Fixed Effects) Logit	65
8.2.2 Random Effects Probit	68
8.1 Non-linear Least Squares with Fixed Effects	72
8.2 Logit and Probit with Individual Effects	74

9 Unit Root Testing	77
9.1 The Example.	78
9.2 Levin-Lin-Chu test	78
9.3 Harris-Tzavalis Test	81
9.4 Im-Pesaran-Shin Test.	81
9.5 Breitung Test	82
9.6 Hadri Test	85
9.1 Panel Unit Root Tests	87
10 Cointegration and Error Correction Models	88
10.1 Pedroni Tests	88
10.2 Estimation with Heterogeneous Cointegrating Vectors.	92
10.2.1 Fully-Modified Least Squares	92
10.2.2 Dynamic OLS	95
10.3 Estimation with Homogeneous Cointegrating Vectors	96
10.3.1 Mark and Sul DOLS	96
10.3.2 Pesaran-Shin-Smith Pooled Mean Group	98
10.1 Panel Cointegration with Heterogeneous Cointegrating Vectors .	101
10.2 Panel Cointegration with Homogeneous Cointegrating Vector . .	104
10.3 Panel Error Correction Model	106
11 Simulation and Bootstrap Methods	109
11.1 Linear Random Effects	109
11.2 Random Effects Probit	113
11.3 Bootstrapping	117
11.1 Random Effects by Gibbs Sampling	122
11.2 Random Effects Probit by Gibbs Sampling	124
11.3 Bootstrapping a Unit Root Test	126
12 Mean Group and Related Estimators	129
12.1 Mean Group Estimator	129
12.2 Swamy Random Coefficients Models	131
12.3 MCMC Estimation for Random Coefficients.	133
12.1 Mean Group and Random Coefficients Models	138

13 Vector Autoregressions	141
13.1 Instrumental Variables Estimators.	141
13.2 Shrinkage Estimators, Univariate Autoregressions	144
13.3 Shrinkage Estimators, VAR's	148
13.4 Causality Tests.	153
13.1 Panel VAR with instrumental variables	155
13.2 Univariate Autoregression: Shrinkage Estimator	156
13.3 Vector Autoregression: Shrinkage Estimator	158
13.4 Causality Test	161
 A Random Effects Component Estimators	 163
A.1 Individual Effects Only	163
A.1.1 Wallace-Hussain (OLS-based)	164
A.1.2 Amemiya or Wansbeek-Kapteyn (Fixed Effects-based) . . .	166
A.1.3 Swamy-Arora (Fixed and Between)	167
A.2 Two-Way Effects	168
A.2.1 Wallace-Hussain	169
A.2.2 Amemiya (Fixed Effects-based)	169
A.2.3 Swamy-Arora (Fixed and Between)	170
 B Corrected Covariance Matrices	 171
 C Probability Distributions	 174
C.1 Chi-Squared Distribution	174
C.2 (Scaled) Inverse Chi-Squared Distribution	175
C.3 Univariate Normal	176
C.4 Multivariate Normal	177
C.5 Gamma Distribution	178
C.6 Inverse Gamma Distribution	179
C.7 Wishart Distribution	180
C.8 Inverse Wishart Distribution	181
 D Properties of Multivariate Normals	 183
 Bibliography	 186
 Index	 188

Preface

This workbook is based upon the content of the RATS e-course on Panel and Grouped Data, offered in spring 2012. While it covers the basic techniques of panel data econometrics, the emphasis is on the “time-series” aspects, with Dynamic Panels (Chapter 7), Unit Root Tests (Chapter 9), Cointegration (Chapter 10) and VAR’s (Chapter 13). It also includes several examples of the use of Gibbs sampling for panel data including linear and non-linear random effects in Chapter 11 and random coefficients models in Chapter 12 and applied to VAR’s in Chapter (13).

This makes use of a number of important features for dealing with panel data added with RATS versions 8 and 8.1. The improvements to the core instructions **PREGRESS**, **PANEL** and **PFORM** provide greater flexibility for dealing with both panel and general grouped data.

We use bold-faced Courier (for instance, **DLM** for any use of RATS instruction or procedure names within the main text, and non-bolded Courier %SCALAR) for any other pieces of code, such as function and variable names. For easy reference, the full text of each example is included. The running examples are also available as separate files.

Introduction

This course covers a wide range of applications of panel or grouped data using RATS. The use of *panel data*¹ has become increasingly important in econometric practice. Most applications twenty to thirty years ago were on survey data with a large number of individuals followed over a short number of time periods. The Panel Study for Income Dynamics (PSID for short), for instance, has 18,000 individuals, with annual data starting in 1968. That type of data set is mainly used to answer questions about the “individual” behavior, where the added data from the “time” dimension allows not just better inference due to more total observations, but also an opportunity to see how people respond to changed circumstances (changes in number of children at home, for instance). That particular data set has been heavily used (over 3000 published papers have used it as the source), but this type of data set is extremely expensive to produce, generally being done either by government agencies or with very substantial government funding.

The tremendous growth in the importance of panel data techniques more recently is due to its application to questions about “time series” behavior. Proper inference regarding unit roots or cointegration in a single time series generally requires a fairly long run of data. Various obvious structural breaks (switch to the Euro, change in Federal Reserve targeting, etc.) restrict the ability to extend one country’s data very far. The only way to bring more data to bear is to use multiple countries, that is, to create a panel data set with countries as individuals. Producing *this* type of data set is within the reach of any one, since it simply requires combining (usually public) time series data from multiple countries.

The number of individuals (or countries or firms) in the data set is usually denoted N , while the number of time periods is usually T . The first type of data set (like the PSID) is known as a *big N -small T* data set, while the second is *big T -small N* . Because of their different natures, different techniques apply to each type, and each type has its own asymptotic behavior regarding what can be estimated consistently. Some estimation methods on the second type of data set also can require *big T -big N* asymptotics, that is, they have asymptotic results that require both $T \rightarrow \infty$ and $N \rightarrow \infty$.

¹also called *cross section-time series data* or *longitudinal data*

1.1 Heterogeneous vs Homogeneous parameters

In most applications of panel data, the set of unknown parameters can be partitioned into those that are the same across individuals (*homogeneous parameters*) and those that are allowed to vary across individuals (*heterogeneous parameters*). The principal (and perhaps sole) interest is in that first set. The two advantages of the panel set compared to a simple cross section data set are:

- You have NT observations (rather than just N) to use in estimating the homogeneous parameters.
- You can control for the heterogeneous parameters.

The heterogeneous parameters are often (in fact, usually) “nuisance” parameters which are of no real interest in themselves. This is good since they often can’t be estimated consistently anyway. An individual-specific variance or intercept has only T data points which apply to it—with T small, the estimates of those won’t be consistent. Even if T is “large”, the heterogeneous parameters are often discarded. For instance, in panel unit root tests (Chapter 9), the typical assumption is that the short-run dynamics (the stationary part) are heterogeneous. In a standard Dickey-Fuller test on a single time series, those short-run values are rarely reported, except, perhaps, for a mention of how many lags were used, and the same is true in the panel version.

The correct choice of this partitioning is very important. Different decisions often force the use of completely different estimation methods. For instance, a frequent question that we get is whether RATS can do “panel VAR’s”. This is not a well-formed question. There are, in fact, a whole range of (very different) methods to estimate VAR’s with panel data depending upon what is assumed to be homogeneous. In some cases (but not always), there are tests available for determining whether a decision to treat a set of parameters as homogeneous is appropriate. For instance, if there are more regression parameters than data points per individual, there is no way to estimate a fully heterogeneous model, so some parameters will have to be considered fixed.

1.2 Panel vs Grouped Data

Most of the course will use panel data, but many of the techniques will also apply to data where the heterogeneity is based upon general groupings within the data set with no “time” dimension. For instance, Empirical Example 9.3 in Baltagi (2008) is a snapshot of U.S. Census data in 1970. The assumed heterogeneity is related to the town (or neighborhood within the city of Boston). Any panel data technique which doesn’t rely specifically upon time sequencing (no lags in the model, no time period dummies or time trends) will generally also apply to data such as this.

1.3 Balanced vs Unbalanced Panels

A *balanced* panel data set has the same set of time periods for each individual. An *unbalanced* panel has some missing periods for one or more individuals. Balanced sets are unquestionably easier to handle, both from the standpoint of data organization and also computationally. While there are quite a few statistical methods which apply to panel data which are advertised as only working for balanced sets, that is rarely actually the case. What is often true is that the additional calculations required for unbalanced data are very unpleasant and perhaps infeasible even with the power of modern computers. In other cases, the restriction is imposed (by the programmer) due to the greater ease of organizing and working with a balanced panel.

1.4 PANEL date scheme

The `PANEL` date scheme in RATS requires that the data set be at least nominally balanced—the data are organized within RATS as blocks of T data points for each of N individuals. However, within those blocks you can have missing data and thus can have a data set which is, in fact, unbalanced. For short, we'll refer to data handled this way within RATS as `PANEL`-dated data. This is set using a `CALENDAR` instruction like:

```
calendar(panelobs=15,a) 1973
```

which blocks 15 observations per individual, with an annual date scheme starting in 1973 within each record. With a panel date scheme in place, you can use `%INDIV(t)` in transformations to refer to the individual number and `%PERIOD(t)` to refer to the time period (starting at 1) within the individual record. The notation `i//date` means individual number `i` at date (within the individual record) `date`. You can also use entry numbers within the record, so `1//1973:1` is equivalent to `1//1`.

What this type of blocking does is to make it easier to handle lags, whether in doing transformations or using lag fields in regressions. The following is the setup code for the replication example for Hansen (1999):

```

open data invest.txt
calendar(panelobs=15,a) 1973
data(format=free,org=columns) 1//1973:01 565//1987:01 $
    inva vala cfa debta
*
* Do data transformations
*
set q = vala
set q2 = vala^2
set q3 = vala^3
set qd = q*debta
*
* Set the threshold and breaking variables
*
set threshvar = debta{1}
set breakvar = cfa{1}

```

The original data set is fully balanced, with 15 observations on each of 565 firms. The four derived series `Q`, `Q2`, `Q3` and `QD` will similarly have 15 observations per individual. However, `THRESHVAR` and `BREAKVAR` both are lags of original data. With the `PANEL` date scheme in place, **SET** realizes that the data entry immediately before the 1973:1 entry for individual 2 isn't for individual 2, but is the final entry for individual 1. Because of this, it makes the value of `BREAKVAR` there missing. This shows the two series (`CFA` and `BREAKVAR`) at the transition from the record for individual 1 to that for 2.

ENTRY	CFA	BREAKVAR
1//1986:01	0.20558	0.22521
1//1987:01	0.15122	0.20558
2//1973:01	0.09633	NA
2//1974:01	0.04428	0.09633
2//1975:01	0.08036	0.04428
2//1976:01	0.10102	0.08036

This keeps the original data aligned, despite the fact that there are now only 14 fully valid data points in each individual. When you use the data in a regression, the year 1973 entries will be dropped out because of the NA. The same thing happens if you use a lag in the regressor list. For instance, the result from

```

linreg inva
# constant q{1} q2{1} q3{1} debta{1} qd{1} cfa{1}

```

is

Linear Regression - Estimation by Least Squares				
Dependent Variable INVA				
Panel(15) of Annual Data From 1//1974:01 To 565//1987:01				
Usable Observations	7910			
Degrees of Freedom	7903			
Skipped/Missing (from 8474)	564			
Centered R ²	0.0775573			
R-Bar ²	0.0768570			
Uncentered R ²	0.6979130			
Mean of Dependent Variable	0.0887200480			
Std Error of Dependent Variable	0.0619148484			
Standard Error of Estimate	0.0594879917			
Sum of Squared Residuals	27.967303622			
Regression F(6,7903)	110.7450			
Significance Level of F	0.0000000			
Log Likelihood	11101.5659			
Durbin-Watson Statistic	1.2263			
Variable	Coeff	Std Error	T-Stat	Signif

1. Constant	0.0668	1.3752e-003	48.58582	0.00000000
2. Q{1}	0.0107	7.9105e-004	13.47231	0.00000000
3. Q2{1}	-3.8654e-004	2.4700e-005	-15.64921	0.00000000
4. Q3{1}	2.7059e-006	1.9628e-007	13.78603	0.00000000
5. DEBTA{1}	0.0133	3.7129e-003	3.57754	0.00034892
6. QD{1}	2.3455e-003	1.5811e-003	1.48347	0.13798830
7. CFA{1}	0.0318	3.0029e-003	10.58032	0.00000000

Note, by the way, that the Skipped/Missing count is 564, not 565. With the **LINREG** asking for the maximum valid range, because the overall entry 1 (year 1973 for individual 1) is missing, the limits of the estimation range go from the year 1974 entry for 1 to the end. 1//1973:1 isn't counted as an embedded missing value since it's outside the range. If you force the regression to start at entry 1:

```
linreg inva 1 *
# constant q{1} q2{1} q3{1} debta{1} qd{1} cfa{1}
```

you'll get exactly the same results, except the Skipped/Missing will be 565 from 8475.

Hansen did his empirical work using Gauss™. With math packages like that, you would typically have to compress the data matrix for further calculations. But that's not just a single shift, as you would do with a single time series with lags; instead you need to go from T to $T - 1$ for each individual (or $T - p$ if you use more than one lag). Getting that right can be somewhat tricky, particularly if there are other calculations, and, in fact, it was done incorrectly in this case.²

²This is where programs often require balanced data even if not theoretically necessary. The bookkeeping required to manually compress unbalanced data sets is considerable.

Preparing Data

Almost all the data sets used in our textbook or paper replication examples are already in the proper form for `PANEL`-dated data, that is, they are one variable per column, blocked by individual. This is very convenient for teaching purposes, but also quite unrealistic in the real world. Actual survey data would be more likely blocked by time period since that's the order in which it's acquired. Macroeconomic data would generally be available as separate time series for each individual.

It's usually a good idea to add one step to the analysis to reformat your original source data, which is what clearly was done (at some point) in all of those example data sets. The instruction **PFORM** (Panel data FORM) can handle quite a few different rearrangements of original source data.

2.1 Combining Multiple Time Series

This is an example from Chapter 4 of Enders (2010).¹ This has quarterly data on real effective exchange rates for eight countries, with separate columns for each country's data. This form of **PFORM** generates the series `EXRATE` by stringing together the data from the listed series (in order). It also makes `%NOBS` equal to the block size of each individual. (There could be missing values, which are inserted to pad out each country's record.) The total number of entries is `EXRATE` will be $8 * \%NOBS$, which is what is used in the **SET** instruction.

```
open data panel.xls
calendar(q) 1980:1
data(format=xls,org=columns) 1980:01 2008:01 australia canada $
    france germany japan netherlands uk us
*
* Form the separate series into a single panel data series
*
pform exrate
# australia canada france germany japan netherlands uk us
set lexrate 1 %nobs*8 = log(exrate)
*
cal(panelobs=%nobs,q) 1980:1
all 8//2008:1
```

¹Example file `ENDERSP244.RPF`

2.2 Changing the Blocking

This is an example of reformatting a data set that we will use for demonstrating panel VAR estimation using the techniques from Holtz-Eakin et al. (1988).²

The original data is in a very dangerous format—it's straight binary, which should *never* be used.³ So we definitely want to re-save the data in a better format. Also, the raw data are grouped by year, which was probably how a research assistant gathered it. (The individuals are municipal governmental units). Because we want to use a dynamic model, we need to reblock by individual. **PFORM** in this form (with the `INDIV` and `TIME` options) takes the input series (`TEMP` which is, in each case, a copy of one of the original series) and rearranges it with the series pointed to by `INDIV` identifying the individual, and the `TIME` series identifying the time period. Note that the blocking for the output data depends only upon the `INDIV` and `TIME` series, not the data in the series being worked on. If, for instance, governmental unit 720001002 has missing values for the series `GRANT` for all time periods, `GRANT` will still have the slots allocated to it for that unit. That way, the data for all series formed with the same `INDIV` and `TIME` options will be aligned properly.

Note that we make copies of the original `GOVID` and `YEAR` variables for use in the **PFORM**. The first two series rearranged by **PFORM** will be the original `GOVID` and `YEAR` since they are the first two covered by the `DO` loop. If we used `INDIV=GOVID, TIME=YEAR` on **PFORM**, the alignment would be off because the original identification would change.

```
all 161*8
open data labor.dat
data(format=binary,org=columns) 1 1288 govid year ftpay full $
  ptpay part pay employ wage pwage fwage pop me ma nh nj ny ri $
  tn va grant own comp debtey assetey fcomp tcomp netass $
  null nullx
set id      = govid
set timeid  = year
do i=govid,nullx
  set temp = i{0}
  pform(indiv=id,time=timeid) i
  # temp
end do i
open copy labor.xls
copy(format=xls,org=columns) / govid to nullx
```

²This was provided by the authors of that paper, but isn't the original data set.

³There is no identifying information on the file, and there isn't even any information to tell anyone what type of file it is.

2.3 Mixed Time-Varying and Time-Invariant Data

The following is from the data set used in Section 10.8 of Baltagi (2008). This has patent data on 346 firms. Each firm has one row in the data set; the first five columns of that have time-invariant data and the last twenty have two blocks of ten years worth of data on (log) spending on research and development and on patents. This needs to be rearranged into a panel data set which requires

- Replicating the time-invariant values for each time period within a company's record
- Combining the ten separate series for R & D and patents into single series

```
open data patentdata.txt
data(format=free,org=columns) 1 346 cusip ardssic scisect logk $
  sumpat logr70 logr71 logr72 logr73 logr74 logr75 logr76 logr77 $
  logr78 logr79 pat70 pat71 pat72 pat73 pat74 pat75 pat76 $
  pat77 pat78 pat79
*
* Rearrange the data into a panel data set
*
pform(input=time) logr
# %slike("logr##")
pform(input=time) patent
# %slike("pat##")
cal(panelobs=%nobs,a) 1970
all %ngroup//%nobs
dofor s = cusip ardssic scisect logk sumpat
  set(nopanel) %s("p_"+%1(s)) = ([series] s)(%indiv(t))
end dofor s
```

The time-varying data is combined into a single panel series using **PFORM** with the **INPUT=TIME** option. This is similar to what is done in Section 2.1 except that the input series represent time periods rather than individuals. `%slike("logr##")` returns a list of series matching “logr” followed by two digits. Note, by the way, that this does the list in order of their creation, not alphabetically. If, for instance, the data series were something like `LOGR98`, `LOGR99`, `LOGR00`, `LOGR01`, etc., they would be still be in the order listed on the **DATA** instruction.

To do the replications of the time-invariant data, we need to switch over to the **PANEL**-dated scheme. The **PFORM** instruction sets both the `%NGROUP` and `%NOBS` variables, where the first is the number of individuals and the second the number of data points per individual. This makes it easier to do the necessary **CALENDAR** and **ALLOCATE** instructions. The **DOFOR** loop creates series named `P_CUSIP`, `P_ARDSSIC`, etc. by replicating the values of `CUSIP`, `ARDSSIC`, etc. We need the **NOPANEL** option on **SET** since the original data are just in the first 346 observations and aren't slotted based upon the new panel data setup.

Computational Tools for Panel Data

3.1 SET, SSTATS and related instructions

We've already briefly described how **SET** deals with transformations involving lags. The same treatment applies to related instructions like **SSTATS** and **GSET**. If any series entry reference in a calculation is from a different individual than the current entry (T), that reference is treated as a missing value.

In some cases (for instance, in doing bootstrapping, section 3.5), we really need to do a transformation that crosses individual boundaries. To make that work, you need to use the **NOPANEL** option on **SET**. That turns off the check.

3.2 PANEL

PANEL does a wide range of useful calculations on panel and grouped data. It was much more important before the addition of the separate **PREGRESS** instruction (with version 5) but still plays a vital role when you want to go outside of the built-in instructions.

The most common use is to remove individual-specific means out of a series. For instance:

```
panel(smpl=psmpl,entry=1.0,indiv=-1.0) y start1 endl ytilde
```

ENTRY is the weight on the original data (y_{it}) and **INDIV** is the weight on the individual mean (average across t at for individual i), so **YTILDE** will be created from **Y** by subtracting individual means. If you just need the individual means themselves, you would use **INDIV=1.0** as the option. You can also get the count of data points for an individual (**ICOUNT=1.0**) or the sum of the data (**ISUM=1.0**), and similarly for counts, sums and averages across individuals for each time period, with options **TCOUNT**, **TSUM** and **TIME**.¹ With version 8.1, you can now use the **GROUP** option to provide a general grouping variable to identify the individuals. For instance,

```
panel(group=townid,icount=1.0) %resids / bcount
```

will create **BCOUNT** as a series with the count of entries in a particular town. (If there are 20 data points sharing a value of **TOWNID**, each entry in **BCOUNT** corresponding to those will have the value 20).

¹The **ICOUNT** and **TCOUNT** options are described incorrectly in the RATS v8 Reference Manual.

Another new feature is the combination of `ID` and `IDENTRIES` options. This will be used for doing non-linear random effects estimators. These require doing a likelihood calculation for each individual aggregating across the time periods for that individual, which must be repeated many times. For an unbalanced sample, you want to make a list of those entries for an individual once and only once rather than having to do it for each function evaluation. (For a balanced `PANEL`-dated sample, the entries for a given individual are easily located.)

```
panel (group=p_cusip, id=vid, identries=identries, smpl=%year(t)>=1975)
```

This makes `VID` into a `VECTOR` with the (sorted) values of the grouping variable (`P_CUSIP`) and `IDENTRIES` into a `VECTOR` of `VECTOR[INTEGER]` with the list of entry numbers for the corresponding element in the `VID` series. For instance, if we have the following grouping series: 10,10,6,6,6, the `VECTOR` generated by the `ID` option would be 6.0, 10.0, the first element of the `IDENTRIES` option would be the list 3,4,5 (entries for grouping value 6) and the second would be 1, 2 (entries for grouping variable 10).

As we'll see later, removing individual effects by subtracting the mean across time periods can create problems with dynamic models because it uses data from both before and after the given entry. There are alternative transformations which use only "forwards" or only "backwards" data. With version 8.1, there's a new option, `GLS`, which allows choices of `GLS=FORWARDS` and `GLS=BACKWARDS`, along with (the default) `GLS=STANDARD` which uses the full individual sample mean.

3.3 SWEEP

SWEEP is a very handy instruction when you need to do a set of linear regressions which take the same set of regressors, whether the set of regressions are different samples, or different dependent variables or both. For analyzing sample splits, the key option is `GROUP`, which gives a formula which takes distinct values for the sets of entries which define the sample splits. For `PANEL`-dated data, this would be `GROUP=%INDIV(T)` for splitting the sample on individuals and `GROUP=%TIME(T)` for splitting it on time periods.

The principal use of **SWEEP** in this course will be to remove heterogeneity in slope coefficients. The **PANEL** (section 3.2) instruction is simpler, but can remove heterogeneity only in the intercept. The following is from Mark & Sul (2003). They are estimating a long-run money demand function (that is, cointegrating relationship) using dynamic OLS (section 10.3.1) of real money on real GDP and interest rate. The coefficients in the cointegrating relationship are assumed to be homogeneous across countries. The intercept in that is allowed to vary (differences in measurement units make that a necessity), and the short-run dynamics and residual variances are also heterogeneous. This

use of **SWEEP** regresses each of the three variables which enter the (homogeneous) cointegrating regression (**REALM**, **REALY** and **RATE**) on the DOLS regressors which are assumed to have heterogeneous coefficients. The residuals are put into the **VECT[SERIES]** called **SWEEPS** in the order from the first supplementary card (for instance, **SWEEPS(1)** are the residuals for **REALM**). The cointegrating relation is then estimated by running a **LINREG** on the swept-out variables.

```
sweep(group=%indiv(t),series=sweeps,variance=heterogeneous)
# realm realy rate
# constant dy{-p to p} dr{-p to p}
labels sweeps
# "Money" "Income" "Rate"
linreg(noprint) sweeps(1)
# sweeps(2) sweeps(3)
```

3.4 PSTATS

PSTATS is short for Panel STATistics. It computes one- or two-way analysis of variance on a series. Alternatively, it can be used to test for equal variances among the individuals. Analysis of variance (and related non-parametric tests) are important tools in working with experimental data, but are less important in non-experimental economic data. Example 9.3 from Greene (2007)² does the following:

```
pstats(group=oced,tests,smpl=country<>-999.0) dale
pstats(group=country,tests,smpl=country<>-999.0) comp
```

which isn't used as a formal test, but rather a description of how much cross-country variation there is in the variables, relative to in-country variation.³

Analysis of Variance for Series COMP					
Source	Sum of Squares	Degrees	Mean Square	F-Statistic	Signif Level
INDIV	4229596.0981304	190	22261.0320954	1927.153	0.000000
ERROR	6503.3558016	563	11.5512536		
TOTAL	4236099.4539320	754			

3.5 BOOT and Bootstrapping

With any type of bootstrapping, it's important to figure out some arrangement of the information in the data set so that the shuffled data would be expected to be from the same distribution. We'll discuss later what panel data can allow that wouldn't be available for single time series. However, no matter how you shuffle the entries with panel data, you need to take an extra step in rebuilding your data. A typical use of the **BOOT** instruction in RATS is the following:

²Example 11.4 in the 7th edition

³The data set is a very unbalanced one with between 1 and 5 data points per country.

```
boot entries 1 50
set shuffle = ressqr(entries(t))
```

which draws random entries, then rearranges one or more series based upon that. If you follow exactly that procedure with `PANEL`-dated data, you'll get a series with mostly missing values. The problem is the behavior of `SET` (section 3.1), which is designed to prevent errors in computations that cross boundaries between individuals, now stands in the way of a calculation which is *supposed* to cross boundaries between individuals. To make the bootstrapping work correctly, you need to use the `NOPANEL` option on `SET`. See Section 11.3 for details on bootstrapping panel data with RATS.

3.6 LWINDOW= PANEL and Clustered Standard Errors

In a regression on a single time series, computing an estimate for the covariance matrix of the coefficients requires an estimator for:

$$E\left(\frac{1}{T}\mathbf{X}'\mathbf{u}\mathbf{u}'\mathbf{X}\right) \quad (3.1)$$

where \mathbf{X} is the full $T \times k$ matrix of regressors. Now $\mathbf{X}'\mathbf{u}\mathbf{u}'\mathbf{X}$ can be simplified considerably to $(\mathbf{X}'\mathbf{u})(\mathbf{X}'\mathbf{u})'$ where $\mathbf{X}'\mathbf{u} = \sum_{t=1}^T X_t' u_t$. This suggests estimating (3.1) using the empirical estimate

$$\frac{1}{T} \left(\sum_{t=1}^T X_t' u_t \right) \left(\sum_{t=1}^T X_t' u_t \right)' \quad (3.2)$$

This allows for arbitrary patterns of serial correlation, is positive semi-definite by construction and is considerably simpler to compute and use than Newey-West or related estimators, all of which require some method of choosing a lag length or band width and require double sum calculations over time and lag. The problem with this is unfortunately also simple—it's by construction a rank one matrix as the outer product of a single vector (each row is just a different linear combination of the same vector).

Suppose, however, that we have a panel data set (for notational convenience, we'll assume it's balanced). If we assume that $X_{it}u_{it}$ is independent of $X_{js}u_{js}$ if $i \neq j$, then

$$\begin{aligned} E\left(\frac{1}{NT}\mathbf{X}'\mathbf{u}\mathbf{u}'\mathbf{X}\right) &= E\frac{1}{NT} \left(\sum_{i=1}^N \sum_{t=1}^T X_{it}' u_{it} \right) \left(\sum_{i=1}^N \sum_{t=1}^T X_{it}' u_{it} \right)' \\ &= E\frac{1}{NT} \sum_{i=1}^N \sum_{j=1}^N \left(\left(\sum_{t=1}^T X_{it}' u_{it} \right) \left(\sum_{s=1}^T X_{js}' u_{js} \right)' \right) \\ &= E\frac{1}{NT} \sum_{i=1}^N \left(\left(\sum_{t=1}^T X_{it}' u_{it} \right) \left(\sum_{t=1}^T X_{it}' u_{it} \right)' \right) \end{aligned} \quad (3.3)$$

where the last line follows because of the independence assumption when $i \neq j$. In the final form, each summand again has rank one, but now we're adding up N of them, which, so long as $N > k$ should give us a rank k matrix.

Using this calculation gives us what are known as *clustered standard errors* which are robust to arbitrary patterns of correlation within the cluster. In this case, they are clustered (grouped) by individuals, though the same general calculation works when the data are grouped according to some other criterion. The key assumption is that the residuals are independent between groups. Particularly with panel data, you would need to be careful about that, since there might be a tendency for residuals to be correlated if they are for the same date.

To get individual-clustered standard errors in a `PANEL`-dated data set, include the option `LWINDOW=PANEL` on instructions like `LINREG` or `NLLS`. The following is an example from Greene (2007), Example 9.1.⁴ This is a data set which has been extracted from the PSID.

```
open data cornwell&rupert.xls
calendar(panelobs=7,a) 1976
data(format=xls,org=columns) 1//1976:01 595//1982:01 exper wks $
    occ ind south smsa ms fem union ed blk lwage
*
set expersq = exper^2
linreg lwage
# constant exper expersq wks occ ind south smsa ms union ed fem blk
*
* Clustered standard errors are done using the options ROBUST and
* LWINDOW=PANEL
*
linreg(robust,lwindow=panel) lwage
# constant exper expersq wks occ ind south smsa ms union ed fem blk
```

Despite the comments, the `ROBUST` option actually isn't necessary; `LWINDOW=PANEL` by itself is enough. The output header for this adds the second line below to identify the difference between this output and simple least squares:

```
Linear Regression - Estimation by Least Squares
With Clustered Standard Error Calculations
```

For clustering on any other variable, use the option `CLUSTER=grouping variable`. The grouping variable must use a different value for each group that it identifies, but they don't have to be in any particular sequence: 60201, 60202, 60208 is every bit as good as 1, 2, 3.

In addition to individual-clustered standard errors, there are several other alternative procedures for "correcting" covariance matrices to allow for other types of effects. See Appendix B.

⁴It's example 11.1 in the 7th edition.

LWINDOW= PANEL and CLUSTER are also options for the instruction **MCOV**, where they will be used to compute values needed in quite a few different testing procedures. For instance,

```
mcof(cluster=%indiv(t))
# u
mcof(cluster=%time(t))
# u
```

compute (respectively) $\sum_{i=1}^N \left(\sum_{t=1}^T u_{it} \right)^2$ and $\sum_{t=1}^T \left(\sum_{i=1}^N u_{it} \right)^2$ both of which are needed for the Breusch-Pagan LM test done using the **BPPANELTESTS** procedure.

Fixed Effects

We'll start with a simple example of the usefulness of panel data. In basic statistics, it's taught that there are two (elementary) ways to test for a difference in two means with and without some form of "treatment". We'll represent this with $D_i = 0$ for individuals that haven't had the treatment and $D_i = 1$ for those that have. With un-matched or random samples, you get random samples of individuals with $D_i = 0$ and with $D_i = 1$. Assuming the populations are Normal with common variance, the test statistic for the treated mean being different from the untreated is

$$\frac{(\bar{y}_1 - \bar{y}_0)}{\sigma \left(N_1^{-1/2} + N_0^{-1/2} \right)}$$

where N_1 and N_0 are the sample sizes. This will be a t with $N_1 + N_0 - 2$ degrees of freedom if we have to estimate σ . Because the two samples are (presumably) independent, the variance of the difference is the *sum* of the variances. It's pointed out that this is likely to be a rather weak testing procedure, requiring quite large samples, because the residual standard deviation σ is covering the full range of differences in the sample other than the treatment of interest.

Now, the whole setup here is overly simplistic. If we are going to the trouble of sampling individuals (for concreteness, we'll assume we're talking about people), we would probably also collect demographic information such as sex, race, marital status, level of education, etc.; anything that might help to reduce the rather considerable variation in the mean in the population as a whole. When we combine all the observations into one sample, we would do the analysis using a regression of the form:

$$y_i = \alpha + Z_i\beta + D_i\delta + \varepsilon_i \tag{4.1}$$

where the Z_i are the demographics. We would run a t -test on $\delta = 0$. If the Z_i are omitted, this reduces to exactly the same test statistic as above. Now there are several important assumptions at work in (4.1):

1. The β and δ are the same for all individuals.
2. The ε_i is uncorrelated with Z_i and D_i .

δ being fixed across individuals is the maintained assumption that's required in order to even do the test. Making sure that β is fixed sometimes requires

some ingenuity with the choice of the demographic shifts. For instance, interaction terms are commonly used (i.e. separate dummies for black female or young, college-educated). The assumption (2) that we have a proper linear regression can be violated if the shift variables are poorly chosen (such as using education in Z_i with y_i being earnings or something similar) or if there is some endogenous sample selection problem.

However, let's assume that, instead of a random sampling of individuals, some with $D = 1$, some with $D = 0$, we have a sampling of individuals where we have one observation with $D = 0$ and another with $D = 1$. Assuming the demographics are the same at each observation, if, for each individual, we subtract (4.1) when $D = 1$ from itself when $D = 0$, this would collapse to:

$$y_{i1} - y_{i0} = \delta + \varepsilon_{i1} - \varepsilon_{i0}$$

where the $i1$ subscript is individual i when $D = 1$ and similarly for $i0$ and $D = 0$. The demographics disappear so we don't even need to collect them. The new error term $\varepsilon_{i1} - \varepsilon_{i0}$ is likely to have a substantially smaller variance than the cross section variance in (4.1) since it eliminates any remaining heterogeneity across individuals. And the test can be done using a simple t -statistic on a one-sample mean. This is a very simple case of the use of a panel data set with 2 observations per individual and in particular, is the simplest case of *fixed effects estimation*.

We could get much better inference about δ if we could do this, but in many (or most) cases, we can't. One individual can't be observed both ways simultaneously so we would either have to choose them at the start and observe them before and after, or choose them at the end and try to retrospectively gather the information from the previous situation. The first method would be a controlled study, which has a number of potential problems:

1. If individuals can choose whether or not to take part, they are more likely to participate if they think the change would be beneficial.
2. If individuals can choose whether to finish the study, they are more likely to drop if they find the change isn't beneficial.
3. Simply participating in a study can sometimes have a "placebo effect", of changing behavior or outlook even if the change should have no effect.

All three of these would work towards pushing up the observed value of δ (assuming positive is "good"), all because $E(\varepsilon_{i1}|\text{insample}) > 0$. The "gold standard" for testing is the double-blind, where neither the subject nor the researcher knows whether a given individual is actually receiving the treatment, which allows control for (3) while somewhat mitigating (1). However, that's almost impossible to arrange in economics or business, since people aren't getting something that's easily disguised.

The alternative of sampling at the end has a similar (and even stronger) sample selection problem if the value of D is a choice. People will likely choose $D = 1$ if $D = 1$ is better for them than $D = 0$ —this will bias the estimate of the difference.

In short, this simple type of panel data estimation is likely to be feasible only if the value of D is out of the hands of the individual, for instance, a change in tax or wage rate that applies to everyone. If you can get pre- and post-intervention data, that will likely allow valid inference without careful handling of the sample selection problems.

4.1 A More Realistic Example

Suppose we have a set of firms with what is assumed to be a relatively similar Cobb-Douglas production technology:

$$\log q_i = A + \alpha \log K_i + (1 - \alpha) \log L_i + \varepsilon_i$$

We assume that the production and labor input can reasonably be measured, but capital (as is typical) really isn't observable. If we run the regression that we can:

$$\log q_i = A^* + \beta \log L_i + \varepsilon_i^*$$

we have a serious problem with omitted variables bias. Firms with more capital generally have more labor as well so the estimate of β will be strongly biased upwards. In fact, if labor is roughly proportional to capital across firms, the estimate of β should be around 1.0. From a cross-section data set, we have very little hope of estimating β at all well.

Suppose, however, that we have a panel data set. Now, we have:

$$\log q_{it} = A + \alpha \log K_{it} + (1 - \alpha) \log L_{it} + \varepsilon_{it}$$

Assume also that this is over a short enough time period that K_{it} is roughly constant at K_i . Collapse the overall intercept and the capital term into the individual-specific intercept:

$$A_i^* = A + \alpha \log K_i$$

Now, we have

$$\log q_{it} = A_i^* + \beta \log L_{it} + \varepsilon_{it} \tag{4.2}$$

This is now a perfectly good regression, with $N + 1$ free coefficients—the N intercepts and β . Since we now have NT data points, we have more than enough to estimate those. If you estimate the model in this form, it's called *Least Squares Dummy Variables* or LSDV since we need to add dummy variables for each individual to the other regressors.

LSDV is sometimes useful, but if N is large, it requires inverting a very large matrix without taking advantage of its special structure. An alternative which

gives exactly the same result for β and all the summary statistics is to subtract individual means from each part of the equation. For the variable x_{it} , let $x_{i\bullet}$ represent the average of x across t at i . The transformed regression is:

$$\log q_{it} - (\log q)_{i\bullet} = \beta (\log L_{it} - (\log L)_{i\bullet}) + (\varepsilon_{it} - \varepsilon_{i\bullet}) \quad (4.3)$$

The A_i^* drop out since they are constant across t at i and thus $A_i^* = A_{i\bullet}^*$. The same would be true of *any* time-invariant regressor, so if we were actually interested in the coefficient on a time-invariant regressor, this wouldn't be the way to estimate it. Note also that we wrote the variables in the form $(\log q)_{i\bullet}$ to make clear that we need to average the log, not log the average. The transformation has to be applied to the variables in the form in which they appear in the regression.

That (4.2) and (4.3) give exactly the same results for the estimate of β follows from the Frisch-Waugh Theorem on partitioned least squares—the regression in matrix form

$$\mathbf{Y} = \mathbf{Z}\gamma + \mathbf{X}\beta + \mathbf{u}$$

can be transformed to

$$\mathbf{Q_Z Y} = \mathbf{Q_Z X}\beta + \mathbf{Q_Z u}$$

where

$$\mathbf{Q_Z} = \mathbf{I} - \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'$$

is the residual projection operator. This is true for any set of regressors \mathbf{Z} , but helps tremendously when, as in this case, the \mathbf{Z} represents a set of mutually exclusive dummies, as then $\mathbf{Z}'\mathbf{Z}$ is a diagonal matrix (with the count of observations for each dummy on the diagonal) and thus can be very easily inverted.¹

One thing to note, however, is that if you compare the output from running (4.2) and (4.3), the estimates of β will match, as will the sums of squared residuals, but the standard errors won't match. Doing the sequential estimation doesn't take into account the fact that N degrees of freedom have been removed from the data in taking the deviations from the mean. With RATS, you can correct for this by using the DFC (Degrees of Freedom Correction) option on **LINREG**. This makes a correction for the degrees of freedom in the dependent variable.

4.2 Balanced vs Unbalanced Samples

There are quite a few panel data estimation methods which are much simpler to compute for balanced samples than they are for unbalanced. Fixed individual effects is *not* one of them. Everything said above applies to balanced or unbalanced samples. The only minor difference is if a particular data set is so

¹For a general \mathbf{Z} matrix, there is little, if any, computational gain in doing the regression this way since inverting $\mathbf{Z}'\mathbf{Z}$ would take almost the same amount of time as inverting the full cross product matrix of \mathbf{Z} and \mathbf{X} together.

unbalanced that some individuals have no data in a particular regression, the degrees of freedom correction is reduced by one for each such individual. If you do LSDV with a data set like that, you'll get a 0 coefficient with 0 standard error for the dummy on each such individual (and that "coefficient" won't be counted in the degrees of freedom adjustments).²

4.3 Implementing with RATS

We'll show how to use RATS to do both (4.2) and (4.3) as written, as there are specialized situations where we will need them, but, in practice, you will be using the **PREGRESS** (Panel REGRESS) instruction with the option **METHOD=FIXED**. Since it's the most common form of panel regression, fixed effects is also the default, so if you do **PREGRESS**, you'll be doing fixed individual effects.

In Example 4.1, we will be using Grunfeld's investment data, which is used as the main example of techniques in Baltagi (2008) and is also used in Greene (2007) and in Koop (2003) in demonstrating Bayesian methods on panel data. As it is generally used, this data set has annual data for 10 firms running from 1935 to 1954. The regression of interest is gross investment on the firm's (prior year) market capitalization and capital stock.³ All variables are in real terms.

We will set this up as a **PANEL**-dated set. The data setup is:

```
open data grunfeld.xls
calendar(panelobs=20,a) 1935
all 10//1954:01
data(format=xls,org=columns) 1//1935:01 10//1954:01 firm year $
    invest value cap
```

Pooled sample estimation

You can run a simple linear regression on the model using either **LINREG** or **PREGRESS (METHOD=POOLED)**. There's no difference in the calculations; it's just a matter of preference if you're working with panel data. There's just a slight difference in the header on the output, with this being from using **METHOD=POOLED**. Note that we put the **CONSTANT** last rather than the usual first. We'll see why in a moment.

²RATS automatically adjusts for collinearity in the regressor list, and a 0 regressor is the simplest example of that.

³The fact that the regressors are prior year values is sometimes left out of the description. There would clearly be simultaneity if they were current.

Panel Regression - Estimation by Pooled Sample				
Dependent Variable INVEST				
Panel(20) of Annual Data From		1//1935:01 To	10//1954:01	
Usable Observations		200		
Degrees of Freedom		197		
Centered R^2		0.8124080		
R-Bar^2		0.8105035		
Uncentered R^2		0.8710896		
Mean of Dependent Variable		145.95825000		
Std Error of Dependent Variable		216.87529623		
Standard Error of Estimate		94.40840333		
Sum of Squared Residuals		1755850.4841		
Regression F(2,197)		426.5757		
Significance Level of F		0.0000000		
Log Likelihood		-1191.8024		
Variable	Coeff	Std Error	T-Stat	Signif

1. VALUE	0.11556216	0.00583571	19.80259	0.00000000
2. CAP	0.23067849	0.02547580	9.05481	0.00000000
3. Constant	-42.71436944	9.51167603	-4.49073	0.00001207

We get the rather unsurprising result that bigger, more valuable firms invest more. However, these are companies in several different industries (automobile, steel, power plants, refining) with different technologies.⁴ We would probably expect that a regression that didn't allow for individual effects would be misspecified. Even if the omitted effects weren't correlated with the regressors (and thus the regression coefficients would still be unbiased), we would be working with much noisier residuals, and thus would be losing efficiency. You can do a quick (informal) test for whether the residuals from OLS are mean zero with

```
pstats(tests) %resids
```

This is informal because it doesn't take into account the fact that the input series are residuals and thus won't satisfy the usual independence assumptions.

Analysis of Variance for Series %RESIDS					
Source	Sum of Squares	Degrees	Mean Square	F-Statistic	Signif Level
INDIV	1168927.0516682	9	129880.7835187	42.045	0.000000
ERROR	586923.4324217	190	3089.0706970		
TOTAL	1755850.4840899	200			

This is significant beyond any measure. So adding fixed effects seems appropriate.

PREGRESS with METHOD=FIXED

If we keep the same list of regressors with **PREGRESS (METHOD=FIXED)**

```
preg(method=fixed) invest
# value cap constant
```

we get

⁴In order, the firms are GM, US Steel, General Electric, Chrysler, Atlantic Refining, IBM, Union Oil, Westinghouse, Goodyear, Diamond Match. These are listed in order of average investment from high to low. Diamond Match is way out of scale compared to the others with a very high market value relative to investment.

Panel Regression - Estimation by Fixed Effects				
Dependent Variable INVEST				
Panel(20) of Annual Data From 1//1935:01 To 10//1954:01				
Usable Observations 200				
Degrees of Freedom 188				
Centered R ² 0.9440725				
R-Bar ² 0.9408002				
Uncentered R ² 0.9615675				
Mean of Dependent Variable 145.95825000				
Std Error of Dependent Variable 216.87529623				
Standard Error of Estimate 52.76796595				
Sum of Squared Residuals 523478.14739				
Regression F(11,188) 288.4996				
Significance Level of F 0.0000000				
Log Likelihood -1070.7810				
Variable	Coeff	Std Error	T-Stat	Signif

1. VALUE	0.1101238041	0.0118566942	9.28790	0.00000000
2. CAP	0.3100653413	0.0173545028	17.86656	0.00000000
3. Constant	0.0000000000	0.0000000000	0.00000	0.00000000

The `CONSTANT` gets zeroed out by the subtracting means, producing the 0's across the board in the output. We recommend that you keep the time-invariant variables like this in, rather than manually altering the regressor lists for fixed effects versus other techniques which *can* estimate these coefficients. The result is the same, but you have less work to do. Note that the degrees of freedom are properly adjusted: one lost for each individual, plus two for the time-varying regressors. In comparing the results of the two regressions, we see there has been quite a substantial change in the coefficient on the capital stock. It's increased by between three and four standard errors, so it seems rather clear that the omitted individual effects in least squares were negatively correlated with capital.

LSDV Regression

To do the LSDV regression, we need to generate a set of dummy variables. This is now very easy to do with `PANEL` with the option `DUMMIES`.⁵ This generates a `VECTOR` of `SERIES` of the individual dummies. That can be inserted into a regression list which will include the whole set. The LSDV regression is done by:

```
panel(dummies=idummies)
*
linreg(title="Least Squares Dummy Variable") invest
# value cap idummies
```

The output that this produces is:

⁵This was added with version 8.1.

Linear Regression - Estimation by Least Squares Dummy Variable				
Dependent Variable INVEST				
Panel(20) of Annual Data From 1//1935:01 To 10//1954:01				
Usable Observations 200				
Degrees of Freedom 188				
Centered R^2 0.9440725				
R-Bar^2 0.9408002				
Uncentered R^2 0.9615675				
Mean of Dependent Variable 145.95825000				
Std Error of Dependent Variable 216.87529623				
Standard Error of Estimate 52.76796595				
Sum of Squared Residuals 523478.14739				
Regression F(11,188) 288.4996				
Significance Level of F 0.0000000				
Log Likelihood -1070.7810				
Durbin-Watson Statistic 1.0789				
Variable	Coeff	Std Error	T-Stat	Signif

1. VALUE	0.1101238	0.0118567	9.28790	0.00000000
2. CAP	0.3100653	0.0173545	17.86656	0.00000000
3. IDUMMIES(1)	-70.2967175	49.7079588	-1.41419	0.15895875
4. IDUMMIES(2)	101.9058137	24.9383232	4.08631	0.00006485
5. IDUMMIES(3)	-235.5718410	24.4316165	-9.64209	0.00000000
6. IDUMMIES(4)	-27.8092946	14.0777538	-1.97541	0.04968535
7. IDUMMIES(5)	-114.6168128	14.1654333	-8.09130	0.00000000
8. IDUMMIES(6)	-23.1612951	12.6687393	-1.82822	0.06910077
9. IDUMMIES(7)	-66.5534735	12.8429734	-5.18209	0.00000056
10. IDUMMIES(8)	-57.5456573	13.9931464	-4.11242	0.00005848
11. IDUMMIES(9)	-87.2222724	12.8918932	-6.76567	0.00000000
12. IDUMMIES(10)	-6.5678435	11.8268910	-0.55533	0.57932822

Note that the dummy coefficients are individual intercepts, not differentials. They are overwhelming negative because the intercept in the original regression was negative.

Within Regression

The regression in (4.3) is known as a *within* regression. This is from terminology from analysis of variance, where we write

$$x_{it} - \bar{x} = (x_{it} - x_{i\bullet}) + (x_{i\bullet} - \bar{x})$$

and can write the sums of squares as:

$$\sum_{it} (x_{it} - \bar{x})^2 = \sum_{it} (x_{it} - x_{i\bullet})^2 + \sum_{it} (x_{i\bullet} - \bar{x})^2$$

as the cross terms drop out. The first term in this is the “within” (individuals) variation and the second is the “between” variation. The instruction for transforming a series to its within individual deviation is **PANEL** with the options **ENTRY=1.0** and **INDIV=-1.0**. This needs to be done to the dependent variable and each of the (time-varying) regressors. In our case,

```
panel(entry=1.0,indiv=-1.0) invest / cinvest
panel(entry=1.0,indiv=-1.0) value / cvalue
panel(entry=1.0,indiv=-1.0) cap / ccap
*
linreg(title="Within Regression",dfc=10) cinvest
# cvalue ccap
```

Obviously, this will require more programming specific to the problem at hand than the other two ways to accomplish the same estimation. You also have to be *very* careful about the de-meaning process if there are any missing values, whether those are present in the original data, or are created in the course of transforming data. For instance, suppose that we don't have the `CAP` value in period 1 for each individual, but have data for the other two series. The **PANEL** instruction above will correctly create `CCAP` which has a missing value for period 1, and the others demeaned using the 19 valid data points per individual. However, `CINVEST` and `CVALUE` will have a full set of 20 entries per individual, and will be demeaned using all 20. Using those series in a regression will *not* give you the correct estimates. If there's any question, you should use a common `SMPL` option across all the **PANEL** instructions. The following (which is at the end of the example) does a version of `CAP` with missing values at each period 1. The first step is to run a simple regression; the valid entries in the residual mark the data points that we need to use in the **PANEL** instructions. You don't need to repeat the `SMPL` option on the **LINREG** since the transformed data will have NA at the points that are "SMPL'ed" out.

```
set capna = %if(%indiv(t)==1,%na,cap)
*
linreg invest
# value capna constant
set psmpl = %valid(%resids)
*
panel(entry=1.0,indiv=-1.0,smpl=psmpl) invest / cinvest
panel(entry=1.0,indiv=-1.0,smpl=psmpl) value / cvalue
panel(entry=1.0,indiv=-1.0,smpl=psmpl) capna / ccap
*
linreg(title="Within Regression",dfc=10) cinvest
# cvalue ccap
```

4.4 Testing Fixed Effects

Fixed effects allows for heterogeneous intercepts, while assuming homogeneous slope coefficients. As such, it sits between two other possible estimators: pooled least squares, which assumes that *all* the coefficients, including the intercept, are homogeneous, and a fully heterogeneous regression. Either direction is formally testable, at least as long as there are enough data points per individual for the full sample split. The testing method is a standard *F* test. We need to save sums of squared residuals and degrees of freedom for each type of regression. We've already seen how to do pooled least squares and fixed effects. With the addition of the **COMPUTE** instructions to save the required statistics, these are done with:

```

preg(method=pooled) invest
# value cap constant
compute rsspools=%rss,ndfpools=%ndf
preg(method=fixed) invest
# value cap constant
compute rssfe=%rss,ndffe=%ndf

```

The fully split regression is most easily done with **SWEEP**. This is done with:

```

sweep(group=%indiv(t))
# invest
# value cap constant
compute rssfull=%sigma(1,1)*%nobs,ndffull=%nobs-%nregsystem

```

The one “target” variable for **SWEEP** is the dependent variable, and the three “instruments” are the three regressors. This does separate regressions for each individual (the `GROUP` option gives a series or formula which has a separate value for each desired group). Internally, it aggregates them into a single estimate for the coefficients, but we aren’t interested in that here. Instead, we need the sum of squared residuals and the degrees of freedom. The sum of squared residuals is gotten indirectly from the `%SIGMA` matrix—**SWEEP** is designed to work on more than one target, so it produces a covariance *matrix* rather than a single number. `%NREGSYSTEM` counts all the regressors across groups and targets, which will here be 30. The three pairwise test for these are done with:

```

compute fstat=((rsspools-rssfull)/(ndfpools-ndffull))/(rssfull/ndffull)
cdf(title="Chow Test for Pooling") ftest fstat $
ndfpools-ndffull ndffull
compute fstat=((rssfe-rssfull)/(ndffe-ndffull))/(rssfull/ndffull)
cdf(title="Chow Test for Pooling, Slopes Only") ftest fstat $
ndffe-ndffull ndffull
compute fstat=((rsspools-rssfe)/(ndfpools-ndffe))/(rssfe/ndffe)
cdf(title="Test for Fixed Effects vs Pooling") ftest fstat $
ndfpools-ndffe ndffe

```

The output shows rather strongly that the basic model (assuming homogeneous slopes) doesn’t appear to be adequate. It’s very clear that the individual effects are necessary (last test), but even given heterogeneous intercepts, the data don’t support homogeneity in the slopes (second test).

Chow Test for Pooling	
F(27,170)=	27.74861 with Significance Level 0.00000000
Chow Test for Pooling, Slopes Only	
F(18,170)=	5.78046 with Significance Level 0.00000000
Test for Fixed Effects vs Pooling	
F(9,188)=	49.17663 with Significance Level 0.00000000

4.5 METHOD=BETWEEN

Baltagi and some other books that describe other methods of estimating linear regressions with panel data include the *between* estimates, which, rather than replacing the regressions with deviations from individual means, use the individual means themselves. Rather than reducing or eliminating the fixed effects, this enhances them, as they are unaffected by the averaging, while the time-varying parts of the regression have their variability reduced. This can be useful for certain calculations where we need at least two distinct estimates of the extent of the individual effects, but they have little value as an actual method of estimation of the model of interest. If, for whatever reason, you need the between estimates, use **PREGRESS** with **METHOD=BETWEEN**.

Example 4.1 Fixed Effects

This demonstrates various ways to handle fixed effects estimation, both using **PREGRESS** and direct estimation with dummies. The detailed description start with section 4.3.

```
open data grunfeld.xls
calendar(panelobs=20,a) 1935
all 10//1954:01
data(format=xls,org=columns) 1//1935:01 10//1954:01 firm year $
    invest value cap
*
* In order, the firms are GM, US Steel, General Electric, Chrysler,
* Atlantic Refining, IBM, Union Oil, Westinghouse, Goodyear, Diamond
* Match.
*
* The following two instructions are equivalent
*
linreg invest
# value cap constant
preg(method=pooled) invest
# value cap constant
*
pstats(tests) %resids
*
preg(method=fixed,indiv=ieffects) invest
# value cap constant
*
* LSDV
*
panel(dummies=idummies)
*
linreg(title="Least Squares Dummy Variable") invest
# value cap idummies
*
* Within regression
*
```

```

panel(entry=1.0, indiv=-1.0) invest / cinvest
panel(entry=1.0, indiv=-1.0) value / cvalue
panel(entry=1.0, indiv=-1.0) cap / ccap
*
linreg(title="Within Regression", dfc=10) cinvest
# cvalue ccap
*****
*
* Testing
*
* Compute fully pooled regression (linreg)
*
preg(method=pooled) invest
# value cap constant
compute rsspools=%rss, ndfpools=%ndf
*
* Compute regression pooled on slopes only (FE)
*
preg(method=fixed) invest
# value cap constant
compute rssfe=%rss, ndffe=%ndf
*
* Compute fully separated regression
*
sweep(group=%indiv(t), coeffs=fullcoeff)
# invest
# value cap constant
compute rssfull=%sigma(1,1)*%nobs, ndffull=%nobs-%nregsystem
*
* Compare fully pooled with fully separated
*
compute fstat=((rsspools-rssfull)/(ndfpools-ndffull))/(rssfull/ndffull)
cdf(title="Chow Test for Pooling") ftest fstat $
ndfpools-ndffull ndffull
*
* Compare FE with fully separated
*
compute fstat=((rssfe-rssfull)/(ndffe-ndffull))/(rssfull/ndffull)
cdf(title="Chow Test for Pooling, Slopes Only") ftest fstat $
ndffe-ndffull ndffull
*
* Compare FE with pooled
*
compute fstat=((rsspools-rssfe)/(ndfpools-ndffe))/(rssfe/ndffe)
cdf(title="Test for Fixed Effects vs Pooling") ftest fstat $
ndfpools-ndffe ndffe
*
* Example with missing data
*
set capna = %if(%indiv(t)==1,%na,cap)
*
linreg invest
# value capna constant
set psmpl = %valid(%resids)

```



```
*  
panel(entry=1.0, indiv=-1.0, smpl=psmpl) invest / cinvest  
panel(entry=1.0, indiv=-1.0, smpl=psmpl) value / cvalue  
panel(entry=1.0, indiv=-1.0, smpl=psmpl) capna / ccap  
*  
linreg(title="Within Regression", dfc=10) cinvest  
# cvalue ccap
```

Random Effects

The one obvious drawback with the fixed effects estimator is that we can't estimate coefficients on time-invariant regressors. If one or more of those are of interest, fixed effects won't be useful. An alternative method that *does* allow estimation for coefficients on both time-varying and time-invariant regressors is *random effects*. In the random effects model, the individual effect is shifted from an individual-specific intercept into the error term, as in:

$$y_{it} = \alpha + Z_i\gamma + X_{it}\beta + (\mu_i + \eta_{it}) \quad (5.1)$$

The Z are the *time-invariant regressors* and X are the time-varying ones. In the error term, μ is the systematic individual shift (assumed to be uncorrelated across individuals) and η is the idiosyncratic error (uncorrelated across both individuals and time, and uncorrelated with μ). To give content to this rearrangement of the regression, we need to make assumptions about the behavior of the error term. To start, we'll assume that the regressors are pre-determined, so we have the assumptions:

$$EZ_i\mu_i = EZ_i\eta_{it} = 0; EX_{it}\mu_i = EX_{it}\eta_{it} = 0 \quad (5.2)$$

It's the assumptions on the μ that change the behavior of the estimates. For instance, the fixed individual effects for all women in the sample could have been systematically higher or lower than those for the men because the individual effects can take any values. In the random effects model, however,

$$E(\mu_i | i \text{ is female}) = 0$$

so the coefficient on a female dummy would have to adjust to make that happen.

The assumptions about the individual effect are quite strong, and there are many reasons why they might not hold in practice. Unobservable individual effects will often be expected to include such things as ability, expertise and ambition which would be expected to be correlated with explanatory variables which represent choices, such as level of education. However, for now, we'll assume that the assumptions are not unreasonable. Also, note that the same problem would be present in a cross-sectional data set on the same variables—panel data will actually give us a way to handle the endogenous regressors that would not be available in a simple cross section.

5.1 The Random Effects Estimator

Unlike the Fixed Effects estimator (FE for short), the Random Effects (RE) estimator is easier to derive and compute if the sample is balanced, so we'll start with that assumption. We'll also assume that the sample is blocked in our usual fashion, with the observations for an individual grouped together. If, in (5.1), σ_μ^2 is the variance of μ_i and σ_η^2 is the variance of η_{it} , then the covariance matrix of the error term for individual i takes the form:

$$\sigma_\eta^2 I_T + \sigma_\mu^2 J_T \quad (5.3)$$

where I_T is the identity matrix of size T and J_T is a matrix of all ones of size T . J_T is a rank one matrix which can conveniently be written $J_T = \iota_T \iota_T'$, where ι_T is a size T (column) vector of all ones. This has the properties that:

$$J_T J_T = \iota_T \iota_T' \iota_T \iota_T' = \iota_T (\iota_T' \iota_T) \iota_T' = T J_T$$

and

$$\left(\frac{1}{T} J_T\right) \left(\frac{1}{T} J_T\right) = \frac{1}{T^2} J_T J_T = \frac{1}{T} J_T$$

A convenient way to rewrite (5.3) is

$$\begin{aligned} \Omega_T &\equiv \sigma_\eta^2 I_T + \sigma_\mu^2 J_T = \sigma_\eta^2 \left(I_T - \frac{1}{T} J_T \right) + \sigma_\mu^2 J_T + \sigma_\eta^2 \frac{1}{T} J_T \\ &= \sigma_\eta^2 \left(I_T - \frac{1}{T} J_T \right) + (T\sigma_\mu^2 + \sigma_\eta^2) \frac{1}{T} J_T \end{aligned}$$

This is convenient because $(I_T - \frac{1}{T} J_T)$ and $\frac{1}{T} J_T$ are

1. symmetric
2. idempotent (they are equal to their own squares)
3. orthogonal to each other (product is zero matrix)

This is known as the *spectral decomposition* of Ω_T , which always exists for symmetric positive semi-definite matrices.¹ The advantage of writing the matrix this way is that powers of Ω_T (positive or negative, integer or non-integer) can be computed as:

$$\Omega_T^r = (\sigma_\eta^2)^r \left(I_T - \frac{1}{T} J_T \right) + (T\sigma_\mu^2 + \sigma_\eta^2)^r \left(\frac{1}{T} J_T \right) \quad (5.4)$$

that is, can be done with no additional calculations besides a couple on scalars.

The standard way to handle a GLS estimator is to find a matrix F such that $F'F = \Omega^{-1}$, and transform the regressors and dependent variable by pre-multiplication by F , which converts the regression to one with uncorrelated errors.²

¹You can rewrite the better-known eigen decomposition in this form: $A = P\Lambda P' = \sum \lambda_i P_i P_i'$, where the P_i are the columns of P (eigenvectors). That the $P_i P_i'$ are idempotent and orthogonal to each other follows because $PP' = I$.

²More specifically, uncorrelated with identity covariance matrix.

In this case, the full covariance matrix for the sample across individuals is block-diagonal (since the $\mu_i + \eta_{it}$ are uncorrelated across i), so this transformation can be done block by block. In general, there are many matrices which solve $\mathbf{F}'\mathbf{F} = \Omega_T$. The easiest one to compute here is to make \mathbf{F} the result of applying (5.4) with $r = -1/2$. This generates a symmetric \mathbf{F} . We can rewrite this as:

$$\begin{aligned}\mathbf{F} &= (\sigma_\eta^2)^{-1/2} \left(I_T - \frac{1}{T} J_T \right) + (T\sigma_\mu^2 + \sigma_\eta^2)^{-1/2} \left(\frac{1}{T} J_T \right) \\ &= (\sigma_\eta^2)^{-1/2} \left\{ \left(I_T - \frac{1}{T} J_T \right) + \left(T \frac{\sigma_\mu^2}{\sigma_\eta^2} + 1 \right)^{-1/2} \left(\frac{1}{T} J_T \right) \right\}\end{aligned}\quad (5.5)$$

The expression in braces consists of the “within” individual deviation operator used in fixed effects, plus the weight $\left(T \frac{\sigma_\mu^2}{\sigma_\eta^2} + 1 \right)^{-1/2}$ times the individual mean itself. The first term will still drop out for a time-invariant regressor, but the second won’t.

The estimators for β and γ will depend only upon the ratio $\sigma_\mu^2/\sigma_\eta^2$ since the lead $(\sigma_\eta^2)^{-1/2}$ just multiplies the whole transformed data set. As $\sigma_\mu^2/\sigma_\eta^2 \rightarrow 0$, the coefficient on the second term goes to 1, so overall (5.5) converges to a constant times the identity, which will just give OLS. As $\sigma_\mu^2/\sigma_\eta^2 \rightarrow \infty$, the coefficient on the second term goes to 0 due to the $-1/2$ power—this leaves us with FE. Thus OLS and FE are special cases of random effects, depending upon relative variances.

One other thing to note is that Ω_T is a positive definite matrix as long as σ_η^2 and $T\sigma_\mu^2 + \sigma_\eta^2$ are positive. This does not require that σ_μ^2 itself be positive—as long as it is relatively small compared with σ_η^2 , it can be negative and still give us a valid GLS estimator. Since $\sigma_\mu^2 < 0$ makes no sense³, it’s possible that we will have a solution with $\sigma_\mu^2 = 0$ that’s on the boundary of the space of values permitted by our interpretation of the model.

The RE estimator can be computed in RATS using the instruction **PREGRESS**, now with the option **METHOD=RANDOM**. However, what we’ve described so far isn’t really feasible since σ_η^2 and σ_μ^2 aren’t, in practice, known. There have been many different proposals for ways to estimate these and we have implemented most of them in **PREGRESS**.

5.2 Estimating the Component Variances

We can’t apply the transformation to do GLS until we have values for the variances. For simplicity, we’ll first assume that we can observe the true residuals:

$$u_{it} = \mu_i + \eta_{it}$$

³Though it indicates that the individual effects aren’t very important.

Then

$$Eu_{it}^2 = \sigma_\mu^2 + \sigma_\eta^2$$

Summing this over the data set gives one equation in the two variances:

$$E \sum_{i,t} u_{it}^2 = \sigma_\mu^2 (NT) + \sigma_\eta^2 (NT) \quad (5.6)$$

Almost any distinct (from the identity) quadratic form in the u vector can yield a second equation. What will be (in the end) the simplest is:

$$\begin{aligned} E \sum_i \frac{1}{T} \left(\sum_t u_{it} \right)^2 &= \sum_i \frac{1}{T} E \left(\sum_t (\mu_i + \eta_{it}) \right)^2 \\ &= \sum_i \frac{1}{T} E \left(T\mu_i + \sum_t \eta_{it} \right)^2 \\ &= \sigma_\mu^2 (NT) + \sigma_\eta^2 N \end{aligned} \quad (5.7)$$

Summing across t before squaring reduces the relative variance from the random part thus giving different multipliers. Replacing expectations with sample values and solving (5.6) and (5.7) simultaneously gives an estimate for the two components.

This calculation can be done using the instruction **PSTATS**.

pstats u

produces no direct output, but sets the variables `%VINDIV` and `%VRANDOM` to the individual-specific variance (σ_μ^2) and purely random variance (σ_η^2) respectively.

Allowing for Sampling Error

In practice, the residuals aren't known, and must be estimated. This creates a small-sample correlation which can be considerable in some cases. There are also several different ways to estimate the regression equation—different estimates for β and γ will give us different estimates for the variances. The technical details of how the variances can be estimated using the three most common procedures are in Appendix A.

As above, we need two distinct conditions to estimate two variances. Some of these procedures use two estimators for the regression, while others use two weightings for a single set of residuals. The most straightforward applies the above analysis using the OLS residuals with small-sample corrections—this is known as the *Wallace-Hussain* method. Wooldridge (2010) gives a simplified version of this, which, unfortunately, isn't quite correct in how it (approximately) adjusts for small samples—he subtracts K where he needs to subtract KT .⁴

⁴If you read the discussion in Wooldridge, it doesn't *look* like it's the same, but actually is.

There are three main alternatives to Wallace-Hussain among “two-step” estimators. *Wansbeek-Kapteyn* (or *Amemiya*), *Swamy-Arora* and *Fuller-Battese* estimate σ_η^2 directly by running fixed effects. Since fixed effects eliminates the individual effect entirely, the sum of squared FE residuals divided by the NT less N (for the individual intercepts) less the number of time-varying regressors (time-invariant regressors also being eliminated by FE) will be an estimate of σ_η^2 . This is just the standard unbiased estimator for the variance from FE. For Fuller-Battese, the second condition comes from the sum of squared OLS residuals as for Wallace-Hussain. For Swamy-Arora, the second condition comes from analyzing the sum of squared residuals from the between estimates. The default for RATS is Wansbeek-Kapteyn (WK for short). For that, the second condition comes out of the “raw” residuals from FE, $y_{it} - X_{it}\hat{\beta}_{FE}$, using the original data, not the deviations from individual means. This puts the individual effects back in. Because these aren’t mean zero (since the intercept is eliminated by FE), they are centered by subtracting the full-sample mean. The sums of squared individual averages (the analogue of (5.7)) are now computed for those new residuals.

While Wallace-Hussain (WH) is conceptually simpler, requiring only a preliminary OLS estimate, OLS is quite inefficient in the presence of strong individual effects, so there’s a good reason to believe that it will provide rather imprecise estimates.⁵ We chose WK as the default because it requires only the one set of estimates, which, of the three (OLS, FE and between) will generally be the most efficient in the presence of strong individual effects.

As we said, these are the “two-step” estimators. It’s also possible to estimate the variances by maximum likelihood. Twenty or more years ago, when the methods described above were proposed, this was infeasible for many models simply because of the time required, since each function evaluation requires computing a full RE estimate. However, that’s no longer the case except in very large data sets with unbalanced panels.

5.3 Using RATS

Example 5.1 is based upon an example from Greene (2007). This estimates a wage equation using data extracted from the PSID. The data are read with:

```
open data cornwell&rupert.xls
calendar(panelobs=7,a) 1976
data(format=xls,org=columns) 1//1976:01 595//1982:01 exper wks $
  occ ind south smsa ms fem union ed blk lwage
*
set expersq = exper^2
```

⁵If the individual effects are weak, we wouldn’t expect much difference in the final RE estimates regardless of the method used. We will get something close to OLS.

We have 595 individuals with 7 annual observations for each. The explanatory variables are:

- EXPER (years of full-time work experience) (and its square)
- WKS (weeks worked, 0 if not)
- OCC (1 for blue-collar)
- IND (1 for works in manufacturing)
- SOUTH (1 for lives in south)
- SMSA (1 for lives in metropolitan area)
- MS (1 if married)
- UNION (1 if member of union)
- ED (years of education)
- FEM (1 if female)
- BLK (1 if black)

The one of special interest is education, which is time-invariant, so fixed effects won't be useful as an estimator. Two of the demographics are also time-invariant (FEM and BLK) while many of the others are almost so.

On **PREGRESS**, random effects is chosen with `METHOD=RANDOM`. You choose the method for estimating the component variances with the `VCOMP` (Variance COMPonent) option. For the different methods described, the choices are

- `VCOMP=WK` (Wansbeek-Kapteyn)
- `VCOMP=WH` (Wallace-Hussain)
- `VCOMP=SA` (Swamy-Arora)
- `VCOMP=FB` (Fuller-Battese)
- `VCOMP=ML` (maximum likelihood)

There are two other choices which we added to allow exact replication of examples from Wooldridge (2010) (`VCOMP=WOOLDRIDGE`) and Greene (2007) (`VCOMP=GREENE`). In practice, there's no need to use either of those.⁶ You can also input the variances directly, rather than having them estimated. Aside from choosing the algorithm with the `VCOMP` option, you can also input them using `VRANDOM=value` for σ_η^2 and `VINDIV=value` for σ_μ^2

The following estimates the regression using each of the principal methods:

⁶Greene's method is Fuller-Battese with approximate small-sample correction.

```

preg(method=random,vcomp=wk) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
preg(method=random,vcomp=wh) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
preg(method=random,vcomp=sa) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
preg(method=random,vcomp=fb) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
preg(method=random,vcomp=ml) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant

```

The header from the WK output is:

Panel Regression - Estimation by Random Effects		
Dependent Variable LWAGE		
Panel(7) of Annual Data From	1//1976:01 To	595//1982:01
Usable Observations	4165	
Degrees of Freedom	4152	
Mean of Dependent Variable	6.6763464010	
Std Error of Dependent Variable	0.4615121896	
Standard Error of Estimate	0.1414517021	
Sum of Squared Residuals	83.075640888	
Log Likelihood	289.6345	
S.D. (eta_it)	0.1520	
S.D. (mu_i)	1.0319	
Hausman Test (9)	419.721213	
Significance Level	0.0000000	

The two lines:

S.D. (eta_it)	0.1520
S.D. (mu_i)	1.0319

show the estimated standard errors (not variances) for the two components. You can also retrieve these (variances now) as the variables %VRANDOM and %VINDIV. The estimated standard deviations of the individual effects vary widely from method to method: WK is the highest at 1.0319, WH the lowest at 0.2567. The estimates of the return to education generally move with those estimates: the highest is .138 for WK (with the wages in log's, this means that one additional year of education increases wages by 13.8%), and the lowest is .082 for WH. These are clearly quite different from each other and are a reminder that simply reporting "random effects" estimates without including the method used to estimate the component variances isn't really adequate.

5.4 Hausman Test

If you look carefully at the output, you'll see that most of the tables include a line for the Hausman test. The variance component methods that use FE as part of the calculation will do this automatically. WH uses only OLS and so doesn't include the test. It also isn't included (automatically) if you input the variances with the VRANDOM and VINDIV options. If you want to make sure that you get a Hausman test, include the option HAUSMAN on the **PREGRESS** instruction. (This is only done when you're doing METHOD=RANDOM as well).

The Hausman test for RE is a special case of Hausman’s general strategy for specification tests. If the underlying assumptions for random effects (5.2) are correct (and the variances are correct as well), then RE is efficient as the Gauss-Markov estimator. FE is consistent (for β) under looser assumptions—it doesn’t require the assumptions about μ . FE is consistent either under the RE or FE assumptions; RE is consistent only if the RE assumptions hold. Hausman shows that under the null that the more restrictive assumptions hold, the covariance matrix of the difference between the two estimators is the difference between the two covariance matrices. Since, under the null, both estimators are consistent, we would expect them to be “similar”. Now, we can only test the time-varying regressors, since FE can’t estimate the time-invariant ones, so the degrees of freedom of the test is 9 (13 regressors less the fixed ones `CONSTANT`, `EDUC`, `FEM` and `BLK`). In every single case, the test is overwhelmingly significant, meaning that the individual effects appear to be correlated with the regressors. We can’t use FE to estimate the return to education, and it appears that we can’t use RE either.

Note that, if you want the Hausman test, you should let **PREGRESS** handle it. If you try to do the test by a direct comparison of fixed and random effects estimators (or use some other theoretically equivalent method), you can end up with a bad test statistic. The problem is that the two covariance matrices of the estimators need to be computed using identical Ω matrices. Only in those circumstances is the difference in the covariance matrices actually correct. In this case, it requires computing the covariance matrix of FE using the σ_η^2 from RE.

5.5 Direct Calculation of Component Variances

This section shows how to do the calculations required for the (small-sample corrected) Wallace-Hussain estimator, in case you need to do something similar, but not identical to this. The other calculations are fairly similar, and generally a bit less complicated. See Appendix A for the technical details and the notation used.

For WH, we need to start with least squares estimates. We need to compute the sum of squared residuals (which we could simply get with `%RSS`, but we’ll compute from the residuals anyway), and the sum of squared individual averages. The individual averages can be done using **PANEL** with the option `INDIV=1.0`. Since we need to solve a bivariate system of linear equations, we also set up the matrices for that. The sample statistics go into the `B` vector.

```
linreg lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
dec rect a(2,2)
dec vect b(2)
set olsresids = %resids
panel(indiv=1.0) olsresids / between
sstats / olsresids^2>>b(1) between^2>>b(2)
```

The matrix $X'JX$ is easy to compute, as that's what the `LWINDOW=`PANEL option is designed to do. The following use of `MCOV` will compute it:

```
mcov(lwindow=panel,lastreg,matrix=xjx)
```

For a balanced sample, we could get $X'JX$ by simply dividing XJX by the T dimension. The following is more general, and allows for an unbalanced sample. Because `MCOV` squares the input series, we need to use the reciprocal square root of the count.

```
panel(icount=1.0) olsresids / icounts
set counts = 1.0/sqrt(icounts)
mcov(lwindow=panel,lastreg,matrix=xjbarx) / counts
```

We now have all the information to fill in the **A** matrix of coefficients.⁷

```
compute xjxterm=%dot(%xx,xjx)
compute a(1,1)=%nobs-%nreg
compute a(1,2)=%nobs-xjxterm
compute a(2,1)=%panelsize()-%dot(%xx,xjbarx)
compute a(2,2)=%nobs-2*xjxterm+%dot(%xx*xjx*%xx,xjbarx)
```

The next solves the system of equations, displays the standard deviations computed from it and estimates the random effects regression with the input variances:

```
compute whvariances=%solve(a,b)
disp "Wallace-Hussian Standard Deviations" %sqrt(whvariances)
preg(method=random,vrandom=whvariances(1),vindiv=whvariances(2)) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
```

5.6 Random Effects Transformations

If you need to apply the random effects GLS transformation to a set of data, use the **PANEL** instruction, applied to each series in turn. If we look at (5.5), we see that the transformation depends not just upon the variances, but also upon T . For a unbalanced sample, the transformation will change from individual

⁷`%DOT(A,B)` is a quick way to compute the trace of a product as long as either matrix is symmetric.

to individual depending upon the value of T_i . For a balanced sample, we can rearrange the expression in brackets in (5.5) to

$$I_T - \left(1 - \left(T \frac{\sigma_\mu^2}{\sigma_\eta^2} + 1 \right)^{-1/2} \right) \left(\frac{1}{T} J_T \right)$$

If we define $\theta = 1 - \left(T \frac{\sigma_\mu^2}{\sigma_\eta^2} + 1 \right)^{-1/2}$, then **PANEL** with the options `ENTRY=1.0` and `INDIV=THETA` will do the required transformation. Beginning with RATS 8.1, a simpler and more general alternative is available: using the `VRANDOM` and `VINDIV` options to input the variance estimates σ_η^2 and σ_μ^2 . Using those, **PANEL** determines the proper transformation required for each individual.

The following shows how to apply this in a very general situation after getting estimates of the component variances. This takes care of arbitrary sets of regressors and possibly unbalanced samples:

```
equation(lastreg) panelreg
dec vect[series] re_rhs(%nreg)
do i=1,%nreg
    set temp = %eqnxvector(panelreg,t)(i)
    panel(vrandom=whvariances(1),vindiv=whvariances(2),$,
        smpl=%valid(%resids)) temp / re_rhs(i)
end do i
*
panel(vrandom=whvariances(1),vindiv=whvariances(2),$,
    smpl=%valid(%resids)) lwage / re_lwage
```

We now have the GLS transformations of the explanatory variables and the dependent variable. Now,

```
linreg(title="RE with Wallace-Hussain Estimates") re_lwage
# re_rhs
```

If you compare the output from this with the WH regression done using **PREGRESS**, you'll see that the coefficients are identical, but the standard errors aren't.⁸ **PREGRESS** uses the estimate of σ_η^2 from the WH procedure, while the **LINREG** computes a new value using the new residuals. The following will reformat the output to include the original regressor labels and match the **PREGRESS** standard errors.

```
compute %xx=whvariances(1)*%xx
linreg(create,equation=panelreg,covmat=%xx,form=chisquared,$
    title="RE with Wallace-Hussain Estimates") re_lwage
```

⁸The likelihood functions are very different, with the one from the **LINREG** being wrong for random effects as it doesn't take into account the determinant of Ω .

Example 5.1 Random Effects

This demonstrates estimation of a linear model by random effects. The detailed discussion starts on page 32.

```

open data cornwell&rupert.xls
calendar(panelobs=7,a) 1976
data(format=xls,org=columns) 1//1976:01 595//1982:01 exper wks occ ind $
  south smsa ms fem union ed blk lwage
*
set expersq = exper^2
*
preg(method=random,vcomp=wk) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
preg(method=random,vcomp=wh) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
preg(method=random,vcomp=sa) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
preg(method=random,vcomp=fb) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
preg(method=random,vcomp=ml) lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
*
* Wallace-Hussain estimator, computed directly.
*
linreg lwage
# exper expersq wks occ ind south smsa ms union ed fem blk constant
dec rect a(2,2)
dec vect b(2)
set olsresids = %resids
panel(indiv=1.0) olsresids / between
*
sstats / olsresids^2>>b(1) between^2>>b(2)
mcov(lwindow=panel,lastreg,matrix=xjx)
panel(icount=1.0) olsresids / icounts
set scounts = 1.0/sqrt(icounts)
mcov(lwindow=panel,lastreg,matrix=xjbarx) / scounts
compute xjxterm=%dot(%xx,xjx)
compute a(1,1)=%nobs-%nreg
compute a(1,2)=%nobs-xjxterm
compute a(2,1)=%panelsize()-%dot(%xx,xjbarx)
compute a(2,2)=%nobs-2*xjxterm+%dot(%xx*xjx*%xx,xjbarx)
compute whvariances=%solve(a,b)
disp "Wallace-Hussain Standard Deviations" %sqrt(whvariances)
*
* Manual transformations
* Do RE transformation on the right side variables from the regression
*
equation(lastreg) panelreg
dec vect[series] re_rhs(%nreg)
do i=1,%nreg
  set temp = %eqnxvector(panelreg,t)(i)
  panel(vrandom=whvariances(1),vindiv=whvariances(2),$

```

```
        smpl=%valid(%resids)) temp / re_rhs(i)
end do i
*
panel(vrandom=whvariances(1),vindiv=whvariances(2),$
      smpl=%valid(%resids)) lwage / re_lwage
linreg(title="RE with Wallace-Hussain Estimates") re_lwage
# re_rhs
*
* Reproduce PREGRESS standard errors and get a better formatted output.
*
compute %xx=whvariances(1)*%xx
linreg(create,equation=panelreg,covmat=%xx,form=chisquared,$
      title="RE with Wallace-Hussain Estimates") re_lwage
```

Two-Way Effects

The previous two chapters have looked only at individual effects. However, it's not unreasonable to think that there may also be some systematic change with time as well. In Example 5.1, for instance, it's hard to imagine that there *isn't* some time effect: it's a wage equation estimated during a period of high (and highly variable) inflation.

We'll use the following notation for a model with both individual and time effects:

$$u_{it} = \mu_i + \lambda_t + \eta_{it} \quad (6.1)$$

As a general rule, leaving out a time effect when it's necessary can lead to underestimating the variance of the individual component. Assume that λ_t is a fixed effect that nets to zero when summed over t . If we apply (A.1) with (6.1), we get (for a balanced sample):

$$E \sum_{i,t} u_{it}^2 = \sigma_\mu^2 (NT) + N \sum_t \lambda_t^2 + \sigma_\eta^2 (NT) \quad (6.2)$$

However, in (A.2), the time effect zeroes out in the inner sum, so that equation doesn't change, giving:

$$E \sum_i \frac{1}{T_i} \left(\sum_t u_{it} \right)^2 = \sigma_\mu^2 (NT) + \sigma_\eta^2 N \quad (6.3)$$

If we solve these without adjusting the sum of squares for the (unknown) $N \sum_t \lambda_t^2$, we will systematically end up with an overestimate of σ_η^2 and consequently (from (6.3)) an underestimate of σ_μ^2 .¹

The typical discussion of two-way effects in textbooks has both treated as fixed, or both as random. However, there isn't a compelling reason (other than simplicity) for assuming that. In two-way random effects, we need to add the orthogonality assumptions:

$$EZ_i \lambda_t = 0, EX_{it} \lambda_t = 0$$

¹With estimated coefficients, this is only a tendency—if there are sizable omitted time effects, the coefficient estimates could be so inefficient that the direction of error in the variance estimates isn't easily determined.

This isn't likely to be controversial. It's the assumption that $\lambda_t \sim N(0, \sigma_\lambda^2)$ *i.i.d.* that seems unlikely—time isn't randomly sampled, and the time effects probably aren't either.

In some cases, replacing general time effects with a linear trend might be appropriate. But if the “shape” of the response isn't quite as clear, a fixed effects approach to time is likely to be best, whether or not random effects are the choice for individual effects.

6.1 Balanced vs Unbalanced

Up until now, an unbalanced sample has been nothing more than a minor inconvenience—the notation is messier, and for the random effects model, the transformation changes from individual to individual based upon T_i . That's no longer the case with two-way models—the calculations are dramatically simplified when the sample is balanced, not just notationally, but in computational complexity. If we are doing two-way fixed effects, an obvious generalization to the within deviation is to take out individual means, then to take time means out of the individual deviations. If all works correctly, we take the series x_{it} to $x_{it} - x_{i\bullet}$ to

$$(x_{it} - x_{i\bullet}) - (x_{\bullet t} - x_{\bullet\bullet}) = x_{it} - x_{i\bullet} - x_{\bullet t} + x_{\bullet\bullet} \quad (6.4)$$

where $x_{\bullet\bullet}$ is the overall sample mean, which needs to be added back to make the result mean zero. We can always do the first step—it's the second step that fails when the sample is unbalanced, as the $x_{i\bullet}$ series should be NA for any i, t combination which doesn't exist in the data set and thus won't have the same set of values for different t . Even doing this explicitly in two steps doesn't get the results correct for unbalanced data, that is, it doesn't produce the same thing as residuals from regression on two-way dummies—the individual means are estimated at different levels of precision depending upon T_i and the joint dummy regression adjusts for that.

The simple calculation (6.4) requires a calculation of order NT for each i and t to get the means, then three more size NT to finish the transformation. Thus, overall, its computational complexity is $(N+T+3)*NT$, which is $O(N^2T+NT^2)$ operations. A brute force inversion of the size $N+T$ cross product matrix of the two-way dummies is $O((N+T)^3)$ which, if either $N \gg T$ or $T \gg N$ will be *much* larger. Use of a partitioned inverse (if the inner inverse is the larger dimension) can reduce the calculation to the same order of complexity, but with a much higher multiplier.²

If T is small, and you are doing fixed time effects (with either fixed or random individual effects), the simplest way to handle the calculation is to include time period dummies and just apply the desired individual effect method. This

²The two blocks on the diagonal in the dummy cross product matrix are diagonal matrices, and thus can be inverted easily individually.

is true whether the sample is balanced or not, and particularly true if it's unbalanced. In fact, Wooldridge (2010) never even discusses two-way effects, recommending the inclusion of time dummies as a matter of course.

6.2 Using RATS

The options on **PREGRESS** that allow for time effects are **EFFECTS=BOTH** (or **EFFECTS=TIME** in the unlikely event you want time effects only), with the **VTIME** option allowing input of the variance of the time effect. The computed value of the time effects variance is in the variable **%VTIME**. Example 6.1 uses a data set from Baltagi (2008) for analyzing gasoline consumption. This has 19 years' worth of annual data on 18 OECD countries over the period 1960-1978. The dependent variable is the log of gasoline consumption per car (**LGASPCAR**), with explanatory variables

1. **LINCOME** (log per capita income)
2. **LRPMG** (log real gasoline price)
3. **LCARPCAP** (log cars per capita)

The following reads the data and estimates with individual (only) random effects:

```
open data gasoline.xls
cal(panel=19) 1960
data(format=xls,org=cols) 1//1960:1 18//1978:1 year lgaspcar $
  lincomep lrpmg lcarpcap
preg(method=random,effects=indiv) lgaspcar
# constant lincomep lrpmg lcarpcap
```

The results are:

$$\begin{aligned} \log Gas/Car &= 2.188 & +0.602 \log Y/N & -0.366 \log P_{MG}/P_{GDP} & -0.621 \log CAR/N \\ & (0.216) & (0.066) & (0.042) & (0.027) \end{aligned} \quad (6.5)$$

The choice of variables is designed to at least somewhat mitigate the effects of time by using relative rather than absolute prices and per capita values to abstract from population growth. Some of the highest values for the dependent variable are in Turkey (throughout the sample) and Japan in the early 1960's as both have very low values of cars per capita—the **LCARPCAP** variable is added to the regressor list to take care of that.

However, if we add random time effects

```
preg(method=random,effects=both) lgaspcar
# constant lincomep lrpmg lcarpcap
```


we get the following:

$$\begin{array}{ccccccc} \log Gas/Car & = & 0.229 & +0.170 \log Y/N & -0.233 \log P_{MG}/P_{GDP} & -0.602 \log CAR/N & \\ & & (0.351) & (0.081) & (0.041) & (0.026) & \end{array} \quad (6.6)$$

The income and price elasticities are lower, with the income elasticity *substantially* lower. The variance of the individual effect goes from .118 with just individual effects to .179 with two-way effects, which is the expected direction of change.

The mixed model with fixed time effects and random individual effects can be done in two ways: one uses `EFFECTS=BOTH` and `METHOD=RANDOM` and with an input value of `%NA` (interpreted as “infinity”) for `VTIME` with `VINDIV` and `VRANDOM` taken from the preceding random effects regression:

```
preg(method=random,effects=both,vrandom=%vrandom,$
      vtime=%na,vindiv=%vindiv) lgaspcar
# constant lincomep lrpmg lcarpcap
```

The other uses time period dummies and `EFFECTS=INDIV`—the `CONSTANT` is dropped. You create the full set of time dummies with `PANEL` with the options `DUMMIES` and `EFFECTS=TIME`.

```
panel(dummies=tdummies,effects=time) %resids
preg(method=random,effect=indiv) lgaspcar
# lincomep lrpmg lcarpcap tdummies
```

With the default of `VCOMP=WK`, those will give identical results—for `EFFECTS=BOTH`, `WK` bases the component estimates off the two-way fixed effect estimates, and the one-way FE combined with time dummies used for `WK` in the second technique matches that exactly. See Appendix A.2 for technical details on variance component calculations. The results for this are:

$$\begin{array}{ccccccc} \log Gas/Car & = & 0.000 & +0.094 \log Y/N & -0.215 \log P_{MG}/P_{GDP} & -0.598 \log CAR/N & \\ & & (0.000) & (0.083) & (0.042) & (0.026) & \end{array} \quad (6.7)$$

This is similar to the fully random two-way regression, but with an even smaller (and quite insignificant) income effect. If the principal interest in doing this analysis is *that* coefficient, rather than the price elasticity, it will not be easy to interpret the results. The incomes are generally growing for all countries through the data set, so differentiating the income effect from a general upward trend is difficult. In the model with just the individual effect, the only variable that can pick up a general trend upwards is the income variable. It's certainly not unreasonable that as living standards improve, cars get bigger and get driven more, so a fairly high income elasticity might be understandable. However, the US and Turkey are on opposite ends of the income scale but

have fairly high values for the dependent variable. When we allow for time effects, the general increase is assigned to that rather than income, leaving the individual effect plus the income variable to try to explain the discrepancy between countries like the US and Turkey.

6.3 Hausman Tests

If you use `METHOD=RANDOM` and `EFFECTS=BOTH`, the included Hausman test is for two-way random effects vs two-way fixed effects. A rejection means that either or both of the individual and time effects seem to be correlated with the regressors. It's also possible to do tests for random vs fixed effects in each direction, conditional on a choice for the other. The only one of these that is likely to be of interest is a test for random individual effects given fixed time effects. That will be produced by the dummy variable method for doing the random individual-fixed time estimators.

6.4 PANEL instruction

We've already seen how **PANEL** with the `EFFECTS=TIME` and `DUMMIES` option can generate time dummies (page 43).

The two-way centering operation (6.4) can be done using **PANEL** with the options, `ENTRY=1.0`, `INDIV=-1.0`, `TIME=-1.0` and `MEAN=1.0`. `TIME` is the weight on the series of time averages and `MEAN` is the weight on the overall sample mean. Note that this only works for a balanced sample—for an unbalanced sample, there is no set of weights on the means that will give the correct result. You can get the identical result (for a balanced sample) with **PANEL** with `EFFECTS=BOTH` and `VRANDOM=1.0`, where the `VINDIV` and `VTIME` variances default to “infinity”—this second method, however, also works for unbalanced data. Similarly, you can do the random effects GLS transformation with `EFFECTS=BOTH` and inputting the desired component variances with the `VRANDOM`, `VINDIV` and `VTIME` options.

Example 6.1 PREGRESS with Two-Way Effects

This estimates a model with two-way fixed and random effects and mixed fixed (time) and random (individual) effects. The detailed description starts on page 42.

```
open data gasoline.xls
cal(panel=19) 1960
data(format=xls,org=cols) 1//1960:1 18//1978:1 year lgaspcar $
  lincomep lrpmpg lcarpcap
*
* Individual (random) effects only
*
preg(method=random,effects=indiv) lgaspcar
# constant lincomep lrpmpg lcarpcap
@regtotex(picture="*.###",dvlabel="$log\,Gas/Car$",$
  rhslabels=||"$log\,Y/N$","$log\,P_{MG}/P_{GDP}$","$log\,CAR/N$"||)
*
* Two-way fixed and random effects
*
preg(method=fixed,effects=both) lgaspcar
# constant lincomep lrpmpg lcarpcap
preg(method=random,effects=both) lgaspcar
# constant lincomep lrpmpg lcarpcap
@regtotex(picture="*.###",dvlabel="$log\,Gas/Car$",$
  rhslabels=||"$log\,Y/N$","$log\,P_{MG}/P_{GDP}$","$log\,CAR/N$"||)
*
* Random individual/fixed time effects using the NA as the VTIME.
*
preg(method=random,effects=both,vrandom=%vrandom,$
  vtime=%na,vindiv=%vindiv) lgaspcar
# constant lincomep lrpmpg lcarpcap
@regtotex(picture="*.###",dvlabel="$log\,Gas/Car$",$
  rhslabels=||"$log\,Y/N$","$log\,P_{MG}/P_{GDP}$","$log\,CAR/N$"||)
*
* Random individual/fixed time effects using dummies
*
panel(dummies=tdummies,effects=time) %resids
preg(method=random,effect=indiv) lgaspcar
# lincomep lrpmpg lcarpcap tdummies
```

Dynamic Panels

7.1 The Bias in Fixed Effects Estimators

Suppose we now want to estimate the following autoregression using panel data:

$$y_{it} = \alpha_i + \rho y_{i,t-1} + \eta_{it} \quad (7.1)$$

There will often be additional explanatory variables, but those aren't essential to the discussion. We're assuming that we have a homogeneous lag coefficient, but differing intercepts. Whether the common dynamics are appropriate might be controversial, but the different intercepts almost certainly aren't—if the series has different means (if $|\rho| < 1$) or different growth rates (if $\rho = 1$), then we need different intercepts to take care of that.

Estimating (7.1) by FE would seem to be fairly straightforward, as long as we de-mean y_{it} and $y_{i,t-1}$ separately. **PREGRESS** takes care of this automatically; if you do the de-meaning with **PANEL**, you will need to copy the lag out to a separate series, and use a **SMPL** option as described on page 23; for instance:

```
set ylag = y{1}
panel(smpl=%period(t)>=2,entry=1.0,indiv=-1.0) y      / py
panel(smpl=%period(t)>=2,entry=1.0,indiv=-1.0) ylag / pylag
```

While fixed effects is only slightly more difficult with dynamic models than with non-dynamic ones, there is a *statistical* problem: by construction, the de-meaned value of $y_{i,t-1}$ is a linear combination of all values of y_{it} from $t = 1$ to $T - 1$, thus including ones from *after* $t - 1$. Similarly, the de-meaned residual is a linear combination of η_{it} from $t = 2$ to T . Because of the dynamics in how y_{it} is built from η_{it} , the residual $\tilde{\eta}_{it}$ will be correlated with the regressor $\tilde{y}_{i,t-1}$.

It's important to note that the same thing would hold if this *weren't* a panel data set, and we simply were estimating an autoregression with a constant term. However, this bias, while substantial for small values of T , declines with $1/T$ and becomes negligible in time series data sets of typical size. In panel data sets, however, we often have a relatively small T and are counting on the N dimension to take care of the asymptotics. However, the bias due to this effect *doesn't* average out as N gets large. Thus the result that FE is biased for dynamic panels with small T . The analytical derivation of the bias was done by Nickell (1981), which (for large N) can be approximated as $-\frac{(1+\rho)}{T-1}$.

Interestingly, while the phrase “biased downwards” in econometrics generally means biased towards zero, in this case, the bias is strictly negative even for $\rho \leq 0$.

Note that this is *not* the source of bias that is usually of concern in analysis of an autoregression. It’s well-known that

$$E\hat{\rho}_{LS} < \rho$$

even for the autoregression without constant $y_t = \rho y_{t-1} + \eta_t$. The source of that is the correlation between the numerator and denominator in:

$$\hat{\rho}_{LS} - \rho = \frac{\sum y_{t-1}\eta_t}{\sum y_{t-1}^2} \quad (7.2)$$

Both numerator and denominator are quadratic forms in the full vector of η . By a two-term Taylor expansion of the ratio A/B about the means of A and B , we get:¹

$$E\left(\frac{A}{B}\right) \approx \frac{E(A)}{E(B)} - \frac{\text{cov}(A, B)}{E(B)^2} + \frac{E(A) \text{var}(B)}{E(B)^3} \quad (7.3)$$

For the autoregression *without* the intercept, $E(A) = 0$, so the first and third terms drop out. The bias is due to the positive correlation between A and B in the middle term. However, when both A and B are sums across i as well as t in panel data, the covariance in the numerator of that middle term goes up with N , while the denominator goes up with N^2 , so that bias goes to zero in the limit as N gets large. For the autoregression *with* an intercept, $E(A)$ is non-zero, and both the numerator and denominator in the first term will go up proportionally with N as that dimension increases; thus the bias doesn’t go away for a fixed value of T .

In practice, the dynamic model takes the more general form:

$$y_{it} = \alpha_i + \rho y_{i,t-1} + X_{it}\beta + \eta_{it} \quad (7.4)$$

where X is assumed to be strictly exogenous (that is, X_{is} is uncorrelated with η_{it} for all t and s) and η_{it} is serially uncorrelated. The same type of bias exists in these more general models, though it will, in general, spill over from ρ to β as well.

There are basically three ways to deal with the bias in fixed effects:

1. Ignore it.
2. Use an instrumental variables procedure rather than least squares.

¹The two-term expansion about (A_0, B_0) is

$$\frac{A}{B} \approx \frac{A_0}{B_0} + \frac{1}{B_0} (A - A_0) - \frac{A_0}{B_0^2} (B - B_0) + \frac{1}{2} 0 (A - A_0)^2 - \frac{1}{2} \frac{2}{B_0^2} (A - A_0) (B - B_0) + \frac{1}{2} \frac{2A_0}{B_0^3} (B - B_0)^2$$

Take expectations with $A_0 = E(A)$ and $B_0 = E(B)$.

3. Compute the bias and correct for it.

If T is large enough to be typical of single time series, ignoring it is certainly the simplest choice and may be the best. If $T = 100$, the bias is at most .02, which will probably be less than the sampling error unless N is also fairly large.

7.2 The Examples

Example 7.1 is from Arellano & Bond (1991). This has an unbalanced data set with 140 firms and up to 9 data points per firm. It's unbalanced, but without gaps—some firms have no data at one or both ends. Gaps in the original data can cause entire firms to drop out of the data set when you start adding lags and differencing operations. In order to get the alignment correct for the lags, we need to transform the original unbalanced data into a `PANEL`-dated data set using `PFORM`. The original data set has quite a few variables that we won't need: `EMP`, `WAGE` and `CAP` are the levels of the logged data series N , W and K that we *do* need and most of the other variables are lags and year dummies that will be created using `RATS` instructions. `ID` and `YEAR` are the firm and year identifiers that are used in re-blocking the data. The equation to be estimated has N (log employment) as the dependent variable, with lags of N , current and lagged W (log real wage), K (log gross capital) and YS (expected demand). A full set of (fixed effect) time dummies are also included.

The following reads and re-organizes the data:

```
open data abdata.dta
data(format=dta) 1 1031 ind year emp wage cap indoutpt n w k ys $
  rec yearm1 id n11 n12 w11 k11 k12 ys11 ys12 yr1976 yr1977 yr1978 $
  yr1979 yr1980 yr1981 yr1982 yr1983 yr1984
pform(indiv=id,time=year) p_n
# n
pform(indiv=id,time=year) p_w
# w
pform(indiv=id,time=year) p_k
# k
pform(indiv=id,time=year) p_ys
# ys
*
cal(panel=%nobs)
all %ngroup//%panelobs()
```

The rearranged data series that we will use are now `P_N`, `P_W`, `P_K` and `P_YS`. Least squares and FE estimators (with clustered standard errors) can be done using:

```

preg(cluster=%indiv(t),effect=time) p_n
# p_n{1 2} p_w{0 1} p_k{0 1 2} p_ys{0 1 2}
preg(cluster=%indiv(t),effect=both) p_n
# p_n{1 2} p_w{0 1} p_k{0 1 2} p_ys{0 1 2}

```

The use of **PREGRESS** allows for easier handling of the fixed time effects—we just add time effects to whatever is the desired handling of the individual effects. These are for comparison only—the first of these is likely to be badly misspecified by ignoring the individual effects, while the latter should have fairly significant bias with such a small value of T .

Example 7.2 is the example used in Kiviet (1995) to demonstrate the use of bias-correction. This is a balanced sample of 94 countries, with 6 time periods for each, in five year intervals from 1960 to 1985. The two series on the data set are:

1. $Y = \log(\text{real per capita GDP (final year in covered period)})$
2. $X = \log(\text{savings rate}) - \log(\text{population growth rate})$ (5 year averages)

The model to be estimated is

$$y_t = \rho y_{t-1} + \beta x_t + \mu_i + \eta_{it}$$

The data file is an unlabeled text file, which is read with:

```

cal(panelobs=6,ypp=5) 1960:1
all 94//6
open data penngrow.txt
data(org=columns) / indiv year y x

```

The **YPP** option on **CALENDAR** is for Years Per Period, which is for this type of situation, where the data are measured every five years.

7.3 First Difference Instrumental Variables Estimators

Instead of extracting individual means to remove the fixed effect, an alternative is to first difference the data, producing:

$$y_{it} - y_{i,t-1} = \rho(y_{i,t-1} - y_{i,t-2}) + (X_{it} - X_{i,t-1})\beta + (\eta_{it} - \eta_{i,t-1}) \quad (7.5)$$

We still can't employ least squares to estimate this because the transformed error $(\eta_{it} - \eta_{i,t-1})$ and transformed regressor $(y_{i,t-1} - y_{i,t-2})$ are correlated through $\eta_{i,t-1}$. However, Anderson & Hsiao (1982) note that linear combinations of $y_{i,s}$ for $s < t-1$ are uncorrelated with $(\eta_{it} - \eta_{i,t-1})$. In particular, $y_{i,t-2}$ or $y_{i,t-2} - y_{i,t-3}$ can be used as instruments for $(y_{i,t-1} - y_{i,t-2})$.² By the exogeneity assumption

²An equivalent alternative that is perhaps a bit easier to understand is differencing *forwards*, rather than backwards:

$$y_{it} - y_{i,t+1} = \rho(y_{i,t-1} - y_{i,t}) + (X_{it} - X_{i,t+1})\beta + (\eta_{it} - \eta_{i,t+1})$$

on X , $(X_{it} - X_{i,t-1})$ is a perfectly good instrument for itself. If we had merely assumed that the X were pre-determined³, this wouldn't be the case. Because it seems to be clearly the superior choice of the two instruments, the Anderson-Hsiao (or AH) estimator is this instrumental variables estimator with $y_{i,t-2}$ as the instrument.⁴ This is consistent, but often *very* inefficient. To see why, recall that to be a valid instrument, a variable must satisfy *two* criteria:

- It must be uncorrelated with the residual.
- It must be *correlated* with the variable it instruments.

The first is true by assumption. The second is also true if $\rho < 1$, but the correlation becomes weaker as $\rho \rightarrow 1$.

The efficiency of this *might* be improved by adding longer lags of y to the instrument set, but each additional lag costs one data point per individual. We're already losing two data points to lags in the basic AH estimator, so if T isn't large, we may not have many to spare.

To implement AH with the data from Example 7.1, we need to create time period dummies and the differenced series. Since there are no gaps in the data, we don't need to create differenced dummies—the original dummies and the differenced dummies span the same space, and since they are nuisance parameters, we don't really care which one we use.

```
set dn = p_n-p_n{1}
set dw = p_w-p_w{1}
set dys = p_ys-p_ys{1}
set dk = p_k-p_k{1}
panel(effects=time,dummies=tdummies) constant
```

The estimates are done with the following:

```
instruments dn{3} dn{2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
linreg(inst,title="Anderson-Hsiao-D",lwindow=panel) dn
# dn{1 2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
instruments p_n{3} dn{2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
linreg(inst,title="Anderson-Hsiao-L",lwindow=panel) dn
# dn{1 2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
```

Note that the second lag of DN doesn't need instrumenting; under the basic assumptions, only the first lag has a problem with correlation with the differenced residuals. However, the standard AH second lag can't be used to instrument both itself and the first lag as well, so the *third* lag (of either the difference or level) needs to be added to the instrument set instead.

With that, the lags beginning with the more natural $t - 1$ are available as instruments. This is similar to the filter forward-instrument back estimator of Hayashi & Sims (1983) in a pure time series context.

³That is, X_{is} is only assumed uncorrelated with η_{it} for $s \leq t$

⁴If both forms are considered, they are generally called AH-L and AH-D for AH with Lag and AH with Difference.

7.4 Expanded Instrument Sets

The Anderson-Hsiao estimator uses only a single instrument for the lagged difference. As pointed out in both Holtz-Eakin et al. (1988) and Arellano & Bond (1991), with a panel data set, there is a substantially larger set of instruments available. For each t and each $s < t - 1$, by assumption $E y_{i,s} (\eta_{i,t} - \eta_{i,t-1}) = 0$. With a single time series, this would also be true, but would be of little use, since there would be just one data point for each s, t combination. However, with panel data, the sample analogue of this would be a sum over N individuals and thus the results from Hansen (1982) can be applied with asymptotics as $N \rightarrow \infty$.

This generates a potentially very large number of instruments. Assuming no other losses to lags, if there are originally T data points per cross section, you have $(T - 1)(T - 2)/2$ for a full set of what we will call *Arellano-Bond instruments* or *A-B instruments*. Unless T is rather small, a full set may be infeasibly large. With $N(T - 2)$ usable data points, if $(T - 1)/2 > N$ there are more A-B instruments than there are data points, so there is no way to compute a full rank cross product of the instruments. Even worse, the second-step estimator (page 53) generates a rank N matrix using the calculation in (3.3), which will be singular if the total number of instruments merely exceeds N .⁵ Thus, the Arellano-Bond estimator, without adjustments to the instrument set, only works in big N -small T situations.

The ABLAGS Procedure

The A-B instruments can be created using the procedure `@ABLAGS`. The basic use of this is

```
@ablags wks abwks
```

which generates a VECTOR[SERIES] named ABWKS which has a full set of A-B instruments for the series WKS, that is, a separate one for each $t \geq 3$ and for each lag of WKS from 2 on to the maximum possible given T . There are several options which can be used to restrict the size of the instrument set. The MAXLAG option restricts the set of lags to the range from 2 to the values in the MAXLAG option. For instance:

```
@ablags(maxlag=2) fwage abfwage
```

will generate an instrument set with lag 2 (only) of FWAGE separately for each time period from 3 to T . One would expect that the instruments get weaker as the lag increases so using a MAXLAG option to cut out distant lags would reduce the instrument set size without costing much in the way of ability to fit the equation. This gives roughly $(T - 2)L$ instruments, where L is the value of MAXLAG minus one.⁶ If something more extreme is needed, the COLLAPSE

⁵And this requires counting *all* instruments, not just the A-B ones

⁶There are a few less than this due to the lack of data for lags when t is small.

option eliminates the replication of instruments for different time periods. Instead, there is a single instrument for each possible lag value, reducing the number of instruments to $T - 2$.

Arellano & Bond (1991) propose two estimators based upon a set of instruments—a one-step and a two-step estimator. The first is based upon the assumption that η_{it} is serially uncorrelated with a constant variance. If that's true, then the error process in (7.5) is a first order moving average process with a -1 lag coefficient, and (along with the other assumptions), this falls into case (iv) in Hansen (1982). Other than scaling by the (unknown) constant variance (which doesn't affect the point estimates), the optimal (inverse) weight matrix can be computed using RATS with

```
mcov(instruments,lwform=||-1.0,2.0,-1.0||,smp1=absample)
```

The covariogram of the first difference error process is:

$$\begin{aligned} E\tilde{\eta}_{i,t}^2 &= 2\sigma_\eta^2 \\ E\tilde{\eta}_{i,t}\tilde{\eta}_{i,t-1} &= -\sigma_\eta^2 \\ E\tilde{\eta}_{i,t}\tilde{\eta}_{i,t-k} &= 0 \text{ if } k > 1 \end{aligned}$$

which is the source for the `LWFORM` weights on the instruments: in the `VECTOR` of values passed to `LWFORM`, they give the covariances (up to scale by σ_η^2) at the -1, 0 and 1 lags of the error process in order. Note that this is *not* one of the standard windows (like Newey-West) used in computing robust standard errors. This use of `MCOV` comes up only in very limited situations where the properties of the error process are known *exactly*.

The `SMPL=ABSAMPLE` option is required because the A-B instruments don't restrict the sample range on the `MCOV` to the range that will eventually be used in the estimation. They are instrumenting out the first difference of the lag of y , which costs two data points per cross section, but are defined over the full range from 1 to T (padded with zeros at the front end). The following generates the proper sample series for the regression; it inputs the dependent and explanatory variables. (The dummies don't restrict the range at all, and so can be left out).

```
inquire(reglist,valid=absample)
# dn{0 1 2} dw{0 1} dk{0 1 2} dys{0 1 2}
```

The first step estimator uses the inverse of the `%CMOM` matrix generated by `MCOV` as the weight matrix. Because the input covariogram doesn't include the factor of σ_η^2 , the standard errors will be way off unless something is done to correct them. The standard handling is to use clustered standard errors. Hence, the instruction:

```
linreg(inst,wmatrix=inv(%cmom),robusterrors,lwindow=panel,$
title="Arellano-Bond One-Step Estimator") dn
# dn{1 2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
```

You don't need to repeat the `SMPL` option here because the estimated equation itself uses the lags of `DN` and thus `LINREG` restricts the sample properly.

The second step is to take the residuals from the first step and do the "optimal" weight matrix using those with the clustered estimates. This is where you can run into a serious problem with too many instruments if T is quite a bit larger than it is here. By construction, the clustered `MCOV` matrix is *at most* rank N so this will be singular if the instrument list is bigger than that.

```
mcov(instruments,lwindow=panel) / %resids
linreg(inst,wmatrix=inv(%cmom), $
      title="Arellano-Bond Two-Step Estimator") dn
# dn{1 2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
```

Endogenous and Pre-Determined Explanatory Variables

Our original assumptions were that all other explanatory variables were strictly exogenous, which allows them to be used as their own instruments, even after first differencing. In many cases, strict exogeneity is too strong an assumption. When a variable is determined simultaneously with the dependent variable, then it will have exactly the same correlation problem as the dependent variable itself. It will also (presumably) satisfy exactly the same type of orthogonality restrictions with its lags. Thus, you can instrument it out with exactly the same type of A-B instruments as you apply to the dependent variable.

If, instead, an explanatory variable is pre-determined, x_{it} (and its lags) will be uncorrelated with η_{it} but not necessarily with lags of η . In first differenced form, we have $x_{it} - x_{i,t-1}$ and $\eta_{it} - \eta_{i,t-1}$. These are potentially correlated through x_{it} with $\eta_{i,t-1}$. Since $x_{i,t-1}$ is, by assumption, uncorrelated with both η_{it} and $\eta_{i,t-1}$, we can use A-B style instruments, but can start with lag 1, rather than lag 2. You can do this using the option `MINLAG=1` on `@ABLAGS`. The default for `MINLAG` is 2, as that's the most common use.

7.5 Bias Correction

This is mainly due to the work of Jan Kiviet, beginning with Kiviet (1995). Although, theoretically, the ideas could be extended to higher-order autoregressive processes, or VAR's, in practice, it's only been applied to (7.4).

Kiviet's formulas can be very complicated. They make use of a matrix generalization of (7.3). There are three levels of correction, depending upon how many of those analogous terms are used.

The model to be estimated is written

$$y_{it} = W_{it}\beta + \mu_i + \eta_{it}$$

where (for convenience) the first element of W is $y_{i,t-1}$ and the rest of the explanatory variables are assumed to be strictly exogenous. As above, we'll call

the coefficient on the lagged dependent ρ .⁷ The within deviations matrix \mathbf{Q} will eliminate the individual effects, so, in matrix notation, we get the result:

$$\hat{\beta}_{LSDV} - \beta = (\mathbf{W}'\mathbf{Q}\mathbf{W})^{-1}\mathbf{W}'\mathbf{Q}\eta$$

If we look closely at $\mathbf{W}'\mathbf{Q}\eta$, we see that the only component of that which will have a non-zero expected value is the first, since all the other explanatory variables are strictly exogenous. With η_{it} assumed to be uncorrelated both across i and t , there also won't be any correlation between blocks corresponding to different i . Although it's possible to work with an unbalanced data set, it's much easier to restrict the analysis to a balanced sample. For a given i , we need to compute the expected value of:

$$(y_{i,0}, \dots, y_{i,T-1})\mathbf{Q}_{T-1}(\eta_{i,1}, \dots, \eta_{i,T})'$$

which is most easily done by transforming it into

$$\text{trace } \mathbf{Q}_{T-1}E(\eta_{i,1}, \dots, \eta_{i,T})(y_{i,0}, \dots, y_{i,T-1})$$

Under the assumptions, the expected value matrix takes the form:

$$\sigma_\eta^2 \begin{bmatrix} 0 & 1 & \rho & \dots & \rho^{T-2} \\ 0 & 0 & 1 & \dots & \rho^{T-3} \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (7.6)$$

Let's call this $\sigma_\eta^2 \mathbf{C}_{T-1}(\rho)$. Note that the correction depends upon the unobservable σ_η^2 and ρ , so we will need some initial guess values to make it feasible. Then

$$E(\mathbf{W}'\mathbf{Q}\eta) = e_1 N \sigma_\eta^2 \text{trace}(\mathbf{Q}_{T-1} \mathbf{C}_{T-1}(\rho)) = -e_1 \sigma_\eta^2 \frac{N}{T-1} \iota'_{T-1} \mathbf{C}_{T-1}(\rho) \iota_{T-1}$$

where e_1 is a unit vector with 1 only in the first slot. The trace calculation simplifies quite a bit because \mathbf{C} has a zero diagonal. $\iota'_{T-1} \mathbf{C}_{T-1}(\rho) \iota_{T-1}$ is just the sum of the elements of $\mathbf{C}_{T-1}(\rho)$, which will always be positive for any value of ρ in the unit interval, thus the overall coefficient is negative. This is one piece of the full calculation—the others are similar but more complicated.

An LSDVC (C for “corrected”) estimator is implemented by:

1. Computing an initial consistent estimator for σ_η^2 and ρ . AH is often used, although an Arellano-Bond estimator, perhaps with a reduced set of instruments, might be a better choice due to the poor small-sample behavior of AH.
2. Computing the fixed effects estimator.

⁷Kiviet uses γ instead.

3. Computing an approximation to the bias using the initial consistent estimators and subtracting it from the fixed effects estimator.
4. Correcting the covariance matrix of the estimators since the correction depends upon sample information.

Implementing in RATS

The Kiviet estimators can be done using the procedure `@LSDVC`. The basic syntax is something like:

```
@lsdvc(method=k3) y
# x
```

You can have more than one X variable, but don't include the lagged dependent on the list of explanatory variables—it assumes that you are using exactly one lag of Y . The `METHOD` option chooses the type of estimator: Kiviet's corrected versions are chosen using `METHOD=K1`, `METHOD=K2` or `METHOD=K3`, depending upon the number of correction terms. You can also choose `METHOD=AH`, which is the AH-L estimator (page 49), `METHOD=FIXED`, which is the uncorrected fixed effects estimator and `METHOD=SIMPLE`, which might be called $K\ 1/2$. It does a one term correction using the sample value of $(W'QW)^{-1}$ rather than its expected value. There's almost no difference in performance among any of the corrected formulas—they're all consistent and give very similar results in practice.

The following from Example 7.2 uses the `@LSDVC` procedure to do three different estimators for the model: AH, the full 3-term correction, and the simple one-term correction:

```
@lsdvc(method=ah) y
# x
@lsdvc(method=k3) y
# x
@lsdvc(method=simple,robust) y
# x
```

The big problem with the LSDVC estimator is that it doesn't generalize easily. Example 7.1 uses two lags of the dependent variable rather than one. That means that the top *two* elements of $E(W'Q\eta)$ are non-zero, the moving average matrix (the equivalent of $C_{T-1}(\rho)$) will be more complicated since it's for an AR(2) process rather than AR(1) and will be not be the same when evaluating the bias in the first lag as it is for the second lag. While clearly more complicated, that is still doable. However, there is no simple way to extend it to allow predetermined X variables—strict exogeneity is absolutely required.

Example 7.1 Dynamic Panel-Instrumental Variables Estimation

This does most of the analysis of dynamic panels from Arellano & Bond (1991). The data set is described on page 48 and the detailed discussion on 50.

```

open data abdata.dta
data(format=dta) 1 1031 ind year emp wage cap indoutpt n w k ys $
  rec yearml id nl1 nl2 wl1 kl1 kl2 ysl1 ysl2 yr1976 yr1977 yr1978 $
  yr1979 yr1980 yr1981 yr1982 yr1983 yr1984
*
* n, w, k and ys are the logged values of emp, wage, cap and indoutpt
*
pform(indiv=id,time=year) p_n
# n
pform(indiv=id,time=year) p_w
# w
pform(indiv=id,time=year) p_k
# k
pform(indiv=id,time=year) p_ys
# ys
pform(indiv=id,time=year) p_year
# year
*
cal(panel=%nobs)
all %ngroup//%panelobs()
*
* OLS with panel-robust standard errors. Done with PREGRESS,
* EFFECTS=TIME (to handle the time dummies) with the CLUSTER option.
*
preg(cluster=%indiv(t),effect=time) p_n
# p_n{1 2} p_w{0 1} p_k{0 1 2} p_ys{0 1 2}
*
* Within-groups estimator, again with clustered standard errors. Since
* this has both time and individual effects, it uses EFFECTS=BOTH.
*
preg(cluster=%indiv(t),effect=both) p_n
# p_n{1 2} p_w{0 1} p_k{0 1 2} p_ys{0 1 2}
*
set dn = p_n-p_n{1}
set dw = p_w-p_w{1}
set dys = p_ys-p_ys{1}
set dk = p_k-p_k{1}
*
* Anderson-Hsiao with differences. The second lag doesn't need
* instrumenting.
*
instruments dn{3} dn{2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
linreg(inst,title="Anderson-Hsiao-D",lwindow=panel) dn
# dn{1 2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
*
* Anderson-Hsiao with lag
*

```

```

instruments p_n{3} dn{2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
linreg(inst,title="Anderson-Hsiao-L",lwindow=panel) dn
# dn{1 2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
*
* Arellano-Bond Estimators
* This uses a full set of A-B lags. (N=140 >> T=9)
*
panel(effects=time,dummies=tdummies) constant
@ablags p_n ablogn
instruments ablogn dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
*
* One-step estimator
*
* It's necessary to determine the final estimation sample for the first
* MCOV since the A-B lags won't restrict the range for that the way the
* lagged n's will in the regression or the residuals will in the second
* step MCOV.
*
inquire(reglist,valid=absample)
# dn{0 1 2} dw{0 1} dk{0 1 2} dys{0 1 2}
mcov(instruments,lwform=||-1.0,2.0,-1.0||,smpl=absample)
linreg(inst,wmatrix=inv(%cmom),robusterrors,lwindow=panel,$
    title="Arellano-Bond One-Step Estimator") dn
# dn{1 2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies
*
* Two-step estimator. This has to follow the one-step estimator since it
* uses the residuals from that.
*
mcov(instruments,lwindow=panel) / %resids
linreg(inst,wmatrix=inv(%cmom),$
    title="Arellano-Bond Two-Step Estimator") dn
# dn{1 2} dw{0 1} dk{0 1 2} dys{0 1 2} tdummies

```

Example 7.2 Dynamic Panel-Biased Correction Estimation

This demonstrates the use of LSDV corrected estimation for dynamic panels. The data set is described on page 49 and the discussion begins on page 55.

```
compute n=94
cal(panelobs=6, ypp=5) 1960:1
all 94//6
open data penngrow.txt
*
* y = log(real per capita GDP)
* x = log(savings) - log(population growth)   (5 year averages)
*
data(org=columns) / indiv year y x
*
* Least squares; doesn't allow for individual effects.
*
linreg y
# constant y{1} x
*
* Fixed effects. Severe bias with so few data points
*
preg(method=fixed) y
# y{1} x
*
* Anderson-Hsiao. Consistent, but has very low efficiency.
*
@lsdvc(method=ah) y
# x
*
* LSDV with Kiviet 3-term correction
*
@lsdvc(method=k3) y
# x
*
* LSDV with simpler correction
*
@lsdvc(method=simple, robust) y
# x
```


Non-Linear Models

So far, we have dealt only with models which are linear in both the parameters and variables. Non-linear models of almost any form are quite a bit more difficult to handle if we allow for individual effects. In most cases, the analogue of a fixed effects estimator won't give consistent estimates even for homogeneous parameters unless the time dimension is large, due to the *incidental parameters problem*: the likelihood can't be arranged to separate the poorly estimated individual effects from the homogeneous parameters. And random effects will often require time-consuming numerical integration. Among the types of models for which some type of fixed effects can be defined and give consistent estimates are non-linear least squares, logit and Poisson models; among those for which it can't are probit and tobit.

8.1 Non-linear Least Squares

We assume that we have a non-linear model with an additive individual effect:

$$y_{it} = \alpha_i + f(X_{it}, \beta) + u_{it}$$

Assuming homoscedastic residuals, the log likelihood (ignoring constants) is

$$-\frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum (y_{it} - \alpha_i - f(X_{it}, \beta))^2$$

The residual can be rewritten as:

$$\{y_{it} - f(X_{it}, \beta) - (y_{i\bullet} - f(X_{it}, \beta)_{i\bullet})\} - \{\alpha_i - (y_{i\bullet} - f(X_{it}, \beta)_{i\bullet})\}$$

where we're using $f(X_{it}, \beta)_{i\bullet}$ to mean the average of $f(X_{it}, \beta)$ for individual i . When this is squared and summed, the cross terms between the two expressions in braces will zero out when summed over the data for i since the second term is a constant for a given i and the first sums to zero for i . Thus, the log likelihood can be written as:

$$\begin{aligned} -\frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum \{y_{it} - f(X_{it}, \beta) - (y_{i\bullet} - f(X_{it}, \beta)_{i\bullet})\}^2 \\ - \frac{1}{2\sigma^2} \sum \{\alpha_i - (y_{i\bullet} - f(X_{it}, \beta)_{i\bullet})\}^2 \end{aligned} \quad (8.1)$$

The second sum is the only place where the individual effects α_i occur, and given β , the maximizing value is simply

$$\hat{\alpha}_i = y_{i\bullet} - f(X_{it}, \beta)_{i\bullet}$$

which zeros out the last term. So the maximizing value of β comes from minimizing the first sum in (8.1) which doesn't depend upon α_i . This is how the incidental parameters problem is avoided—the final term doesn't restrict β in any way so the α_i have been isolated from β . To estimate the parameters, we need to minimize:

$$\sum \{y_{it} - f(X_{it}, \beta) - (y_{i\bullet} - f(X_{it}, \beta)_{i\bullet})\}^2 \quad (8.2)$$

which requires, for a function evaluation, extracting the individual means from $f(X_{it}, \beta)$. If the function is linear in the variables, non-linear in parameters: $f(X_{it}, \beta) = X_{it}g(\beta)$, the residual can be simplified to:

$$y_{it} - y_{i\bullet} - (X_{it} - X_{i\bullet})g(\beta)$$

which can be handled by extracting individual means from y and X (using **PANEL**) and writing the non-linear formula using the de-meaned data. However, it's very rare to have a model like that. More common is a model with some linear components and some non-linear. For instance, a threshold model will have the general form:

$$f(X_{it}, \beta) = X_{it}\beta_0 + g(X_{it}, \gamma)X_{it}\beta_1$$

Given γ , this is linear in X_{it} and $g(X_{it}, \gamma)X_{it}$. A function evaluation (for a setting of γ) could be done by a fixed effects regression of y on X_{it} and $g(X_{it}, \gamma)X_{it}$. You could also get the de-meaned copies of y and X first, then generate the series $g(X_{it}, \gamma)X_{it}$ for a test value of γ , de-mean it and run a linear regression to get the regression parameters β_0 and β_1 . This is computationally more efficient than a brute force calculation of (8.2), but will take more work to set up. Whether it's worth the effort will depend upon the size of the data set and the number of parameters (particularly the number of parameters in the linear part β_0).

Example 8.1 is based upon one of the models estimated in Baltagi (2008) (his Example 2.6.3). The data set has annual data from 1970 to 1986 on 48 states. The basic model is a Cobb-Douglas production function for total state production using public capital, private capital and labor as inputs. The model estimated, however, adds the unemployment rate as an extra explanatory variable “to capture business cycle effects.” Suppressing the it subscripts:

$$\log Y = \alpha + \beta_1 \log K_1 + \beta_2 \log K_2 + \beta_3 \log L + \beta_4 U + u$$

where K_1 is public and K_2 private capital. The data are read and transformed to logs using:

```

cal(panelobs=17) 1970
open data produc.xls
data(format=xls,org=cols) 1//1970:1 48//1986:1 year p_cap $
    hwy water util pc gsp emp unemp
*
set logy = log(gsp)
set logk1 = log(p_cap)
set logk2 = log(pc)
set logl = log(emp)

```

Allowing for fixed individual effects, the basic model is estimated with:

```

preg(method=fixed) logy
# logk1 logk2 logl unemp

```

Instead of simply adding the unemployment rate to the end of a Cobb-Douglas function, suppose that we assume that unemployment affects the productivity of measured private capital with a logistic smooth transition. Instead of the linear effect $\beta_2 \log K_2$, the capital term is the non-linear

$$\beta_{20} \log K_2 + \beta_{21} [1 + \exp(-\gamma(U - c))]^{-1} \log K_2$$

For values of U substantially higher than c , the logistic will have a value near 1, so the coefficient on $\log K_2$ will be $\beta_{20} + \beta_{21}$. For values substantially lower than c , the logistic will have a value near zero, so the coefficient will be β_{20} .

We'll show two ways to estimate this, one through the direct use of (8.2), that is, by first computing then de-meaning the non-linear explanatory function, the other by using linear regression on transformed data, searching only over the inherently non-linear γ and c parameters.

For the first, we set up the non-linear parameter set and initialize the values. The values of the scaling parameter γ are highly dependent upon the scale of the threshold series, so it's recommended that you divide the logistic parameter by a measure of scale (here the standard deviation of the threshold series), which will generally make the new γ within a few orders of magnitude of 1.

```

nonlin b1 b20 b21 b3 gamma c
compute b1=%beta(2),b20=%beta(3),b21=0.0,b3=%beta(4)
stats unemp
compute c=%mean,scalef=1.0/sqrt(%variance),gamma=1.0
*
frml tfrml = b1*logk1+logk2*$
    (b20+b21*%logistic((unemp-c)*scalef*gamma,1.0))+b3*logl

```

You can't apply **NLLS** directly to **TFRML**, since we need the de-meaned version of that. Instead, you need to use a **START** option to do the calculation, then the **FRML** on the **NLLS** just takes that series as the value, though actually, we'll have that function return the demeaned non-linear part plus the individual

means of $\log Y$ so the dependent variable will still be $\log Y$.¹ This will be pretty much the same for any model handled by direct estimation of (8.2)—you just have to change the definition of `TFRML`.

```
dec series tdemean
panel(indiv=1.0) logy / ymean
frml twork = tdemean+ymean
function Demean
set tdemean = tfrml(t)
panel(entry=1.0,indiv=-1.0) tdemean
end Demean
```

Threshold models often take a couple of passes to estimate properly. This first holds γ and c at their guess values to move `B21` off the zero value. If you start with $\beta_{21} = 0$ and try to estimate the full set of parameters, γ and c don't enter the function value at all at the initial values.

```
nonlin b1 b20 b21 b3
nlls(start=Demean(),frml=twork) logy
```

The actual model is estimated with:

```
nonlin b1 b20 b21 b3 gamma c
nlls(start=Demean(),frml=twork) logy
```

If we compare the results from the original fixed effects regression:

Sum of Squared Residuals	1.1111885088			
1. LOGK1	-0.026149654	0.029001575	-0.90166	0.36751996
2. LOGK2	0.292006925	0.025119673	11.62463	0.00000000
3. LOGL	0.768159473	0.030091739	25.52725	0.00000000
4. UNEMP	-0.005297741	0.000988726	-5.35815	0.00000011

with the output from the threshold model:

Sum of Squared Residuals	1.1107994523			
1. B1	-0.028505941	0.028647307	-0.99507	0.32000168
2. B20	0.296914536	0.028961314	10.25211	0.00000000
3. B21	-0.011803681	0.023320222	-0.50616	0.61288454
4. B3	0.769531647	0.029434453	26.14391	0.00000000
5. GAMMA	0.386429107	0.816659167	0.47318	0.63621022
6. C	8.181783980	5.785750037	1.41413	0.15770892

the threshold model has the expected sign (though insignificant) on β_{21} and fits slightly better, but probably is not worth the extra two parameters.

The alternative approach is to extract means from the series which enter linearly (which will be all but one), then, for each test value of γ and c , generate and de-mean the threshold times capital series and run a linear regression. Search for the minimizing sum of squared residuals.

¹This affects only the R^2 measures in the output, not the sum of squared residuals, log likelihood, etc.

```

panel(entry=1.0,indiv=-1.0) logy / cy
panel(entry=1.0,indiv=-1.0) logk1 / ck1
panel(entry=1.0,indiv=-1.0) logk2 / ck2
panel(entry=1.0,indiv=-1.0) logl / cl

```

We'll use the same guess values as before. Since we're not treating the β parameters using **NLLS**, we don't need guess values for them.

```

stats unemp
compute c=%mean,scalef=1.0/sqrt(%variance),gamma=1.0

```

Because we're minimizing the sum of squared residuals out of a **LINREG**, the instruction to use is **FIND**.

```

nonlin gamma c
*
find min %rss
    set logk2x = logk2*%logistic((unemp-c)*scalef*gamma,1.0)
    panel(entry=1.0,indiv=-1.0) logk2x / ck2x
    linreg(noprint) cy
    # ck1 ck2 ck2x cl
end do find

```

Because this isn't doing a full Gauss-Newton estimation, the output from **FIND** is quite a bit less informative:

FIND Optimization - Estimation by Simplex	
Function Value	1.1108
Variable	Coeff

1. GAMMA	0.3864304641
2. C	8.1818390924

The regression coefficients can be obtained with:

```

linreg(print) cy
# ck1 ck2 ck2x cl

```

1. CK1	-0.028505988	0.027940139	-1.02025	0.30791274
2. CK2	0.296914467	0.024742195	12.00033	0.00000000
3. CK2X	-0.011803643	0.002126571	-5.55055	0.00000004
4. CL	0.769531680	0.029005812	26.53026	0.00000000

Note that this gives a *very* misleading *t*-statistic on the threshold variable since it takes γ and c as fixed, when they are, in fact, not particularly well-determined. Because this will always be a problem with this second method, we would generally recommend using the first unless it is just too slow. If, because of speed issues, you need to use the **FIND** method, you can do 1 iteration with the **NLLS** method to get the same standard errors that you would get if you used **NLLS** for the entire estimation.

8.2 Probit and Logit Models

These are best understood by using an index model formulation:

$$V_{it} = \alpha_i + X_{it}\beta + u_{it}$$

which gives the (unobservable) utility V of choosing $y_{it}=1$. If this is positive, $y_{it} = 1$, and if negative, $y_{it} = 0$. The probit uses a standard $N(0, 1)$ for the u , while the logit uses a logistic.

One problem with applying fixed effects to logit and probit models is immediately apparent. If individual i chooses all 1's or all 0's, the data are predicted perfectly with $\alpha_i = +\infty$ if they are all 1's and $\alpha_i = -\infty$ if they are all 0's. In either case, there is no usable information about β . For inference on β , we will have to restrict ourselves to individuals who have a mix of responses.

It turns out that there is no way to isolate the α_i in the probit—the Normal CDF doesn't decompose in any nice way. However, the logit can be transformed into a slightly altered conditional likelihood, which, while not the same as the true likelihood, ends up providing a way to estimate the β . After describing the example that will be used, we will discuss first the conditional (or fixed effects) logit and then random effects probit.

The Example

Example 8.2 is from Wooldridge (2010). This estimates a model of labor force participation (1 for in, 0 for out) for a sample of 5663 married women, with five data points for each. The explanatory variables are:

- KIDS (number of children under 18)
- LHINC (log of husband's income in dollars per month)
- EDUC (years of schooling at the start of the sample)
- BLACK (race indicator)
- AGE (age at start of sample)
- AGESQ (age squared at start of sample)

As they are defined here, the last four are time-invariant. The data are read with

```
open data lfp.dta
calendar(panelobs=5)
data(format=dta) 1//1 5663//5 id period lfp black educ age agesq $
  kids hinc per1 per2 per3 per4 per5 lhinc
```

This also includes time period dummies (PER1 to PER5) on the data set. We'll use those rather than creating them, which you could do with **PANEL** with the **EFFECTS=TIME** and **DUMMIES** option (see page 43). A fixed effects linear probability model can be estimated with:

```
preg(method=fixed) lfp
# kids lhinc per1 per2 per3 per4 per5
```

which leaves out the time-invariant regressors (PER5 is actually redundant, but that has no effect on the estimates). The pooled probit model (which can use all the variables) is done with

```
ddv(dist=probit) lfp
# kids lhinc constant per1 per2 per3 per4 educ black age agesq
```

The two estimators disagree rather sharply on the effect of the husband's income regressor. Both have it negative, which would be the expected sign, but the pooled probit gives it a t statistic of 16, while fixed effects has a more modest 2.3. Both, however, are flawed estimators—fixed effects because it's not really a probability model, the pooled probit because it doesn't take into account individual effects. The conditional logit and random effects probit estimators are designed to correct those flaws.

8.2.1 Conditional (Fixed Effects) Logit

The logit probability of a 1 at it is

$$\frac{\exp(\alpha_i + X_{it}\beta)}{1 + \exp(\alpha_i + X_{it}\beta)}$$

and the probability of a 0 is

$$\frac{1}{1 + \exp(\alpha_i + X_{it}\beta)}$$

In log form, this can be combined to:

$$y_{it}(\alpha_i + X_{it}\beta) - \log(1 + \exp(\alpha_i + X_{it}\beta))$$

If we sum this over t for individual i and use the shorthand Y_i for the count of 1's in y_{it} , we get:

$$\alpha_i Y_i + \sum_t y_{it} X_{it} \beta - \sum_t \log(1 + \exp(\alpha_i + X_{it} \beta))$$

This is close to what we need, except for that final term which stubbornly refuses simplification. However, the final term is common to *any* calculation of the probabilities for individual i . If we had any other combination of y_{it} adding up to Y_i , the first term would be the same, the last term would be the same; the only difference would be the center term. The difference between the two log probabilities (thus the log ratio of probabilities) would involve only β . While certainly true, what would this accomplish? Of all possible ways to come up with Y_i 1's, why did we observe the ones that we did? We would hope it's because that center term is relatively large compared to what it would be for the other possibilities, and thus provides information about β . This is the idea

behind the *conditional logit estimator*. We decompose the likelihood (for individual i) into

$$P(\{y_{i1}, \dots, y_{iT}\}) = P(\{y_{i1}, \dots, y_{iT}\} | Y_i) P(Y_i) \quad (8.3)$$

The first factor requires computing

$$P(\{y_{i1}, \dots, y_{iT}\}) / \sum_c P(\{y_{c1}, \dots, y_{cT}\})$$

where the c 's form an exhaustive list of the combinations of settings for y with Y_i 1's. Because of the cancellations described above, the log of this simplifies to

$$\sum_t y_{it} X_{it} \beta - \log \left(\sum_c \sum_t \exp(y_{ct} X_{it} \beta) \right)$$

Summed across i , this gives the conditional log likelihood.

Note that this is not the same thing as the log likelihood itself. That is the conditional log likelihood plus (from (8.3))

$$\sum_i \log(P(Y_i)) \quad (8.4)$$

While this can be fairly easily maximized over α_i ², unlike the situation with non-linear least squares, the maximized probability still depends upon β . The actual log likelihood for the logit still suffers from incidental parameters—it's just that we can get consistent estimates of β by using the conditional part only.

Note that for a data set with a fairly large T dimension (say 20), summing across the number of combinations could be quite a sizeable calculation as this has to be computed for each function evaluation. For $T = 20$ and $Y_i = 5$, there are over 15,000 combinations and if $Y_i = 10$, there are nearly 200,000. If N is relatively modest for a data set like that, direct estimation using the log likelihood with individual dummies for the fixed effects will be quite a bit faster, and T is large enough that any problems with incidental parameters is likely to be quite modest compared to other sampling errors.

As with other fixed effects estimators, conditional logit can't estimate coefficients on time-invariant regressors. Also, although it's standard to report the "log likelihood" from the estimates, this isn't at all comparable to the log likelihood from pooled logit or probit. This isn't because individuals are dropped for having all 1's or 0's—the log likelihood for those is in fact 0.0 since they are predicted perfectly by the extreme individual effect. It's because (8.4) is omitted. Since each term in that is negative, leaving it out of the reported value produces a higher log likelihood, often dramatically higher. This should be used only for comparison with alternative conditional logit models.

²It's not closed form, but the condition is the implicit equation $Y_i = \sum_t P(y_{it} = 1 | \beta, \alpha_i)$.

Subcalculations for an Individual

The conditional log likelihood isn't computed entry by entry, but individual by individual. You can use **MAXIMIZE** to estimate the model, but the sample needs to run over individuals rather than entries, and we need to be able to do the calculation over the entries which correspond to an individual. Although this is a balanced **PANEL**-dated data set, it's more convenient to use pointer vectors for the bookkeeping since they can be used for unbalanced data as well. For this data set, you can create them with:

```
panel(id=vid,identries=identries)
```

If the data set were unbalanced or blocked differently, you would add to this the option **GROUP=ID** (or whatever the series was with the individual identifiers). To avoid having to make editing changes, use those exact names for the other options (**VID** for the **VECTOR** of identifiers and **IDENTRIES** for the lists of entries). You also need to define a **FRML** which will evaluate the index function. Since we need to leave out the time-invariant variables, that will be:

```
frml(regression,vector=b,parmset=regparms) xbeta
# kids lhinc per1 per2 per3 per4
```

The **XBETA** is actually used in the working function and so should be used as the **FRML** name, but the **B** and **REGPARMS** are your choice.

The conditional log likelihood for an individual is computed using the user function **FELogitElement**. This takes as its argument the individual number. Note that this is the index number in the **VID** vector (which will range from 1 to the number of individuals), not the identifying numbers that you might be using in the data set. In this case, they are the same, but **PANEL (GROUP=...)** remaps the grouping values to start at 1.

Most of **FELogitElement** is standard and shouldn't be changed. There are two lines which use the dependent variable:

```
sstats(nopanel) 1 ti lfp(it(t))>>xOnes
```

and

```
sstats(nopanel,smpl=lfp(it(t))>0) 1 ti xbetax(t)>>actual
```

The **LFP** needs to be replaced with whatever your dependent variable is. Most of the procedure is very specific to the conditional logit, as it needs to generate all combinations of entries with the observed number of 1's. However, some parts will be used in other types of estimators that require a subcalculation over all data for an individual. First:

```
compute it=identries(indiv)
compute ti=%size(it)
dim xbetax(ti)
ewise xbetax(i)=xbeta(it(i))
```

`IT` is the list of entries for an individual and `TI` is the number of them. `XBETAX` is a `VECTOR` with the index values $X_{it}\beta$ across t for i . If you need a loop over the entries for the individual, those will be (as in the **EWISE**) in entries `IT(1)` to `IT(TI)`.

With **FELogitElement** properly adjusted for the data set, the model is estimated with something like:

```
frml felogit = log(FELogitElement(t))
maximize(parmset=regparms,pmethod=simplex,piters=5,method=bfgs,$
  title="Conditional Fixed Effects Logit",trace) felogit 1 %size(vid)
```

The key here is that the **MAXIMIZE** runs over 1 to `%SIZE(VID)`, not over the full data set.

8.2.2 Random Effects Probit

Random effects are generally applied to the probit model in binary choice applications, though the same method could be used with a logit as well. Let's return to the index formulation, but now shift the individual effect into the error term:

$$V_{it} = X_{it}\beta + \mu_i + u_{it}$$

We're still assuming that u_{it} are i.i.d. $N(0, 1)$. The μ_i are assumed to be independent of each other and all the u (and X) and are $N(0, \sigma^2)$, where the variance is unknown. Note that, as with the linear random effects model, this brings time-invariant regressors back into the analysis, and also individuals with all 1's or all 0's, since the individual effect can't be arbitrarily large.

We still assume that 1 is chosen at it if $V_{it} > 0$, which means that

$$P(y_{it} = 1) = \Phi(X_{it}\beta + \mu_i)$$

However, the μ_i are unknown, unobservable, and common to all it observations, so we can't use this in this form. Instead, we will have a single probability calculation for all the data for individual i . Conditional on μ_i and β , the likelihood for individual i is

$$f(\mu_i, \beta) = \prod_t \Phi(\tilde{y}_{it}(X_{it}\beta + \mu_i)) \quad (8.5)$$

where, for notational convenience, we use the remapped \tilde{y}_{it} which is 1 and -1 rather than 1 and 0. Note that this needs to be kept in the product form (not sum of logs). The marginal likelihood of β requires integrating (8.5) over μ_i . Unfortunately, there is no closed form expression for this integral. However, there is a well-known procedure for approximating numerical integrals when the integral can be written

$$\int_{-\infty}^{\infty} f(x) \exp(-x^2) dx$$

This is *Gauss-Hermite* numerical integration. The integral is approximated with

$$\int_{-\infty}^{\infty} f(x) \exp(-x^2) dx \approx \sum_{h=1}^H w_h f(x_h) \quad (8.6)$$

which is an H term approximation. The weights w_h and abscissas x_h are different for different values of H —the abscissas are always symmetrical about zero, and the weights match for abscissas of opposite sign. The integral in (8.6) isn't quite in the correct form, since, instead of $\exp(-x^2)$, we need to integrate against the density of a $N(0, \sigma^2)$. However, a simple change of variables gives us:

$$\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) \exp\left(-\frac{1}{2\sigma^2}x^2\right) dx \approx \sum_{h=1}^H \frac{1}{\sqrt{\pi}} w_h f(\sigma\sqrt{2}x_h) \quad (8.7)$$

The procedure `@GaussHermite` can be used to produce the weight and abscissa vectors for any number of integration terms. With the `STANDARD` option, it takes returns weights $\tilde{w}_h = \pi^{-1/2}w_h$ and abscissas $\tilde{x}_h = \sqrt{2}x_h$, so you just need to compute:

$$\sum_{h=1}^H \tilde{w}_h f(\sigma\tilde{x}_h)$$

Gauss-Hermite integration can be used to approximate any integral over the real line, since

$$\int_{-\infty}^{\infty} g(x) dx = \int_{-\infty}^{\infty} [g(x) \exp(x^2)] \exp(-x^2) dx$$

so you can apply Gauss-Hermite with $f(x) = g(x) \exp(x^2)$. However, if g has thick tails (and particular thicker tails than $\exp(-x^2)$) the accuracy of the numerical integral will be quite low.

To evaluate the log likelihood for random effects probit, we need to use Gauss-Hermite integration to compute an approximation to the integral of (8.5) over μ_i . This will give the joint probability of the observed data for individual i . The sum of the logs of this (note again that the integral is over the probabilities not log probabilities) over individuals gives the full sample likelihood. As with the conditional logit, this requires adding up the log likelihoods over individuals, not entries. This requires the same set up as described on page 67 to create the `VID` and `IDENTRIES` information. Since that's already done, we won't repeat it. We also need to define a `FRML` for the index function (again to be called `XBETA`), which for this includes all the explanatory variables, not just the time-varying ones:

```
ddv(dist=probit) lfp
# kids lhinc educ black age agesq constant per1 per2 per3 per4
frml(lastreg, parmset=regparms, vector=b) xbeta
```

The RE probit estimator requires one extra parameter: the variance (or standard deviation) of the random individual effect. We also need to create the arrays for Gauss-Hermite integration. Note that the results can (unfortunately) be fairly sensitive to the choice of the number of terms. So if you report RE probit results, you should include (at least as a footnote) the number of terms used.

```
compute ngh=15
@GaussHermite(n=ngh,w=wgh,x=xgh,standard)
nonlin(parmset=extras) sigma
```

The structure of the user function **REProbitIntegral** has quite a few similarities to **FELogitElement**. The working code is quite a bit more compact than for the logit:

```
compute REProbitIntegral=0.0
do k=1,ngh
  compute prod=1.0
  do i=1,ti
    compute z=xbetax(i)+sigma*xgh(k)
    compute prod=prod*%if(lfp(it(i)),%cdf(z),%cdf(-z))
  end do i
  compute REProbitIntegral=REProbitIntegral+wgh(k)*prod
end do k
```

The **COMPUTE PROD** instruction is the one spot that depends upon the data—replace **LFP** with your dependent variable.

While shorter in length, it isn't (generally) shorter in calculation time. On each function evaluation, it has to compute the Normal CDF for each data point for each term in the Gauss-Hermite approximation. Because of the simpler structure of the logistic CDF, each function evaluation there requires one *exp* for each individual for each shuffled combination of entries. Unless *T* is large enough that the number of combinations is quite large, that will be a quicker calculation than the random effects estimator. The other problem with RE probit is that the function values are only approximated rather than exact. While each integral is done fairly accurately, there are over 5000 of them done in each function evaluation. Also, the performance of a derivative-based optimization algorithm depends upon accurate predictions of function behavior from the gradients. Numerically computed gradients computed with even slightly inaccurate function values will cause the optimization to take longer than it would if the calculations were done to higher precision.³

Estimation is done with:

³In this case, the probit is also slower because it allows for four extra time-invariant explanatory variables.

```
compute sigma=1.0
frml REProbit = log(REProbitIntegral(t))
maximize(pmethod=simplex,piters=5,method=bfgs,$
  parmset=regparms+extras,
  title="Random Effects Probit") REProbit 1 %size(vid)
```

The 1.0 guess value for `SIGMA` should be adequate in almost all cases, as the scale of the residuals are already set by normalizing the random shock to unit variance.

Example 8.1 Non-linear Least Squares with Fixed Effects

This demonstrates estimation of a non-linear (threshold) model with fixed effects. The detailed discussion begins on page 60

```

cal(panelobs=17) 1970
open data produc.xls
data(format=xls,org=cols) 1//1970:1 48//1986:1 year p_cap hwy water $
  util pc gsp emp unemp
*
set logy = log(gsp)
set logk1 = log(p_cap)
set logk2 = log(pc)
set logl = log(emp)
*
preg(method=fixed) logy
# logk1 logk2 logl unemp
*
* Non-linear threshold effect in the unemployment rate applied to
* private capital
*
nonlin b1 b20 b21 b3 gamma c
compute b1=%beta(2),b20=%beta(3),b21=0.0,b3=%beta(4)
stats unemp
compute c=%mean,scalef=1.0/sqrt(%variance),gamma=1.0
*
frml tfrml = b1*logk1+logk2*$
  (b20+b21*%logistic((unemp-c)*scalef*gamma,1.0))+b3*logl
*
* Direct de-meaning of the non-linear formula
*
dec series tdemean
panel(indiv=1.0) logy / ymean
frml twork = tdemean+ymean
*
function Demean
set tdemean = tfrml(t)
panel(entry=1.0,indiv=-1.0) tdemean
end Demean
*
* This first get a guess value for the threshold shift b21 given guess
* values for the threshold effect. (If b21 is initialized to zero, the
* threshold coefficients don't enter the function).
*
nonlin b1 b20 b21 b3
nlls(start=Demean(),frml=twork) logy
*
* This estimates the full model
*
nonlin b1 b20 b21 b3 gamma c
nlls(start=Demean(),frml=twork) logy
*

```

```

* Using de-meaned data and searching only over the threshold parameters
*
panel(entry=1.0,indiv=-1.0) logy / cy
panel(entry=1.0,indiv=-1.0) logk1 / ck1
panel(entry=1.0,indiv=-1.0) logk2 / ck2
panel(entry=1.0,indiv=-1.0) logl / cl
*
* We'll use the same guess values as before. We don't need guess values
* for any of the regression parameters, since they will be calculated
* using a linear regression.
*
stats unemp
compute c=%mean,scalef=1.0/sqrt(%variance),gamma=1.0
*
nonlin gamma c
find min %rss
    set logk2x = logk2*%logistic((unemp-c)*scalef*gamma,1.0)
    panel(entry=1.0,indiv=-1.0) logk2x / ck2x
    linreg(noprint) cy
    # ck1 ck2 ck2x cl
end do find
linreg(print) cy
# ck1 ck2 ck2x cl

```

Example 8.2 Logit and Probit with Individual Effects

This demonstrates fixed effects (conditional) logit and random effects probit estimators.

```

open data lfp.dta
calendar(panelobs=5)
data(format=dta) 1//1 5663//5 id period lfp black educ age agesq kids hinc $
  per1 per2 per3 per4 per5 lhinc
*
preg(method=fixed) lfp
# kids lhinc per1 per2 per3 per4 per5
*
ddv(dist=probit) lfp
# kids lhinc constant per1 per2 per3 per4 educ black age agesq
*
* Get lists of the entries for each value of ID
*
panel(id=vid,identries=identries)
*****
*
* Conditional FE Logit
*
* Set up the formula leaving out the constant and any time-invariant
* variables.
*
frml(regression,vector=b,parmset=regparms) xbeta
# kids lhinc per1 per2 per3 per4
*
* This computes the partial log likelihood for individual "t". This is
* the relative probability of the individual having a particular set of
* "1"'s among all possible ways of having that many.
*
function FELogitElement indiv
type real FELogitElement
type integer indiv
*
local integer i ti nOnes done
local vect[int] it
local real sumz sumcombos xOnes actual
local vect xbetax
local vect[int] ones
*
compute it=identries(indiv)
compute ti=%size(it)
*
* Evaluate the common component in the index.
*
dim xbetax(ti)
ewise xbetax(i)=xbeta(it(i))
*
* If all values are 0 or 1, the ML value is to make the individual
* effect either -infinity or +infinity, so there is no information about

```



```

* the slope coefficients.
*
sstats(nopanel) 1 ti lfp(it(t))>>xOnes
compute nOnes=fix(xOnes)
if nOnes==0.or.nOnes==ti {
    compute FELogitElement=%na
    return
}
sstats(nopanel, smpl=lfp(it(t))>0) 1 ti xbetax(t)>>actual
*
* Set up the list of "ones"
*
dim ones(nOnes)
ewise ones(i)=i
*
* Go through all combinations which produce the proper number of ones.
* Do the exp's in an overflow-safe way by subtracting off the "actual"
* index before exping.
*
compute sumcombos=0.0
compute done=0
while .not.done {
    compute sumz=0.0
    do i=1,nOnes
        compute sumz=sumz+xbetax(ones(i))
    end do i
    compute sumcombos=sumcombos+exp(sumz-actual)
    *
    * Update the set of ones
    *
    compute done=1
    do i=nOnes,1,-1
        compute ones(i)=ones(i)+1
        if ones(i)>ti-nOnes+i
            next
        do j=i+1,nOnes
            compute ones(j)=ones(j-1)+1
        end do j
        compute done=0
        break
    end do i
}
compute FELogitElement=1.0/sumcombos
end
*
* The maximize runs over count of individuals, not the actual entries.
*
frml felogit = log(FELogitElement(t))
maximize(parmset=regparms, pmethod=simplex, peters=5, method=bfgs, $
    title="Conditional Fixed Effects Logit") felogit 1 %size(vid)
*****
ddv(dist=probit) lfp
# kids lhinc educ black age agesq constant per1 per2 per3 per4
frml(lastreg, parmset=regparms, vector=b) xbeta

```

```

*
* This is fairly sensitive to the choice of the number of nodes for the
* numerical integral.
*
compute ngh=15
@GaussHermite(n=ngh,w=wgh,x=xgh,standard)
*
nonlin(parmset=extras) sigma
*****
*
* This does the integral over the data for individual "indiv"
*
function REProbitIntegral indiv
type real REProbitIntegral
type integer indiv
*
local integer    i k ti
local vect[int] it
local real      prod z
local vect      xbetax
*
compute it=identries(indiv)
compute ti=%size(it)
*
* Evaluate the common component in the index.
*
dim xbetax(ti)
ewise xbetax(i)=xbeta(it(i))
*
* Do Gauss-Hermite integral
*
compute REProbitIntegral=0.0
do k=1,ngh
  compute prod=1.0
  do i=1,ti
    compute z=xbetax(i)+sigma*xgh(k)
    compute prod=prod*%if(lfp(it(i)),%cdf(z),%cdf(-z))
  end do i
  compute REProbitIntegral=REProbitIntegral+wgh(k)*prod
end do k
end REProbitIntegral
compute sigma=1.0
frml REProbit = log(REProbitIntegral(t))
maximize(pmethod=simplex,piters=5,method=bfgs,parmset=regparms+extras,$
  title="Random Effects Probit") REProbit 1 %size(vid)

```

Unit Root Testing

Despite over three decades of applied work aimed at determining whether unit roots exist in important macroeconomic variables, the question remains open in many cases. It's extremely difficult with a single time series of limited length to tell the difference between the permanent response to a shock implied by a unit root, and a response with a half-life of 20 quarters (dominant root roughly .97). We can't simply get more data by extending the front end of the data set, as the results from the literature on unit root testing and structural breaks show that breaks can cause false acceptances of unit roots. As in other cases, panel data can offer an alternative way to bring more data to bear on a questions. If we can't get a longer time span from one country, what about doing joint inference on multiple countries?

The general structure used by most (though not all) panel unit root testing procedures is:

$$\Delta y_{it} = \rho_i y_{i,t-1} + \sum_{l=1}^{p_i} \phi_{i,l} \Delta y_{i,t-l} + \alpha_i d_{it} + \varepsilon_{it} \quad (9.1)$$

where the d_{it} are the deterministic components. $\rho_i = 0$ means the y process has a unit root for individual i , while $\rho_i < 0$ means that the process is stationary around the deterministic part.

There have been quite a few possible procedures which have been proposed for testing unit roots in panel data. As part of the decision as to which to employ, we have to deal with the following questions:

1. What is the null hypothesis? In most cases, that will be unit root for all individuals.
2. What is the alternative?
3. What's heterogeneous and what (if anything) is homogeneous?
4. How do we deal with the small sample effects?

Item 2 is uninteresting in a single time series—if the null is unit root, the alternative is a stationary dominant root. In a panel setting, however, the alternative can be a single common dominant stationary root, or heterogeneous stationary roots, or even the rather vague “not all unit roots” (that is, some could have unit roots, but not all do).

Regarding item 3, the testing procedures almost uniformly allow the short-run dynamics (the lag polynomial in $\Delta y_{i,t}$) to differ among individuals, not just in coefficients, but also in the number of lags p_i . Because differing values of p_i mean different samples, the testing procedures need to allow for unbalanced samples. The coefficients on deterministic variables and the variance of ε_{it} will also generally be allowed to vary, which means that almost everything other (perhaps) than ρ_i will be heterogeneous. It's important to note that there are many perfectly reasonable ways to choose the lag length p_i which won't necessarily give the same result, particularly when applied to multiple short time series. As a result, there won't be a unique "correct" value for any test which relies upon lag pruning. This is also true if a test depends upon a long-run variance, as the value will depend upon the lag window chosen.

The small sample effect is embedded in the Dickey-Fuller and other such test statistics on single time series. A different calculation will be needed for each form of the test using panel data.

9.1 The Example

The data file from Example 9.1 is `pennxrate.dta`, which is a file with real exchange rate data on a balanced panel consisting of 151 countries observed over 34 years, from 1970 through 2003. The data are derived from the Penn World tables.¹ The U.S. is treated as the base country for exchange rates, and isn't included in the data set. The main series of interest is `LNRXRATE`, which is the log real exchange rate. The data file includes dummies for the G7 countries (there will be six of them, since the U.S. is the base) and OECD countries. If PPP holds, then there should *not* be a unit root in this series. The data file is read with

```
open data pennxrate.dta
calendar(panelobs=34,a) 1970
data(format=dta) 1//1970:01 151//2003:01 year xrate ppp id $
    capt realxrate lnrxrate oecd g7
```

9.2 Levin-Lin-Chu test

Levin et al. (2002) propose a test which has an alternative hypothesis that the ρ_i are identical and negative.² Because ρ_i is fixed across i , this is one of the most complicated of the tests because the data from the different individuals need to be combined into a single final regression. To isolate only the ρ_i in (9.1), the residuals from regressions of Δy_{it} and $y_{i,t-1}$ from all the "nuisance" variables (lags and deterministics) are obtained using individual by individual

¹This is an example taken from the documentation for the Stata command `xtunitroot`.

²This circulated as a working paper with just Levin and Lin, so is more commonly known as the Levin-Lin test.

regressions. This is an application of the Frisch-Waugh Theorem to a linear regression stacked across individuals.

Each individual's data is scaled down by a feasible estimate of the standard deviation of the variance of ε_{it} . This produces two series $\tilde{e}_{i,t}$ (from Δy_{it}) and $\tilde{v}_{i,t-1}$ (from $y_{i,t-1}$). The basic test statistic is the t statistic on the linear regression of $\tilde{e}_{i,t}$ on $\tilde{v}_{i,t-1}$. For a single individual, this would be identically the Dickey-Fuller t -test statistic for the given set of augmenting lags and deterministic components—that's what the Frisch-Waugh Theorem gives us. However, we can't use the tabulated D-F distribution since both the numerator and denominator are aggregated across individuals. Instead, the authors provide centering and normalizing constants such that, as $N \rightarrow \infty$ ³ an adjusted t converges to a standard Normal. In addition to centering and normalizing constants (which depend upon (average) T and the choice of deterministic), there's also a need to correct for the different long-run variances of the $\tilde{v}_{i,t}$ processes across i . For a single time series, the long-run variance cancels out of the asymptotic distribution of the t (under the null of a unit root), but won't when both numerator and denominator are first aggregated across individuals with different short-run dynamics.

As a complicated multi-step procedure, there are a number of places where two implementations can differ. In addition to possibly different choices for lag length, both the short- and long-run variance calculations can be computed either under the null that ρ is zero, or under the alternative where ρ has been estimated. And the long-run variance depends upon choice of lag window and lag length.

The procedure @LEVINLIN can be used to do this test. This has quite a few options for controlling the calculation. First, the deterministic components are selected using a standard (for unit root tests) DET option:

```
DET=NONE/[CONSTANT]/TREND
```

where the default is constant (that is, individual fixed effects since the coefficients on the deterministic variables are heterogeneous). DET=TREND adds individual-specific constant and trend.

The lag length selection for the lagged differences is governed by:

```
LAGS=(maximum) number of additional lags of the differenced series
CRIT=FIXED/[GTOS]/AIC/BIC/HQ
SLSTAY=significance level for keeping the marginal lag in CRIT=GTOS
```

These again are fairly standard for RATS procedures for ADF type analysis. The default is to do general-to-specific, dropping lags as long as the significance

³With $N/T \rightarrow 0$.

level of the t -statistic on the final one is greater than the `SLSTAY` value (.10 as default).

Finally, the lag window for estimating the long-run variance is chosen using the options:

```
LWINDOW=[NEWKEY]/BARTLETT/FLAT/PARZEN/QUADRATIC
BANDWIDTH=# of lags (or bandwidth for LWINDOW=QUADRATIC)
```

Since the `LAGS` option is already being used, the lags or bandwidth is chosen using the `BANDWIDTH` option.

In addition to these options, the `@LEVINLIN` procedure offers an alternative to compute the long-run variance. The standard calculation takes the residuals from a regression of Δy_{it} on the deterministics and applies the chosen lag window to those to get the long-run variance for an individual.⁴ Under the null, however, the regression of the Δy_{it} on both the deterministics and the augmenting lags offers an estimate of the long-run variance by recoloring: the ratio of long- to short-run variances is

$$\frac{1}{\left(1 - \sum_{l=1}^{p_i} \phi_{i,l}\right)^2}$$

You can choose this with the option `RECOLOR`, which is off by default. This has the advantage that the two variances are computed over the same range, while the standard calculation uses a longer sample range for the long-run variance (basically the full individual sample) than the short-run. It also eliminates one additional choice that could affect the test statistic.

On the example program, we use three different choices for computing the long-run variance: Bartlett (or Newey-West) windows of width 10 and 5, and the recolored AR. We use the `SMPL` option to restrict the sample to just the G7 countries.⁵

```
@levinlin(smpl=g7, lags=10, crit=aic, band=10) lnrxrate
@levinlin(smpl=g7, lags=10, crit=aic, band=5) lnrxrate
@levinlin(smpl=g7, lags=10, crit=aic, recolor) lnrxrate
```

The output from the final one of these is

```
Levin-Lin Unit Root Test: Series LNRXRATE
Test has large N, N/T-->0
Null is Unit Root. Alternative is Common Stationary Root.
Individual Specific Components: Constant
With average lags 1.00 chosen from 7
Long-run variances by recolored AR

N          6
T          34
t-unadjusted -6.73614
t-adjusted  -1.95162
Signif      0.02549
```

⁴In Baltagi (2008), Step 2 on page 276 omits the step of extracting the deterministics.

⁵The usable sample will be further restricted by lags, but that's all done internally.

The t -adjusted in the actual Levin-Lin test statistic and the significance level is the one-tailed (negative) comparison of that with a standard Normal. The bandwidth of 10 strikes me as too wide for just 34 data points. It gives a much more significant result of -3.44300. The narrower bandwidth gives -2.23023 which is more in agreement with the recolored value. While you would reject a unit root (thus accepting PPP) with any of these, we have a difference between a lukewarm rejection at a significance level of .02 vs. a rather emphatic rejection of with a p -value of 0.0003 for the wide bandwidth from different variations on the same test.

9.3 Harris-Tzavalis Test

A similar but simpler test is described by Harris & Tzavalis (1999). This also has a null of unit root versus an alternative with a single stationary value. It's designed to be applied to data sets which are relatively short in T . In order to provide relatively exact corrections for small values,⁶ they very tightly restrict the model to exclude the augmenting lags. Thus if the original panel is balanced (which they require), it will remain so. They also assume a homogeneous variance which the Levin-Lin test doesn't. The test, as implemented, uses y_{it} rather than Δy_{it} as the dependent variable, which means that the test is for $\rho = 1$ rather than $\rho = 0$. It has large N , fixed T asymptotics, again, with the centered and rescaled test statistic being $N(0, 1)$.

This is implemented in RATS using the procedure `@HTUNIT`, which is a relatively simple procedure since there are so few options. This applies the test to the G7 and OECD subsamples. Note that $N = 6$ for the first of these is probably too small.

```
@htunit(smpl=g7) lnrxrate
@htunit(smpl=oeed) lnrxrate
```

The output from the G7 test is:

```
Harris-Tzavalis Test: Series LNRXRATE
Test has fixed T, large N asymptotics
Null is rho(i)=1. Alternative is rho(i)==rho<>1
Individual Specific Components: Constant

N          6
T          34
Rho        0.80615
Z          -2.95168
Signif     0.00158
```

9.4 Im-Pesaran-Shin Test

Im et al. (2003) start with the same basic model (9.1), but, unlike Levin-Lin-Chu and Harris-Tzavalis, they allow the more general alternative that the ρ_i

⁶Levin-Lin-Chu provide a rather coarse table with (average) T values of 25, 30, 35, 40 etc.

can vary and, in fact, that some individuals can have a unit root. Of course, the power of the test diminishes quite severely if a substantial fraction have a unit root. With everything heterogeneous, the simplest approach is to compute separate ADF test statistics on each individual and combine those (by simple averaging of the t -statistics). The final test statistic is a normalized and rescaled version of this called $Z_{\bar{t}-bar}$ which has an asymptotic $N(0,1)$ distribution. This has large T -large N asymptotics.

In RATS, this is performed with the `@IPSHIN` procedure. Because it does an ADF test for each individual, it has the standard options for controlling that:

```
LAGS=(maximum) number of additional lags of the differenced series
CRIT=FIXED/[GTOS]/AIC/BIC/HQ
SLSTAY=significance level for keeping the marginal lag in CRIT=GTOS
```

Note that the limit on the number of lags is 8 as that is as high as the adjustment tables go.

The authors allow only for constant or constant and trend. However, to maintain the syntax with the other tests, the `DET` option still reads:

```
DET=NONE/[CONSTANT]/TREND
```

but `DET=NONE` generates an error.

For the G7 and full OECD countries, the instructions are:

```
@ipshin(smpl=g7,lags=8,crit=aic) lnrxrate
@ipshin(smpl=oced,lags=8,crit=aic) lnrxrate
```

The output for the G7 is

```
Im-Pesaran-Shin Unit Root Test: Series LNRXRATE
Test has large N on large T asymptotics
Null is Unit Root. Alternative is rho(i)<>1 for some i
Individual Specific Components: Constant

N          6
T          31 to 33
Avg P      1.000000 chosen from 8 by AIC
Statistic Signif Level
Z tbar     -2.902291 0.001852
Z ttildebar -2.328060 0.009954
```

9.5 Breitung Test

Breitung (2000) proposes an alternative set of procedures to Levin-Lin-Chu that use unbiased estimators rather than bias-corrected ones. First consider the case with no drift:

$$y_{it} = \alpha_i + x_{it}, \phi_i(L)x_{it} = \varepsilon_{it}$$

For simplicity, we'll assume that there are no nuisance short-run dynamics, so

$$\phi_i(L) = (1 - L) - \rho L$$

The null is $\rho = 0$ vs the alternative $\rho < 0$. (This is the same as Levin-Lin-Chu). Under the null,

$$\Delta y_{it} = \varepsilon_{it}$$

and

$$y_{it} = y_{i0} + \sum_{s=1}^t \varepsilon_{is}$$

If, instead of extracting the sample mean, we subtract y_{i0} from y_{it} , then

$$\widetilde{\Delta y}_{it} \equiv \Delta y_{it} = \varepsilon_{it}$$

and

$$\tilde{y}_{i,t-1} = y_{i,t-1} - y_{i0} = \sum_{s=1}^{t-1} \varepsilon_{is} \quad (9.2)$$

are uncorrelated by construction (under the null). Now assume that we have individual-specific trends:

$$y_{it} = \alpha_i + \beta_i t + x_{it}, \phi_i(L)x_{it} = \varepsilon_{it}$$

Now, again with no short-run dynamics, we have

$$\Delta y_{it} = \beta_i + \varepsilon_{it}$$

and

$$y_{it} = y_{i0} + \beta_i t + \sum_{s=1}^t \varepsilon_{is}$$

We can detrend y_{it} using

$$\tilde{y}_{it} = y_{it} - \left(y_{i0} + \frac{t}{T} (y_{iT} - y_{i0}) \right) = \sum_{s=1}^t \varepsilon_{is} - \frac{t}{T} \sum_{s=1}^T \varepsilon_{is}$$

However, unlike (9.2), this is a function of all ε_{is} rather than just those dated t and earlier. To produce an unbiased estimator, Δy_{it} is de-meanned using a forward operation like

$$\widetilde{\Delta y}_{it} = \Delta y_{it} - \frac{1}{T-t} \sum_{s=t+1}^T \Delta y_{is} = \varepsilon_{it} - \frac{1}{T-t} \sum_{s=t+1}^T \varepsilon_{is}$$

While both $\widetilde{\Delta y}_{it}$ and $\tilde{y}_{i,t-1}$ are functions of all ε_{is} , they are uncorrelated since the first term in $\tilde{y}_{i,t-1}$ uses only subscripts through $t-1$ and thus has zero correlation with $\widetilde{\Delta y}_{it}$ constructed with subscripts from t on, while the second is

uncorrelated with $\widetilde{\Delta y_{it}}$ because the weights are equal in the sum in $\tilde{y}_{i,t-1}$ and the weights sum to zero in $\widetilde{\Delta y_{it}}$.⁷

In both cases, in regressing $\widetilde{\Delta y_{it}}$ on $\tilde{y}_{i,t-1}$, we have a non-stationary regressor under the null. However, in the regression over the panel, it's a sum of N independent functions of non-stationary regressors. Breitung uses N before T asymptotics to eliminate the problem with that.⁸

Short-run dynamics are handled by taking deviations from regressions onto lagged differences for the difference and the lagged dependent variable (as in Levin-Lin-Chu). However, this is done *before* the detrending/de-meaning operations.⁹ $\widetilde{\Delta y_{it}}$ and $\tilde{y}_{i,t-1}$ are divided by an estimate of σ_i to correct for heterogeneous variances. The test is the t statistic from regressing the full sample (standardized) $\widetilde{\Delta y_{it}}$ on $\tilde{y}_{i,t-1}$, which has an asymptotic $N(0, 1)$ distribution.

In RATS, this is performed with the procedure `@BREITUNG`. This has the standard controls for choosing individual-specific lag lengths, though the asymptotics actually rely on fixed large T , so using different numbers of lags would invalidate that. Whether that's a problem in practice isn't clear.

LAGS=(maximum) number of additional lags of the differenced series
CRIT=FIXED/[GTOS]/AIC/BIC/HQ
SLSTAY=significance level for keeping the marginal lag in CRIT=GTOS

Again, the `DET` option has the form:

DET=NONE/[CONSTANT]/TREND

The cases where `DET=TREND` is appropriate are the ones where Breitung's procedure should perform better than the Levin-Lin-Chu and Im-Pesaran-Shin tests.

Finally, there is the option

ROBUST/[NOROBUST]

With `ROBUST`, the final regression uses an Eicker-White heteroscedasticity consistent covariance matrix in constructing the test statistic. With `NOROBUST`, it uses a standard t calculation (though without scaling by σ^2 since the data are already scaled).

⁷ $\widetilde{\Delta y_{it}}$ is actually scaled by $\sqrt{\frac{T-t}{T-t+1}}$ to standardize the variances. That doesn't change the fact that the weights sum to zero.

⁸For large N , fixed T , a central limit theorem reduces the non-stationary part to calculation of its sample mean. I'm not 100% convinced of one step in Breitung's proof, but if the final regression is done using Eicker-White standard errors, it should be fine.

⁹Levin-Lin-Chu does them at the same time.

For the OECD sample, the Breitung test is done with:

```
@breitung(smpl=oeed,det=constant) lnrxrate
```

with output

```
Breitung Unit Root Test: Series LNRXRATE
Test has large N, large T (sequential) asymptotics
Null is Unit Root. Alternative is Common Stationary Root.
Individual Specific Components: Constant
With average lags 1.59 chosen from 9

N                27
T                34
Test Statistic  -3.22993
Signif          0.00062
```

9.6 Hadri Test

Unlike the previous tests, the Hadri (2000) proposes a test where the null is stationarity. This is a generalization of the KPSS fluctuations test (Kwiatkowski et al. (1992)) for a single time series. If the residuals from the deterministic part of the series form a stationary process, the partial sums of the residuals (properly scaled) form a *Brownian Bridge*. If the residuals are non-stationary, those same partial sums should have more extreme values than would be compatible with a Brownian Bridge. The typical test statistic for univariate time series is

$$\frac{1}{T^2\psi^2} \sum_{t=1}^T S_t^2$$

where ψ^2 is the long-run variance of the residual process. The panel test statistic aggregates across i , centers and normalizes to create an asymptotically $N(0,1)$. It rejects in the right-tail (fluctuations too large).

In RATS, this is done using the `@HADRI` procedure. This has the usual `DET` option to choose the deterministic components. Again, `NONE` is included, but isn't a valid option.

```
DET=NONE/[CONSTANT]/TREND
```

There are three forms of test statistic, which govern how the scaling coefficients ψ_i^2 are determined. If the residuals are assumed to be serially uncorrelated, they can be either be the same or different across individuals. Or, you can allow for serial correlation. You choose among the three using the option:

```
VARIANCE=[HOMOGENEOUS]/HETEROGENEOUS/ROBUST
```

For the calculations robust to serial correlation, the ψ_i^2 are computed as separate long-run variance estimates across individuals. The method of computing the long-run variance is chosen using the options:

```
LWINDOW=[NEWKEY]/BARTLETT/FLAT/PARZEN/QUADRATIC
LAGS=# of lags (or bandwidth for LWINDOW=QUADRATIC)
```

just as they would be for HAC standard error calculations on a **LINREG** instruction. For the OECD data, this does one test using homogeneous variances, and one with long-run variance calculations.

```
@hadri(smpl=oeed,det=constant) lnrxrate
@hadri(smpl=oeed,det=constant,variance=robust,$
      lwindow=bartlett,lags=5) lnrxrate
```

The output from the second is:

```
Hadri LM Unit Root Test: Series LNRXRATE
Test has large N, large T asymptotics
Null is Stationary. Alternative is Some Unit Roots
Individual Specific Components: Constant
Robust to Serial Correlation, Bartlett(Newey-West) (5)

N          27
T          34
Z          6.932146
Signif 0.000000
```

Note that this gives the opposite conclusion from the other tests—since the null is stationarity, rejection means that we find fairly strong evidence of non-stationarity. All of the tests have been based upon independence across individuals. If there is, for instance, a common time component, that wouldn't be true. We can repeat tests with deviations from means at each time period with:

```
panel(entry=1.0,time=-1.0,smpl=oeed) lnrxrate / cxrate
@levinlin(smpl=oeed,lags=10,crit=aic,band=5) cxrate
@htunit(smpl=oeed) cxrate
@ipshin(smpl=oeed,lags=8,crit=aic) cxrate
@hadri(smpl=oeed,det=constant,variance=robust,$
      lwindow=bartlett,lags=5) cxrate
```

However, this doesn't change the problem with conflicting results.

Example 9.1 Panel Unit Root Tests

This implements the different unit root testing procedures described in this chapter.

```
open data pennxrate.dta
calendar(panelobs=34,a) 1970
data(format=dta) 1//1970:01 151//2003:01 year xrate ppp id $
  capt realxrate lnrxrate oecd g7
*
* Levin-Lin-Chu test (G7 only)
*
@levinlin(smpl=g7,lags=10,crit=aic,band=10) lnrxrate
@levinlin(smpl=g7,lags=10,crit=aic,band=5) lnrxrate
@levinlin(smpl=g7,lags=10,crit=aic,recolor) lnrxrate
*
* Harris-Tzavalis test
*
@htunit(smpl=g7) lnrxrate
@htunit(smpl=oecd) lnrxrate
*
* Im-Pesaran-Shin test
*
@ipshin(smpl=g7,lags=8,crit=aic) lnrxrate
@ipshin(smpl=oecd,lags=8,crit=aic) lnrxrate
*
* Breitung test
*
@breitung(smpl=oecd,det=constant) lnrxrate
*
* Hadri test
*
@hadri(smpl=oecd,det=constant) lnrxrate
@hadri(smpl=oecd,det=constant,variance=robust,$
  lwindow=bartlett,lags=5) lnrxrate
*
* Tests with deviations from common time effects
*
panel(entry=1.0,time=-1.0,smpl=oecd) lnrxrate / cxrate
*
@levinlin(smpl=oecd,lags=10,crit=aic,band=5) cxrate
@htunit(smpl=oecd) cxrate
@ipshin(smpl=oecd,lags=8,crit=aic) cxrate
@breitung(smpl=oecd,det=constant) cxrate
@hadri(smpl=oecd,det=constant,variance=robust,$
  lwindow=bartlett,lags=5) cxrate
```

Cointegration and Error Correction Models

As with unit root tests, there are many open questions about whether and how variables are cointegrated which are difficult to answer empirically because they are such long-run properties. And, as with unit root tests, panel data offers some hope for answering these questions by bringing additional data to bear on them. However, cointegration brings with it one major additional issue—are the (presumed) cointegrating vectors the same across individuals (for instance, in testing PPP) or are they allowed to be different? After discussing testing (where heterogeneous is really the clear choice for reasons we will see), we will split the description of estimation into methods applicable to heterogeneous (section 10.2) and homogeneous (section 10.3) cointegrating vectors.

10.1 Pedroni Tests

Peter Pedroni has written a number of papers on testing and estimation in *heterogeneous* cointegrated panels. These allow not just differing short-run dynamics, but also differing cointegrating vectors. For testing purposes, using heterogeneous cointegrating vectors makes sense. The simplest testing procedures use a null hypothesis that the series are *not* cointegrated, that is, that the residuals from a regression on the variables are still $I(1)$. If you use as the alternative that the series are cointegrated *and* have a common cointegrating vector, then your null actually becomes that the series are not cointegrated, *or* that they are cointegrated but don't have a common vector.¹ Allowing for heterogeneous cointegrating vectors doesn't rule out that they are, in fact, the same.

The testing method aggregates across individuals cointegration test statistics well-known in the time series literature. The procedure for doing this is called `@PANCOINT`. As with conventional cointegration testing, the first step is to test that the series involved are, in fact, integrated, that is, they have unit roots. That can be done either with the procedures from Chapter 9 or you can use `@PANCOINT` with just one input, when it calculates Im-Pesaran-Shin and Levin-Lin tests rather than the cointegration tests.

The syntax for the procedure is

¹If you estimate a common cointegrating vector, then any individual with a different vector will still have unit roots in its residuals.

```
@PANCOINT( options )
# list of series
```

This has a standard set of options for choosing the deterministic components and selecting the lag length for the short-run dynamics. In addition, it has standard `SMPL`, `PRINT` and `TITLE` options, though the `SMPL` has to be a series, rather than the formulas allowed in standard RATS instructions.

```
DET=NONE/[CONSTANT]/TREND
LAGS=(maximum) number of additional lags of the
differenced series
CRIT=FIXED/[GTOS]/AIC/BIC/HQ
SLSTAY=significance level to keep lag in model with
METHOD=GTOS [.10]
SMPL=dummy series with 0's in entries to skip
[PRINT]/NOPRINT
TITLE=title for report
```

The test statistic is generated by running through the individuals in the sample, for each regressing the first variable in the `list of series` on the remainder of the list plus the deterministics. Under the null that the series are not cointegrated, the residuals should have a unit root; under the alternative, they should be stationary.

The Engle-Granger (or ADF) type test regresses the first-differenced residuals on the lagged residuals and augmenting lagged differences, with the number of those selected based upon your `LAGS`, `CRIT` and `SLSTAY` options. The test statistic is the t on the lagged residual in that regression. This is the `adf-stat` in the output.

Two other tests (PP and RHO) are based on the tests for single time series from Phillips & Ouliaris (1990). These use the regression on the residuals without the augmenting lags and make corrections for the remaining serial correlation in a “non-parametric” way. The PP is a Phillips-Perron t test from the unit root literature and RHO is direct test of the (adjusted) regression coefficient. These require some type of windowed estimator, which in this case is a Bartlett (Newey-West) with an automatic lag length.

Each of these three test statistics has a “panel” and a “group” version. The group averages the individual test statistics and thus allows the ρ to also be heterogeneous under the alternative. The panel version sums sufficient statistics for the components of the estimates (cross products, for instance) and computes the test statistics using those. This, in effect, forces ρ to be homogeneous under the alternative.

All of these test statistics individually have a non-standard distribution, which depends upon the choice for the deterministics and on the number of $I(1)$ variables. However, each has an asymptotic mean and variance under the null.

Assuming independence across individuals, the average of each test statistic, properly centered and scaled, will have an asymptotic Normal distribution. The asymptotics of the testing procedure is $T \rightarrow \infty$ followed by $N \rightarrow \infty$.

There are two additional options:

```
UNWEIGHTED/ [NOUNWEIGHTED]
TDUM/ [NOTDUM]
```

By default the “panel” statistics are computed using weighted sums of the individual information, where the cross product weights are inversely proportional to the long-run variance.² This type of weighting gives roughly the same behavior for the aggregate test statistics as averaging the test statistics does in the group statistic.³ You can choose to use unweighted statistic with the `UNWEIGHTED` option.

If you use `TDUM`, each of the original series is transformed to deviations from time period means before any of the other calculations are done. This removes any *common* time effects.

The Example

Example 10.1 is from Pedroni (2007). It will be discussed in this and the next section. The data file has 29 countries, and 43 annual observations from 1950 to 1992. The regression of interest is log per capita GDP on the log ratio of real investment to real GDP. The original data file has the two variables in non-logged form, so we read and transform it.

```
open data pedroni-data.txt
calendar(panelobs=43,a) 1950:1
data(format=free,org=columns) 1//1950:01 29//1992:01 country $
  year pop rgdpl rinvest
set logypc = log(rgdpl)
set logiy  = log(rinvest)
```

For use in labeling the estimates of the individual cointegrating vectors, we input the country labels using the following (showing just three lines). Using a `VECT[STRINGS]` and `INPUT` like this allows you do handle country names with multiple words, like South Africa.

```
dec vect[strings] labels(29)
input labels
Egypt
Kenya
Morocco
```

²which is effectively the same as dividing an individual’s data by the square root of its long-run variance.

³Without weighting, the panel test statistic will tend to be dominated by the individuals with high $X'X$.

Again, the first step is to do unit root tests on the two variables. The tests for each variable (without time dummies) are done with:

```
@pancoint (det=constant, lags=3, crit=gtos, $
           title="Log Investment Shares-No Time Dummies")
# logiy
@pancoint (det=trend, lags=3, crit=gtos, $
           title="Log PC Income-No Time Dummies")
# logypc
```

The output from the first of these is:

```
Log Investment Shares-No Time Dummies
raw panel unit root test results

Levin-Lin rho-stat   -4.60
Levin-Lin t-rho-stat -1.15
Levin-Lin ADF-stat   -0.10

IPS ADF-stat         -1.02
(using large sample adjustment values)
```

These reject in the left tail, so the .05 critical value is -1.65 using the $N(0, 1)$ asymptotics. If we use the IPS test as the standard, this is outside the rejection zone. There isn't any ambiguity about the income variable:

```
Log PC Income-No Time Dummies
raw panel unit root test results

Levin-Lin rho-stat   5.15
Levin-Lin t-rho-stat 1.60
Levin-Lin ADF-stat   1.48

IPS ADF-stat         2.49
(using large sample adjustment values)
```

which has statistics well out in the right tail. Note that these test statistics look very different from what you would get with a comparable Engle-Granger test for a single set of time series. There, a value bigger than 0 is quite rare. Remember, however, that these all have had the mean of the base test statistic subtracted, so a positive value simply implies that the test statistics average higher than the (negative-valued) mean.

The cointegration test can be done with:

```
@pancoint (det=trend, lags=3, crit=gtos, tdum, $
           title="Cointegration Test-With Time Dummies")
# logypc logiy
```

This allows for individual trends, since income has that. This test is done with time effects removed.

```

Cointegration Test-With Time Dummies
panel v-stat      1.05
panel rho-stat    -0.93
panel pp-stat     -2.76
panel adf-stat    -1.55

group rho-stat    0.60
group pp-stat     -2.14
group adf-stat    -1.23

```

For the pair of series to be cointegrated, we would want the test statistics to be in the left tail. The PP statistics are certainly there, while the ADF statistics aren't as clear.⁴ The estimation proceeds on the basis that the series *are* cointegrated.

10.2 Estimation with Heterogeneous Cointegrating Vectors

With differing cointegrating vectors and short-run dynamics, estimates of the cointegrating vectors are done individual by individual. The standard result that linear regression produces superconsistent estimates of the cointegrating vector still holds in panel data, however, there is a substantial amount of bias in finite samples. Pedroni (2001) provides two estimators taken from the time series literature for dealing with this: extensions of Phillips-Hansen Fully Modified least squares and of Stock-Watson Dynamic OLS. Fully Modified (FM) regression estimates a linear regression, then adjusts the estimates and covariance matrix for endogeneity, while Dynamic OLS (DOLS) adds regressors to eliminate the endogeneity. In both cases, the procedure estimates both individual-specific cointegrating vectors and an aggregated estimator.

10.2.1 Fully-Modified Least Squares

Fully Modified Least Squares was proposed originally in Phillips & Hansen (1990) and extended to allow for deterministic trends in Hansen (1992). In

$$y_{1,it} = \beta'_i y_{2,it} + \text{deterministics} + u_{it} \quad (10.1)$$

where y_1 is the first of the cointegrated variables and y_2 is the vector of the remaining ones, the least squares estimate of β has substantial bias because of correlation between y_2 and u . Under the assumption that the series are cointegrated, u will be stationary, but is almost certain to be serially correlated. $\hat{\beta}$ suffers not just from simultaneity bias, but also a dynamic endogeneity due to the serial correlation. The idea behind FM is to estimate that bias and correct for it.

⁴There are some differences between the statistics reported here and those in the paper because of slight differences in the handling of lag selection. Pedroni used a fixed level for the t as the threshold, while the RATS procedure uses a fixed significance level. In a single time series, this is unlikely to make a difference, but across 29 countries, there are a few that come up with different lag lengths.

The correction is rather involved, but is the same for heterogeneous panel data as for a single set of time series. It *does* depend upon the choice of window for estimates of long-run variance. With RATS, you can implement this using the procedure `@PANELFM`. Because there is no choice for augmenting lags, there are somewhat fewer options. Note that the `LAGS` option now applies to the Bartlett (Newey-West) window width.

```
DET=NONE/[CONSTANT]/TREND
SMPL=dummy series with 0's in entries to skip
TITLE=title for report
LAGS=(maximum) number of lags in Bartlett kernel
TDUM/[NOTDUM]
```

The `PRINT` option takes a different form since you might or might not be interested in seeing the individual estimates. `PRINT=SHORT` gives just the aggregate estimate and `PRINT=FULL` gives both that and the individual level results.

```
PRINT=NONE/SHORT/[FULL]
```

The remaining options are:

```
BVEC=VECTOR with hypothesized slope coefficients [all zeros]
AVERAGE=[SIMPLE]/SQRT/PRECISION
```

`BVEC` is used if you have a specific cointegrating vector that you want to test against (such as the PPP restrictions). This is used in all the t statistics, both in the aggregate and at the individual level. The `AVERAGE` option determines how the coefficient estimates are weighted in producing the grouped estimates. `AVERAGE=SIMPLE` is an equally weighted average. `AVERAGE=SQRT` weights each individual by the diagonal matrix formed by taking the square roots of the precision matrix (inverse covariance matrix) of the estimates for that individual. This matches up with the averaging done in computing the t -statistics, that is, the coefficients and covariance matrix from `AVERAGE=SQRT` will reproduce the average t -statistics. `AVERAGE=PRECISION` weights each individual by the precision of its estimates.

The following variables are set by the procedure based upon the grouped estimates:

%BETA	group estimates of coefficients
%STDERRS	group estimates of standard errors
%TSTATS	group estimates of t -statistics (for testing <code>BVEC</code> if you include it).
%XX	estimates of covariance matrix of group estimates

The following have information on the individual-specific estimates. Except for `%IXX`, these are all matrices with dimensions regressors \times individuals. The regressors are the endogenous variables first, followed by intercept and (if needed) trend. The individuals' dimension includes only the individuals actually in the sample.

%%IBETAS	individual coefficients
%%ISTDERRS	individual standard errors
%%ITSTATS	individual <i>t</i> -statistics (for testing BVEC if you include it)
%%IXX	individual covariance matrices. This is a VECTOR of SYMM with %%IXX(<i>i</i>) as the covariance matrix for individual <i>i</i> .

The Example

Continuing with the example from page 90, the FM estimates using the full sample are done with:

```
@panel fm(det=trend, print=short, lags=2)
# logypc logiy
```

The example program then reproduces most of Table II from the paper. The coefficient estimates here aren't the same because of an error in the program used for the paper. The α in the table is the non-linear function $\alpha = \beta / (1 + \beta)$ of the slope coefficient. The standard errors are computed using the delta method. The country-specific part of the table is built with

```
do i=1,29
    compute alphai=%%ibetas(1,i)/(1+%%ibetas(1,i)), $
           stderri=%%istderrs(1,i)/(1+%%ibetas(1,i))^2
    report (row=new, atcol=1) labels(i) %%ibetas(2,i) %%ibetas(3,i) $
           %%ibetas(1,i) %%istderrs(1,i) alphai stderri
end do i
```

The following does a heterogeneity test for the slope coefficients using the estimated individual coefficient vectors and covariance matrices using the restriction matrix that each individual's coefficient is equal to the average of all.⁵ This is different from the *F* in the paper which uses sums of squared residuals, though both give very clear rejections.

```
sstats(mean) 1 %ngroup %%ixx(t)(1,1)>>meanvar %%ibetas(1,t)>>meanbeta
dec symm v(%ngroup,%ngroup)
dec vect b(%ngroup)
ewise v(i,j)=(i==j)*%%ixx(i)(1,1)-(1.0/%ngroup)*%%ixx(i)(1,1)-$
           (1.0/%ngroup)*%%ixx(j)(1,1)+1.0/(%ngroup)*meanvar
ewise b(i)=%%ibetas(1,i)-meanbeta
compute hettest=%qforminv(v,b)
cdf(title="Heterogeneity Test") chisqr hettest %ngroup-1
```

The estimates for the OECD subsample is done with

⁵Any restriction matrix which forces all the coefficients to be the same will give the same result.

```

set suboecd      = %indiv(t)==14.or.%indiv(t)>=17
@panelfm(det=trend,print=short,lags=2,smpl=suboecd)
# logypc logiy

```

10.2.2 Dynamic OLS

The general idea behind DOLS was proposed independently by a number of authors, but is most closely associated with Stock & Watson (1993), who showed how it could be used not just for a single cointegrating vector, but a space of more than one. In order to deal with the endogeneity in (10.1), DOLS adds current, lags and leads of the first difference of y_2 to the regression. This soaks up the endogeneity while leaving the (true) cointegrating vector unaffected. Theoretically, you would want infinite lags and leads, which is obviously impossible. So a truncation point needs to be set. However, DOLS can very quickly exhaust the degrees of freedom in a data set. If you choose truncation at lag p , there are $2p + 1$ added regressors in the differences for *each* right side endogenous variable, plus you lose $2p + 1$ data points allowing for lags and leads and differences. So with 20 observations per individual, two right side endogenous variables, $p = 2$ leaves you with 15 usable observations, and 13 regressors.⁶ The estimates from such an overparameterized model will likely be unusable.

This is done using RATS with the procedure **@PANELDOLS**. This is almost identical in structure to **@PANELFM** except for two options:

```

DLAGS=number of lags and leads on the differences [2]
LAGS=number of lags to use in the Bartlett kernel (for
computing the variance) [5]

```

The p in the discussion above is set using the **DLAGS** option. DOLS requires an estimate of the long-run variance of the residuals for scaling the covariance matrix. (This has no effect on the point estimates). The **LAGS** option is used for choosing the window width for that.

Example

Continuing from the previous section, the 2007 paper does all the estimation with FM. Everything could also have been done by replacing **@PANELFM** with **@PANELDOLS** and adjusting the options a bit. For instance, the analog of the original full sample FM estimates can be done by:

```

@paneldols(det=trend,print=short,dlags=2)
# logypc logiy

```

⁶constant + 2 current RHS + 2 x 5 additional lags and leads on the differences.

10.3 Estimation with Homogeneous Cointegrating Vectors

This is more involved than estimation with heterogeneous cointegration since the data have to be organized to allow estimation across individuals, despite differing short-run dynamics. In both of the methods that we will describe, the key to that preparation is the **SWEEP** instruction (Section 3.3).

10.3.1 Mark and Sul DOLS

Example 10.3.1 is from Mark & Sul (2003). They use DOLS to estimate a money demand function as a cointegrating vector connecting real money, real GDP and an interest rate. They use four different estimators, all combinations of with and without a (heterogeneous) trend and with and without a (common) time effect. The first step in all cases is to sweep out the heterogeneous components from the endogenous variables. The heterogeneous components are the augmenting lags and leads on the differences, the constant and (if needed) the trend. The residuals from the sweeping operation are then run through a separate linear regression which includes (if needed) the time dummies for the common time effect.

Computing the point estimates for the cointegrating vector is actually quite simple. This is the case with no trend and no time period dummies:

```
sweep(group=%indiv(t), series=sweeps)
# realm realy rate
# constant dy{-p to p} dr{-p to p}
*
linreg(noprint) sweeps(1)
# sweeps(2) sweeps(3)
```

where p is the number of augmenting leads and lags. Again, the caution from section 10.2.2 applies: DOLS very quickly exhausts the degrees of freedom in that preliminary regression. This application uses $p = 2$ on a $T = 40$ data set.

What's more complicated here is the calculation of the covariance matrix for the estimates. The augmentation with lags and leads of the differences is designed to produce residuals which are uncorrelated with the *regressors* at all leads and lags, but not with themselves. So we still have heterogeneous serial correlation left in those. The least squares estimator for the cointegrating vector is

$$\left(\sum_{i=1}^N \sum_{t=1}^T \tilde{\mathbf{Y}}'_{it} \tilde{\mathbf{Y}}_{it} \right)^{-1} \left(\sum_{i=1}^N \sum_{t=1}^T \tilde{\mathbf{Y}}'_{it} \tilde{z}_{it} \right)$$

where the $\tilde{\mathbf{Y}}$ are the swept-out right side endogenous variables and \tilde{z}_{it} are the residuals. For each individual, the standard DOLS asymptotics give:

$$\left(\sum_{t=1}^T \tilde{\mathbf{Y}}'_{it} \tilde{\mathbf{Y}}_{it} \right)^{-1/2} \left(\sum_{t=1}^T \tilde{\mathbf{Y}}'_{it} \tilde{z}_{it} \right) \rightarrow N(0, \text{var}(\tilde{z}_i) \mathbf{I}) \quad (10.2)$$

which admits an interpretation that the distribution of the estimator is Normal with the standard OLS covariance matrix, except with the long-run variance of the residuals taking the place of the standard variance. If we write⁷ (10.2) as

$$\left(\sum_{t=1}^T \tilde{\mathbf{Y}}'_{it} \tilde{z}_{it} \right) \approx N \left(0, lvar(\tilde{z}_i) \left(\sum_{t=1}^T \tilde{\mathbf{Y}}'_{it} \tilde{\mathbf{Y}}_{it} \right) \right)$$

and we use independence to aggregate across i , we get:

$$\left(\sum_{i=1}^N \sum_{t=1}^T \tilde{\mathbf{Y}}'_{it} \tilde{z}_{it} \right) \approx N \left(0, \sum_{i=1}^N lvar(\tilde{z}_i) \left(\sum_{t=1}^T \tilde{\mathbf{Y}}'_{it} \tilde{\mathbf{Y}}_{it} \right) \right) \quad (10.3)$$

The covariance term from this:

$$\sum_{i=1}^N lvar(\tilde{z}_i) \left(\sum_{t=1}^T \tilde{\mathbf{Y}}'_{it} \tilde{\mathbf{Y}}_{it} \right) \quad (10.4)$$

is the center term in a “sandwich” estimator of the covariance matrix, pre- and post-multiplied by the full sample $(\tilde{\mathbf{Y}}' \tilde{\mathbf{Y}})^{-1}$ matrix.

We have a separate procedure which recomputes the covariance matrix for panel DOLS using this. This, of course, depends upon the method used to compute the long-run variances (individual by individual). The `@PDOLSRobust` procedure follows Mark and Sul by using Andrews & Monahan (1992) which is an “automatic” procedure, using an AR(1) pre-whitening, followed by use of a quadratic-spectral window. Even this, however, is sensitive to the method used for choosing the AR coefficient if the residuals have a near-unit root. This could occur here if the series aren’t, in fact, cointegrated, or if they fail to be cointegrated for one or more individuals. Recoloring requires dividing the long-run variance of the AR residuals by $(1 - \rho^2)$, so if ρ is estimated very close to 1 for an individual, that individual’s component of (10.4) will tend to dominate the overall value. Our implementation of Andrews-Monahan is quite a bit more stable numerically than the one used by Mark and Sul, hence we get a match on the point estimates, but often quite different standard errors.

One thing to note is that it’s completely justifiable to use panel-robust standard errors rather than the `@PDOLSRobust` procedure. That wouldn’t be available for single time series, but would be in a panel setting, as long as $N < T$.

⁷None of these are in the proper form as probability limits, but are standard approximate statements.

10.3.2 Pesaran-Shin-Smith Pooled Mean Group

Example 10.3 is taken from Pesaran et al. (1999). They estimate a consumption function with cointegrating relationship:

$$c_{it} = \theta_{0i} + \theta_{1i}y_{it}^d + \theta_{2i}\pi_{it} + u_{it}$$

where c is log real consumption per capita, y_d is log real disposable income per capita and π is the rate of inflation. However, instead of estimating the model directly in that form, they embed it in an error correction model, with heterogeneous short-run dynamics. With one lag on the dynamics, this produces the model:

$$\Delta c_{it} = \phi_i(c_{i,t-1} - \theta_{0i} - \theta_{1i}y_{it}^d - \theta_{2i}\pi_{it}) - \delta_{1i}\Delta y_{it}^d - \delta_{2i}\Delta \pi_{it}^d + \varepsilon_{it} \quad (10.5)$$

If all the coefficients are heterogeneous, (10.5) can be estimated by least squares individual by individual. As written, this is linear in the data, non-linear in the parameters, with just-identified coefficients, so it can be estimated in the linear form and the non-linear coefficients backed out. The estimation of the linear form can be done quite simply with:

```
set dc = lpc-lpc{1}
set dy = lndi-lndi{1}
set ddp = dp-dp{1}
sweep(group=%indiv(t), var=hetero)
# dc
# lpc{1} constant lndi dp dy ddp
```

LPC is the consumption variable, LNDI is income and DP is inflation. The coefficients from the individual regressions are pooled by averaging, which gives mean group estimates; the VAR=HETERO option doesn't change the point estimates, but does change the calculation of the covariance matrix. **SWEEP** doesn't produce any displayed output, but does compute %BETA and %XX as the mean group estimate and its covariance matrix, along with (among others) %LOGL (log likelihood) and %NFREE (number of freely estimated coefficients, including the number of variances). The coefficients on the cointegrating vector are backed out using:

```
summarize(noprint) -%beta(3)/%beta(1)
disp "Income Elasticity" @30 %sumlc sqrt(%varlc)
summarize(noprint) -%beta(4)/%beta(1)
disp "Inflation Effect" @30 %sumlc sqrt(%varlc)
disp "Speed of Adjustment" @30 %beta(1) %stderrs(1)
disp "Log Likelihood" @30 %logl
disp "Estimated Parameters" @30 %nfree
```

Their Pooled Mean Group (PMG) estimator is more complicated. This fixes the θ_{1i} and θ_{2i} but allows the other variables to vary as well as the variances. While it would be possible to estimate (10.5) directly using weighted non-linear least

squares, that would require a very large parameter set to allow for all the heterogeneous coefficients. Instead, the authors use an iterative procedure which solves the first order conditions for the two blocks of coefficients (heterogeneous vs homogeneous) given the other. The log likelihood will increase at each step of this.

Because the constant in the cointegrating relation is heterogeneous, the regression with the fixed θ_1 and θ_2 can be rearranged to:

$$\Delta c_{it} = \mu_i + \phi_i(c_{i,t-1} - \theta_1 y_{it}^d - \theta_2 \pi_{it}) - \delta_{11i} \Delta y_{it}^d - \delta_{21i} \Delta \pi_{it}^d + \varepsilon_{it}$$

This is linear in the variables, and can be simplified by sweeping out the individual-specific intercept, y_{it}^d and π_{it} since their coefficients don't interact with the others. This reduces the equation to:

$$\widetilde{\Delta c_{it}} = \phi_i(\tilde{c}_{i,t-1} - \theta_1 \tilde{y}_{it}^d - \theta_2 \tilde{\pi}_{it}) + \tilde{\varepsilon}_{it}$$

Note that Δc is transformed separately—it's not the same as transforming c and then differencing. The elimination of the “nuisance” variables is done with:

```
sweep(group=%indiv(t),series=tvar)
# dc lpc{1} lndi dp
# dy ddp constant
```

This produces a 4-vector of SERIES named TVAR, where TVAR(1) is the swept-out version of DC, TVAR(2) is that for lagged consumption, etc. Given θ , the ϕ_i and the variances can be estimated by least squares individual by individual. In order to do that, we need guess values for the THETA coefficients. At least in this application, the process appears to be rather sensitive to the choice of those values, probably because a poor choice (like all zeros) leaves the unit root in the data. This uses the theoretical value of 1 on the log income, and 0 for inflation, though a better choice in practice probably would be the mean group estimates.

```
compute [vector] theta=||1.0,0.0||
dec vect phi(n) sigmasq(n)
```

This computes the transformed regressor and does the least squares estimates, saving the coefficient and estimated variance:

```
set xitilde = tvar(2)-theta(1)*tvar(3)-theta(2)*tvar(4)
do i=1,n
  linreg(noprint) tvar(1) i//1 i//tend
  # xitilde
  compute phi(i)=%beta(1),sigmasq(i)=%sigmasq
end do i
```

The θ are then estimated by a regression in the form:

$$\widetilde{\Delta c_{it}} - \phi_i \tilde{c}_{i,t-1} = \theta_1 (-\phi_i \tilde{y}_{it}^d) + \theta_2 (-\phi_i \tilde{\pi}_{it}) + \tilde{\varepsilon}_{it}$$

This is again linear in transformed data. Because of the presumed heteroscedasticity, this needs to be estimated by weighted least squares.

```
set uvar(1) = ix=%indiv(t),tvar(1)-phi(ix)*tvar(2)
do j=3,4
    set uvar(j) = ix=%indiv(t),-phi(ix)*tvar(j)
end do j
set spread = sigmasq(%indiv(t))
linreg(spread=spread,noprint) uvar(1)
# uvar(3) uvar(4)
compute theta=%beta
```

Given the new θ , we now need to re-estimate the ϕ . This is repeated until convergence. Because of the way it's constructed, at convergence, the sum of squared residuals on the final regression should be equal to the number of data points.⁸ Thus the check for breaking the iteration loop is:

```
if abs(%rss/%nobs-1)<=1.e-8
    break
```

Outside the loop, the final estimates are created and displayed using:

```
linreg(spread=spread,form=chisquared,title="Pooled Mean Group") uvar(1)
# uvar(3) uvar(4)
*
compute loglpmg=%logl
disp "Income Elasticity" @30 %beta(1) %stderrs(1)
disp "Inflation Effect" @30 %beta(2) %stderrs(2)
```

We use `FORM=CHISQUARED` because the residual variances were already built into the regression. Note that the standard errors are computed as if the ϕ and variances are known. A mean group estimate of the speed of adjustment can be computed using the final transformed variables:

```
sweep(group=%indiv(t),var=hetero)
# tvar(1)
# xitilde
disp "Speed of Adjustment" @30 %beta(1) %stderrs(1)
```

The authors also present dynamic fixed effects estimates of the reduced form model. This is, of course, subject to the bias problem from Chapter 7. However, so are the other estimators in the paper since they all involve lagged dependent variables and individual de-measured data. It's hoped that the T dimension is large enough in this application that that can safely be ignored.

⁸This is because we used the `%SIGMASQ` estimate for the variance rather than the degrees of freedom corrected `%SEESQ`.

Example 10.1 Panel Cointegration with Heterogeneous Cointegrating Vectors

This is adapted from Pedroni (2007). It tests for cointegration and estimates a model with heterogeneous cointegrating vectors. The original paper used only FM. This also includes for illustration, estimation by DOLS. The detailed discussion begins on page 90.

```
open data pedroni-data.txt
calendar(panelobs=43,a) 1950:1
data(format=free,org=columns) 1//1950:01 29//1992:01 country $
    year pop rgdpl rinvest
*
* Countries are (in order): Egypt, Kenya, Morocco, Nigeria, South
* Africa, Dominican Republic, Bolivia, Brazil, Chile, Columbia,
* Paraguay, Peru, Venezuela, Japan, Philippines, Sri Lanka, Austria,
* Belgium, Denmark, Finland, West Germany, Ireland, Italy,
* Luxembourg, Netherlands, Norway, Spain, Sweden, Turkey.
*
dec vect[strings] labels(29)
input labels
Egypt
Kenya
Morocco
Nigeria
South Africa
Dominican Republic
Bolivia
Brazil
Chile
Columbia
Paraguay
Peru
Venezuela
Japan
Philippines
Sri Lanka
Austria
Belgium
Denmark
Finland
West Germany
Ireland
Italy
Luxembourg
Netherlands
Norway
Spain
Sweden
Turkey
*
set logypc = log(rgdpl)
```

```

set logiy = log(rinvest)
*
* Do unit root tests
*
@pancoint(det=constant,lags=3,crit=gtos,$
  title="Log Investment Shares-No Time Dummies")
# logiy
@pancoint(det=constant,lags=3,crit=gtos,tdum,$
  title="Log Investment Shares-With Time Dummies")
# logiy
@pancoint(det=trend,lags=3,crit=gtos,$
  title="Log PC Income-No Time Dummies")
# logypc
@pancoint(det=trend,lags=3,crit=gtos,tdum,$
  title="Log PC Income-With Time Dummies")
# logypc
*
* Do cointegration test
*
@pancoint(det=trend,lags=3,crit=gtos,tdum,$
  title="Cointegration Test-With Time Dummies")
# logypc logiy
*
* Do panel FM estimates. (These are somewhat different from those
* in the paper due to an error in the original calculations).
*
@panelfm(det=trend,print=short,lags=2)
# logypc logiy
*
report(action=define,title="Table II: Individual Coefficient Estimates")
report(atrow=1,atcol=1,align=left) "Country"
report(atrow=1,atcol=2,align=center) "Intercept" "Trend" "Slope" $
  "Std Err" "$\alpha_i$" "std err"
do i=1,29
  compute alphai=%ibetas(1,i)/(1+%ibetas(1,i)),$
    stderri=%istderrs(1,i)/(1+%ibetas(1,i))^2
  report(row=new,atcol=1) labels(i) %ibetas(2,i) %ibetas(3,i) $
    %ibetas(1,i) %istderrs(1,i) alphai stderri
end do i
report(action=format,picture="*.###")
report(action=format,picture="*.###",atcol=3,tocol=3)
report(action=show)
*
* Heterogeneity test (slope coefficient only)
*
sstats(mean) 1 %ngroup %%ixx(t) (1,1)>>meanvar %ibetas(1,t)>>meanbeta
dec symm v(%ngroup,%ngroup)
dec vect b(%ngroup)
ewise v(i,j)=(i==j)*%%ixx(i)(1,1)-(1.0/%ngroup)*%%ixx(i)(1,1)-$
  (1.0/%ngroup)*%%ixx(j)(1,1)+1.0/(%ngroup)*meanvar
ewise b(i)=%ibetas(1,i)-meanbeta
compute hettest=%qforminv(v,b)
cdf(title="Heterogeneity Test") chisqr hettest %ngroup-1
*

```

```

report(action=define,title="Table III: Panel Group Mean Coefficients")
report(atrow=1,atcol=1,align=left) "Sample"
report(atrow=1,atcol=2,align=center) "Intercept" "Trend" "Slope" $
  "Std Err" "$\alpha_i$" "std err"
report(row=new,atcol=1) "full panel (29)" %beta(2) %beta(3) %beta(1) $
  %stderrs(1) %beta(1)/(1+%beta(1)) %stderrs(1)/(1+%beta(1))^2
*
set sub23      = %indiv(t)<>3.and.%indiv(t)<>4.and.%indiv(t)<>9.and.$
               %indiv(t)<>18.and.%indiv(t)<>19.and.%indiv(t)<>28
set suboecd    = %indiv(t)==14.or.%indiv(t)>=17
set subnotoecd = 1-suboecd
*
* Do subsamples
*
@panel fm(det=trend,print=short,lags=2,smpl=sub23)
# logypc logiy
report(row=new,atcol=1) "subset (23)" %beta(2) %beta(3) %beta(1) $
  %stderrs(1) %beta(1)/(1+%beta(1)) %stderrs(1)/(1+%beta(1))^2
*
@panel fm(det=trend,print=short,lags=2,smpl=subnotoecd)
# logypc logiy
report(row=new,atcol=1) "non-oecd (15)" %beta(2) %beta(3) %beta(1) $
  %stderrs(1) %beta(1)/(1+%beta(1)) %stderrs(1)/(1+%beta(1))^2
*
@panel fm(det=trend,print=short,lags=2,smpl=suboecd)
# logypc logiy
report(row=new,atcol=1) "oecd (14)" %beta(2) %beta(3) %beta(1) $
  %stderrs(1) %beta(1)/(1+%beta(1)) %stderrs(1)/(1+%beta(1))^2
*
report(action=format,picture="*.###")
report(action=format,picture="*.###",atcol=3,tocol=3)
report(action=show)
*
* Panel DOLS estimates
*
@panel dols(det=trend,print=short,dlags=2)
# logypc logiy

```

Example 10.2 Panel Cointegration with Homogeneous Cointegrating Vector

This is taken from Mark & Sul (2003). It uses DOLS to estimate a money demand function with a common cointegrating vector. The detailed discussion is in Section 10.3.1.

```
cal(panelobs=40)
open data panelmoney.xls
data(org=cols,format=xls) 1//1 19//40 realm realy rate
*
* Number of lag/leads to use on the differences
*
compute p=2
*
* Create differences needed for DOLS, individual-specific time
* trend.
*
set dy    = realy-realy{1}
set dr    = rate-rate{1}
*
set time  = %period(t)
*
* Create time period dummies
*
dec vect[series] tdummies(40)
do i=1,40
    set tdummies(i) = %period(t)==i
end do i
*
* Case one - no time trend, no common time effect.
*
* Get rid of effects of heterogeneous elements.
*
sweep(group=%indiv(t),series=sweeps)
# realm realy rate
# constant dy{-p to p} dr{-p to p}
labels sweeps
# "Money" "Income" "Rate"
*
* Estimate the linear regression on the swept out variables. Do the
* covariance matrix correction and show the output.
*
linreg(noprint) sweeps(1)
# sweeps(2) sweeps(3)
@PDOLSRobust %resids
linreg(create,lastreg,form=chisquared,title="Panel DOLS")
*
* Case two - no time trend, with common time effect. (There are some
* extra dummies in tdummies that are for periods lost in taking
* leads and lags. That has no effect on the results.) This uses the
* same sweep variables as case one.
*
```

```
linreg(noprint) sweeps(1)
# sweeps(2) sweeps(3) tdummies
@PDOLSRobust %resids
linreg(create,lastreg,form=chisquared,title="Panel DOLS")
*
* Case three - heterogeneous time trend, no common time effect
*
sweep(group=%indiv(t),series=sweeps)
# realm realy rate
# constant dy{-p to p} dr{-p to p} time
linreg(noprint) sweeps(1)
# sweeps(2) sweeps(3)
@PDOLSRobust %resids
linreg(create,lastreg,form=chisquared,title="Panel DOLS")
*
* Case four - heterogeneous time trend, common time effect.
* This uses the same sweep variables as case 3
*
linreg(noprint) sweeps(1)
# sweeps(2) sweeps(3) tdummies
@PDOLSRobust %resids
linreg(create,lastreg,form=chisquared,title="Panel DOLS")
```

Example 10.3 Panel Error Correction Model

This is taken from Pesaran et al. (1999). It estimates a consumption function in error correction form with heterogeneous speeds of correction, but common cointegrating vector. The details are in Section 10.3.2.

```
cal(panelobs=34) 1960
open data oecddata.xls
data(format=xls,org=columns,missing=8934567) 1//1960:1 24//1993:1 $
    lpc lpdi lndi dp
compute n=24,tend=34
*
set dc = lpc-lpc{1}
set dy = lndi-lndi{1}
set ddp = dp-dp{1}
*
* MG estimates
*
sweep(group=%indiv(t),var=hetero)
# dc
# lpc{1} constant lndi dp dy ddp
*
* Back out estimates for the income elasticity and inflation
* effect. The point estimates will differ slightly from those in
* the article, since those compute averages of the coefficients
* from the (equivalent) non-linear functional form, while these
* compute averages in the linear form and convert back. The
* standard errors are different because (we assume) those in the
* article take into account that the sigma's are estimated.
*
summarize(noprint) -%beta(3)/%beta(1)
disp "Income Elasticity" @30 %sumlc sqrt(%varlc)
summarize(noprint) -%beta(4)/%beta(1)
disp "Inflation Effect" @30 %sumlc sqrt(%varlc)
disp "Speed of Adjustment" @30 %beta(1) %stderrs(1)
disp "Log Likelihood" @30 %logl
disp "Estimated Parameters" @30 %nfree
*
* Estimate of PMG
*
dec vect[series] uvar(4)
*
* Get residuals from projections onto the "nuisance" variables.
* This is done individual by individual. Note that the error
* correction term actually leaves out the intercept as they are
* allowing that to vary, while constraining the slope coefficients.
*
sweep(group=%indiv(t),series=tvar)
# dc lpc{1} lndi dp
# dy ddp constant
*
* Initial guess value for the slope coefficients in the
* cointegrating vector. It appears that the behavior of the
```



```

* estimation process is quite sensitive to good guess values for
* this.
*
compute [vector] theta=||1.0,0.0||
dec vect phi(n) sigmasq(n)
*
do iters=1,60
  *
  * Estimate the speeds of adjustment (phi's) for each individual
  *
  set xitilde = tvar(2)-theta(1)*tvar(3)-theta(2)*tvar(4)
  do i=1,n
    linreg(noprint) tvar(1) i//1 i//tend
    # xitilde
    compute phi(i)=%beta(1),sigmasq(i)=%sigmasq
  end do i
  *
  * Filter through by the phi's to get the transformed regressors
  * for re-estimating the theta's
  *
  set uvar(1) = ix=%indiv(t),tvar(1)-phi(ix)*tvar(2)
  do j=3,4
    set uvar(j) = ix=%indiv(t),-phi(ix)*tvar(j)
  end do j
  set spread = sigmasq(%indiv(t))
  linreg(spread=spread,noprint) uvar(1)
  # uvar(3) uvar(4)
  compute theta=%beta
  *
  * This should eventually stabilize with %rss==%nobs
  *
  if abs(%rss/%nobs-1)<=1.e-8
    break
end do iters
*
* This is the final estimate for the PMG. It looks as if there's an
* error in the log likelihood for this branch.
*
linreg(spread=spread,form=chisquared,title="Pooled Mean Group") uvar(1)
# uvar(3) uvar(4)
*
compute loglpmg=%logl
disp "Income Elasticity" @30 %beta(1) %stderrs(1)
disp "Inflation Effect" @30 %beta(2) %stderrs(2)
sweep(group=%indiv(t),var=hetero)
# tvar(1)
# xitilde
disp "Speed of Adjustment" @30 %beta(1) %stderrs(1)
disp "Log Likelihood" @30 loglpmg
disp "Estimated Parameters" @30 n*(3+2)+2
*
* Estimate of fixed effects
*
preg(method=fixed,title="Dynamic Fixed Effects") lpc

```

```
# lndi lndi{1} dp dp{1} lpc{1}
*
summarize(noprint) (%beta(1)+%beta(2))/(1-%beta(5))
disp "Income Elasticity" @30 %sumlc sqrt(%varlc)
summarize(noprint) (%beta(3)+%beta(4))/(1-%beta(5))
disp "Inflation Effect" @30 %sumlc sqrt(%varlc)
summarize(noprint) -(1-%beta(5))
disp "Speed of Adjustment" @30 %sumlc sqrt(%varlc)
disp "Log Likelihood" @30 %logl
disp "Estimated Parameters" @30 %nfree
```

Simulation and Bootstrap Methods

11.1 Linear Random Effects

With random effects, we don't need to concern ourselves with the difference between time-varying and time-invariant regressors, so we simplify the linear model (with individual random effects) to:

$$y_{it} = \alpha + X_{it}\beta + (\mu_i + \eta_{it}) \quad (11.1)$$

We're assuming that μ_i are i.i.d. $N(0, \sigma_\mu^2)$ across i and η_{it} are i.i.d. $N(0, \sigma_\eta^2)$ across both i and t . The parameters that we estimated before were α , β and the variances and we still need to estimate those. However, we can treat the μ_i as parameters as well, which eliminates the complicated correlation among observations with each individual. The steps in a Gibbs sampler will be to do (in some order):

1. Sample α and β given the μ_i and σ_η^2
2. Sample σ_η^2 given the residuals from step 1.
3. Sample μ_i given the regression coefficients and variances.
4. Sample σ_μ^2 given the μ_i

Of these, the only one that's not completely standard is step 3. Step 1 is simple because with μ_i treated as parameters, we can subtract them from y_{it} and sample the coefficients from the adjusted regression.

In Example 11.1, we will use the Grunfeld investment data set used in Example 4.1 for fixed effects. This is the model used in Koop (2003) as the example for Bayesian analysis of random effects. We'll first estimate the model with maximum likelihood components:

```
open data grunfeld.xls
calendar(panelobs=20, a) 1935
all 10//1954:01
data(format=xls, org=columns) 1//1935:01 10//1954:01 $
    firm year invest value cap
*
preg(method=random, vcomp=ml) invest
# constant value cap
```

The results from Gibbs sampling would be expected to be somewhat similar to these, as the Gibbs sampler will trace out the shape of the likelihood function. The setup code for the sampler is almost as long as the code for handling the draws themselves. Instead of going through that first, we'll look at the four steps in the sampler in order and show both the draw code and the setup code for each. You can tell them apart by the indenting—Gibbs loop code is indented while the setup code isn't.

We will be initializing the sampler with the OLS estimates, so the individual effects will be zero. We are treating those as parameters, so we need to keep track of them. We will be using a VECTOR named `ADRAW` for that.

```
compute [vector] adraw =%zeros(nindiv,1)
```

We need to draw the regression coefficients using the adjusted equation:

$$\tilde{y}_{it} \equiv y_{it} - \mu_i = \alpha + X_{it}\beta + \eta_{it} \quad (11.2)$$

Because it's easiest to do draws from a linear equation by first computing the cross product of the regressors and dependent variable, as part of the setup code we will define an equation with \tilde{y} (which we will call `YADJUST`) as the dependent variable to use in computing the cross product.

```
set yadjust = invest
linreg(define=eqn) yadjust
# constant value cap
```

The cross product matrix is computed inside the loop using:

```
set yadjust = invest-adraw(%indiv(t))
cmom(equation=eqn)
```

This has to be done in the loop since the “dependent variable” changes each time through. From these, we can draw the regression coefficients with:

```
compute bdraw =%ranmvpostcmom(%cmom,hdraw,hprior,bprior)
```

`HDRAW` is the regression precision, which is the current value of σ_η^{-2} —precision is generally more convenient than the corresponding variance in Bayesian analysis. Although it isn't necessary to initialize `BDRAW` since it gets drawn right away, the values set in the startup code are:

```
compute [vector] bdraw =%xsubvec(%beta,1,%nreg)
compute hdraw =1.0/%seesq
```

We need the prior mean (`BPRIOR`) and precision matrix (`HPRIOR`) for the regression coefficients. We want a non-informative prior, so

```
compute [vector] bprior=%zeros(1,%nreg)
compute [symm] hprior=%zeros(%nreg,%nreg)
```

(The values of `BPRIOR` don't matter if `HPRIOR` is all zeros). The regression precision can be drawn immediately with

```
compute rssplus=nuprior*s2prior+%rsscmom(%cmom,bdraw)
compute hdraw  =%ranchisqr(nuprior+%nobs)/rssplus
```

The `S2PRIOR` and `NUPRIOR` are the scale parameter (effectively the mean) and degrees of freedom for the prior on the regression variance (done as an inverse chi-squared). Informative priors (`NUPRIOR` bigger than zero) for variances can be a bit arbitrary. In some cases, they're necessary. However, this has 200 observations, so we can quite safely use a non-informative prior:

```
compute s2prior=0.0
compute nuprior=0.0
```

(Again, `S2PRIOR` has no effect if `NUPRIOR` is zero).

This brings us to step 3: drawing the individual effects. We can rearrange (11.1) to

$$y_{it} - \alpha - X_{it}\beta = \mu_i + \eta_{it} \quad (11.3)$$

Since the μ_i are assumed to be independent across i , this is just inference on the sample means of

$$\varepsilon_{it} \equiv y_{it} - \alpha - X_{it}\beta$$

for each i separately. The data information on μ_i has mean equal to the sample mean and precision equal to T times the current `HDRAW` (precision version of σ_η^2). The prior is mean 0 and precision σ_μ^{-2} , which we will call `HADRAW`. The mean of the posterior for μ_i is the precision-weighted average:

$$\frac{\bar{\varepsilon}(T\sigma_\eta^{-2}) + 0}{T\sigma_\eta^{-2} + \sigma_\mu^{-2}}$$

and it's precision is the denominator (which will be called `HMEAN` in the code below). The draws can be done with:

```
do i=1,nindiv
  sstats(smpl=%indiv(t)==i) / $
  invest-%eqnvalue(eqn,t,bdraw)>>sumepsi
  compute hmean=hdraw*%nobs+hadraw
  compute adraw(i)=sumepsi*hdraw/hmean+%ran(sqrt(1.0/hmean))
end do i
```

which uses `SSTATS` with a `SMPL` option to compute the mean of the raw residuals for one individual at a time.

The final step is to draw the variance (or actually the precision) of the “pool” from which the individual effects are drawn. This is the straightforward draw for the variance of Normal population since we now are treating the μ_i as data.

```

compute nuvpost=nindiv+nuvprior
compute s2vpost=(%normsqr(adraw)+s2vprior*nuvprior)/nuvpost
compute hadraw=%ranchisqr(nuvpost)/(s2vpost*nuvpost)

```

where `S2VPRIOR` and `NUVPRIOR` are the scale and degrees of freedom for the inverse chi-squared prior for σ_μ^2 . However, unlike the previous situation where we drew the *regression* precision, this is *not* the place for a non-informative prior. First off, the natural degrees of freedom are *much* smaller (10 rather than 200). But more important, because of how this is used, a non-informative prior could give rise to an *absorbing state* in the Gibbs sampler. In the case of the regression precision, an abnormally high value for the regression precision (that is, low variance) has no lasting effect—the model doesn’t and can’t fit perfectly, so the next draw for the precision won’t be strongly influenced by the low previous value. However, an abnormally low value for σ_μ^2 becomes “self-fulfilling”. If `HADRAW` is high, then, on the next sweep the draws for *mu* will be pulled down towards zero, which will mean that, with little variation among the *mu*, the next draw for σ_μ^2 will also probably be small. If you do enough sweeps, you are very likely to, at some point, get stuck in a low σ_μ^2 state. Although $\sigma_\mu^2 = 0$ is a permitted value, you don’t want to incorrectly infer that from a poorly designed sampler.

Almost any informative prior, even with very small degrees of freedom, will prevent this problem. What we are using is

```

compute s2vprior=50.0^2
compute nuvprior=2.0
compute hadraw =0.0

```

The standard deviation of 50 is somewhat data-determined, but this is quite loose. This also includes the initial value for `HADRAW` that we used above in drawing the μ .

This takes care of everything but the bookkeeping. We need to choose the number of burn-in draws and keeper draws. In this case, we’ll do 10000 of each, which takes almost no time on a model this simple:

```

compute nburn =10000
compute ndraws=10000

```

We want to keep track of the draws for regression coefficients and the individual effects. Outside the loop, we initialize the storage for this with:

```

dec series[vect] bgibbs
dec series[vect] agibbs
*
gset bgibbs 1 ndraws = %zeros(%nreg,1)
gset agibbs 1 ndraws = %zeros(nindiv,1)

```

while inside the loop, we do:

Table 11.1: Random Effects Estimator

Linear Regression - Estimation by Least Squares			
Dependent Variable YADJUST			
Panel(20) of Annual Data From 1//1935:01 To 10//1954:01			
Usable Observations		200	
Degrees of Freedom		197	
Centered R ²		0.8124080	
R-Bar ²		0.8105035	
Uncentered R ²		0.8710896	
Mean of Dependent Variable	145.95825000		
Std Error of Dependent Variable	216.87529623		
Standard Error of Estimate	94.40840333		
Sum of Squared Residuals	1755850.4841		
Regression F(2,197)	426.5757		
Significance Level of F	0.0000000		
Log Likelihood	-1191.8024		
Durbin-Watson Statistic	0.3582		
	Variable	Coeff	Std Error
1.	Constant	-42.714	9.512
2.	VALUE	0.116	0.006
3.	CAP	0.231	0.025

```
compute bgibbs(draw)=bdraw
compute agibbs(draw)=adraw
```

The draws are converted to means and standard errors with:

```
@mcmcpostproc(ndraws=ndraws,mean=bmean,stderrs=bstderrs) bgibbs
@mcmcpostproc(ndraws=ndraws,mean=amean,stderrs=astderrs) agibbs
```

and we use a set of REPORT directives to format the results. The output from PREGRESS is in Table 11.1

while the Gibbs sampler output (subject to slight Monte Carlo differences) is in Table 11.2 which is remarkably similar. The differences would likely be more pronounced with a data set with smaller T .

11.2 Random Effects Probit

The random effects probit model of Section 8.2 is likely to be an even better candidate for treatment with MCMC techniques than the linear model. The alternative to simulation techniques for marginalizing out the individual effects is time-consuming (and potentially inaccurate) numerical integration. If we look at the index model for random effects probit:

$$V_{it} = X_{it}\beta + \mu_i + u_{it} \quad (11.4)$$

the standard MCMC handling for the probit model is to treat the V_{it} as additional parameters. The only information we have on those is whether V_{it} is

Table 11.2: Random Effects by Gibbs Sampling

Variable	Coeff	Std Error
Constant	-58.264	31.196
VALUE	0.110	0.011
CAP	0.308	0.017
Indiv-1	-9.336	43.627
Indiv-2	157.901	31.861
Indiv-3	-172.214	31.718
Indiv-4	30.253	31.367
Indiv-5	-53.915	32.634
Indiv-6	34.561	31.901
Indiv-7	-7.520	32.573
Indiv-8	1.056	31.174
Indiv-9	-27.610	31.982
Indiv-10	50.703	32.683

bigger than zero or not, so this requires sampling from a truncated Normal. However, if we have β and μ_i , that step is identical to the step in the MCMC estimation of a standard probit. And if we have V_{it} , then (11.4) is just a linear random effects model, and we just saw how to handle that in Section 11.1 (with one less step since the equation variance is pegged at 1).

In Example 11.2, we will start the sampler with the standard probit estimates, with `BDRAW` used for the probit coefficients and `ADRAW` for the `VECTOR` of individual effects (which will be zero to start).

```
ddv(dist=probit) lfp
# kids lhinc educ black age agesq constant per1 per2 per3 per4
*
set y          = lfp
set yadjust    = lfp
equation(lastreg) eqn yadjust
compute bdraw=%beta
compute [vector] adraw
compute adraw=%zeros(nindiv,1)
```

Again, we'll use a non-informative prior on the regression coefficients:

```
compute [vector] bprior=%zeros(1,%nreg)
compute [symm]    hprior=%zeros(%nreg,%nreg)
```

As with the linear random effects model, we need a somewhat informative prior on the precision of the pool for the random effects. Here, though, the scale is already set by the fact that the variance of u process is pegged to one. We'll use a standard deviation of .5 with 2 degrees of freedom.

```
compute s2vprior=.5^2
compute nuvprior=2.0
compute hadraw    =0.0
```


This time, we'll keep track of the regression coefficients and the standard deviation of the random effects, which are the parameters estimated by the maximum likelihood method. The setup for those is:

```
dec series[vect] bgibbs
dec series[vect] sgibbs
*
gset bgibbs 1 ndraws = %zeros(%nreg,1)
gset sgibbs 1 ndraws = %zeros(1,1)
```

That takes care of the setup code. Inside the Gibbs loop, we need to first draw the values of the “utility” given the probit coefficients and the current individual effects. That is done with:

```
set ystar = z=%eqnvalue(eqn,t,bdraw)+adraw(%indiv(t)), $
%if(y,%rantruncate(z,1.0,0.0,%na),%rantruncate(z,1.0,%na,0.0))
```

Given that, everything else is largely the same as with the linear model. The regression coefficients are drawn, using the pegged value of 1.0 for the regression precision:

```
set yadjust = ystar-adraw(%indiv(t))
cmom(equation=eqn)
compute bdraw =%ranmvpostcmom(%cmom,1.0,hprior,bprior)
```

We don't need to draw the regression precision since it's 1. This draws the random effects, again using the fact that the regression precision is 1:

```
set eps = ystar-%eqnvalue(eqn,t,bdraw)
do i=1,nindiv
  sstats i//1 i//5 eps>>sumepsi
  compute hmean=%nobs+hadraw
  compute adraw(i)=sumepsi/hmean+%ran(sqrt(1.0/hmean))
end do i
```

Drawing the precision of the random effects is identical to before:

```
compute nuvpost=nindiv+nuvprior
compute s2vpost=(%normsqr(adraw)+s2vprior*nuvprior)/nuvpost
compute hadraw=%ranchisqr(nuvpost)/(s2vpost*nuvpost)
```

We save the two desired pieces of information with, where we convert the precision to the standard deviation used in the maximum likelihood estimation in Example 8.2.

```
compute bgibbs(draw)=bdraw
compute sgibbs(draw)=1.0/sqrt(hadraw)
```

In the post-loop processing, we'll compute and draw an empirical estimate for the sigma with:

Table 11.3: Maximum Likelihood Estimates of RE Probit

Random Effects Probit - Estimation by BFGS					
Convergence in 53 Iterations. Final criterion was 0.0000057 <= 0.0000100					
Panel(5) of Undated Data From 1//1 To 1133//3					
Usable Observations		5663			
Function Value		-8855.6708			
	Variable	Coeff	Std Error	T-Stat	Signif
1.	KIDS	-0.4405	0.0426	-10.3458	0.0000
2.	LHINC	-0.2314	0.0417	-5.5520	0.0000
3.	EDUC	0.2348	0.0214	10.9786	0.0000
4.	BLACK	0.4066	0.1231	3.3021	0.0010
5.	AGE	0.2755	0.0268	10.2680	0.0000
6.	AGESQ	-0.0043	0.0003	-13.0090	0.0000
7.	Constant	-3.0327	0.5932	-5.1121	0.0000
8.	PER1	0.1795	0.0430	4.1757	0.0000
9.	PER2	0.1420	0.0449	3.1609	0.0016
10.	PER3	0.0754	0.0451	1.6713	0.0947
11.	PER4	0.0596	0.0449	1.3285	0.1840
12.	SIGMA	2.9260	0.0493	59.3410	0.0000

```

set svalues 1 ndraws = sgibbs(t) (1)
density(grid=automatic,smoothing=1.5) svalues / xs fs
scatter(style=line,vmin=0.0)
# xs fs

```

With 10000 burn-in and 10000 keeper draws, this takes a while to run, about twice as long as the maximum likelihood estimates with a relatively small (15 term) Gauss-Hermite integral. The results are surprisingly different. Maximum likelihood (Table 11.3) gets a fairly sharp estimate for σ near 3, while Gibbs sampling (Table 11.4) comes up with an estimated value near 4 and gives effectively zero mass to anything below 3.5.

Given that we have a generally quite non-informative prior and that the only part that is informative is in the other direction (our prior on σ is centered at .5), the results are quite surprising. However, if we were to revisit the maximum likelihood estimates and greatly increase the number of terms (to, for instance, 45), we would find that, when the approximated log likelihood is computed more accurately, it produces results much more similar to the Gibbs sampler. The sensitivity of maximum likelihood to number of terms used is obviously quite high in this case. Unlike a numerical integral on a fixed interval where increasing the number of terms simply changes the density of the abscissas, with Gauss-Hermite, it also spreads them out, allowing for more accurate evaluation of the tails. Here, that obviously makes a big difference.

Table 11.4: Gibbs Sampling Estimates of RE Probit

Variable	Coeff	Std Error
KIDS	-0.6374	0.0507
LHINC	-0.2263	0.0434
EDUC	0.2920	0.0238
BLACK	0.9833	0.2196
AGE	0.5154	0.0510
AGESQ	-0.0072	0.0006
Constant	-7.9379	0.9585
PER1	0.1973	0.0482
PER2	0.1484	0.0498
PER3	0.0720	0.0500
PER4	0.0564	0.0503
sigma	3.9779	0.1193

11.3 Bootstrapping

The ideal situation for bootstrapping is a data set which can reasonably be described as i.i.d. With that, you can resample complete observations to generate a bootstrapped data set. It's more difficult with time series models, since shuffling data entries will break the ordering. The most commonly used way to handle such a data set is to use a *parametric bootstrap*, which requires estimating a model which produces (roughly) uncorrelated residuals. The derived residuals are then a reasonable candidate for bootstrapping. Those are shuffled, and the parametric model is then used to rebuild a bootstrapped data series.

Panel data gives us additional problems, in the sense that the data can have correlations in both directions, but also can offer additional opportunities. There are basically five ways to bootstrap in panel data:

1. You can resample all i, t .
2. You can resample i , then resample t within the record for i .
3. You can resample t , then resample i within the record for t .
4. You can resample i with the time direction fixed.
5. You can resample t with the individual direction fixed.

Type 1 is most similar to bootstrapping in cross section data, but requires the strongest assumptions about the behavior of the data. The other types are only possible in panel data, and allow bootstrapping in situations where they would be impossible for single time series or single cross sections.

Let's look at some alternatives for bootstrapping unit root tests. The test statistics from Chapter 9 are all based upon double asymptotics which may be far from holding in practice. If we look at the panel version of the ADF test:

$$\Delta y_{it} = \rho_i y_{i,t-1} + \sum_{l=1}^{p_i} \phi_{i,l} \Delta y_{i,t-l} + \alpha_i d_{it} + \varepsilon_{it} \quad (11.5)$$

there are several potential problems with applying bootstrapping to this. Clearly, we have to do a parametric bootstrap. The standard method would be to estimate (11.5) under the null (that is, with $\rho_i = 0$), and use that to simulate data. Estimating (11.5) will give us sample residuals ε_{it}^* . What can we do with those? If our underlying assumption is that the ε_{it} are i.i.d. across both i and t , we could bootstrap those across individuals. However, only the very simple Harris-Tzavalis test assumed that the variance is constant across individuals. If, instead, we assume that ε_{it} is i.i.d. within individuals, and independent across individuals but with differing variances, we can rescale by individual-specific standard deviations to get standardized residuals to resample. The tests other than HT aren't sensitive to rescaling of individuals, so rebuilding data with standardized residuals won't be a problem. However, the $\alpha_i d_{it}$ terms *are* sensitive to scale and are specific to a given i . This, however, can be finessed by simply generating the resampled data with $\alpha_i = 0$. As long as we do the tests with the original deterministic regressors, it doesn't matter what the coefficients are when you generate the data.

The real problem is with the pre-sample values for Δy . The reconstructed data residuals only help with the data range from $t = p_i + 2$ on. To rebuild a full data series, we will need to resample values of $\Delta y_{i,t-l}$ for the first part of the data. There are three principal ways to handle this from the literature on single time series. The standard reference on this is Berkowitz & Kilian (2000).

1. Treat them as fixed.
2. Generate them from bootstrapped standardized residuals by using the stationary distribution of the AR model.
3. Resample as a consecutive block of p_i values from somewhere in the data range.

Method 1 is obviously the simplest, but the observed initial conditions could be unrepresentative of the entire process. An example of Method 2 is for an AR(1) process with coefficient ρ , the draw for the initial value (leaving out the deterministics) would be

$$\frac{\sigma \varepsilon_{i1}^*}{\sqrt{1 - \rho^2}}$$

where ε_{i1}^* is a standardized bootstrap draw from the residuals. A potentially serious problem with that is that the autoregression might be non-stationary, in which case there won't be a stationary distribution from which to draw. That's less likely to be a problem in this application since the dominant root is being handled by differencing.

One problem with applying Methods 1 or 3 with panel data is that the actual data for individual i are compatible only with the matching individual deterministics $\alpha_i d_{it}$. Note also that Method 3 can't be applied (easily) if the deterministics include anything other than a constant, since a random block of data

from elsewhere in the data range won't match up with the deterministic part at the start of the sample.

For our application (Example 11.3), we will be looking at the Im-Pesaran-Shin test for the G7 data from Example 9.1. This has only six individuals (US is numeraire), so the large N part of the asymptotics are clearly not reasonable. That uses only `CONSTANT` for the deterministic part, so we don't have to worry about that last issue. To deal with the initial values, we will use the second type of bootstrap—resample i , then resample t within i .

The first step is to do the IPS test and save the test statistics. The standard IPS test is the normalized one that (theoretically) is $N(0, 1)$ asymptotically. We will also save the average t statistic. These should give very similar bootstrapped p-values, only differing because the centering and normalizing constants for the z-statistic depend (slightly) upon the number of lags.

```
@ipshin(smpl=g7,lags=8,crit=aic) lnrxrate
compute sampleips=%cdstat
compute sampletbar=%tbar
```

To make it easier to work with the data, we'll copy it over to short and standard names:

```
set y = lnrxrate
set dy = lnrxrate-lnrxrate{1}
```

We will keep the residuals in the original positions rather than moving around the raw data, so we use the following to get a list (in `VID`) of the individual numbers for the covered data):

```
panel(smpl=g7,id=vid)
```

We run the regressions for each of the individuals under the null and save a separate equation for each. We also save the residuals into a single series.

```
compute nindiv=%size(vid)
compute ntime =%panelobs()
dec vect[equation] eqns(nindiv)
do i=1,nindiv
  linreg(smpl=%indiv(t)==vid(i),define=eqns(i)) dy
  # constant dy{1 to %ilags(i)}
  set(smpl=%indiv(t)==vid(i)) u %regstart() %regend() = %resids
end do i
```

While we're keeping the original data and the residuals in their natural positions, the bootstrapped data will be moved up to the first `NINDIV` positions.

```
set yboot 1//1 nindiv//ntime = 0.0
set dyboot 1//1 nindiv//ntime = 0.0
```

The first step inside the bootstrap loop is to resample the individuals. This draws (with replacement) six integers in the range of 1 to 6 into the `SERIES[INTEGER]` named `INDIVSHUFFLE`:

```
boot indivshuffle 1 nindiv
```

We then build the bootstrapped data one individual at a time. To simplify the coding, we copy information about this into shorter names. `IS` will be the number between 1 and 6 that is serving as the individual whose data we're using in building bootstrapped individual `J` and `II` is the original position of that source individual's data.

```
do j=1,nindiv
  compute is=indivshuffle(j)
  compute ii=fix(vid(is))
  compute pj=laglengths(is)
```

The first thing we need to do is to draw the initial values. The differenced data still needs to start at time period 2 and the source is any set of PJ consecutive entries between 2 and T . That's most easily done by using the `BLOCK` option on `BOOT`:

```
if pj>0 {
  boot(block=pj) timeshuffle j//2 j//(1+pj) ii//2 ii//ntime
  set(nopanel) dyboot j//2 j//(1+pj) = dy(timeshuffle(t))
}
```

The `BOOT` instruction does exactly what we want—it draws a set of consecutive entries from the correct entry range in individual number `II`.¹ The `SET` instruction copies values from the original `DY` series into the bootstrapped version. The `NOPANEL` option is needed since this is crossing individual boundaries, so we don't want the panel-data treatment which would make the values `NA`.

We then need to draw residuals for the rest of the data. The source range will be `II//(2+PJ)` to `II//NTIME`; the target range is the same set of entries within individual `J`. The residuals are copied over to the proper entries in the `SHOCKS` series, which is then input as shocks into the `FORECAST` instruction using the `PATHS` option. This uses the saved equation for the source individual.

```
boot timeshuffle j//(2+pj) j//ntime ii//(2+pj) ii//ntime
set(nopanel) shocks j//(2+pj) j//ntime = u(timeshuffle(t))
forecast(paths,from=j//(2+pj),to=j//ntime,noprint)
# eqns(is) dyboot
# shocks
```

The final step in this reconstructed individual is to rebuild `Y` for the full range. The pre-sample value for the `Y` is pegged at zero.

¹Ordinarily, `BOOT` with `BLOCK` would draw repeated blocks of the indicated size, but here we only ask for PJ values, so it just does one block.

```
set yboot j//1 j//ntime = %if(%period(t)==1,0.0,yboot{1}+dyboot)
```

With a bootstrapped data set complete, we redo the `IPSHIN` procedure (with `NOPRINT`) and save the computed statistics.

```
@ipshin(lags=8,crit=aic,noprint) yboot
compute ipszstat(draw)=%cdstat
compute ipstbar(draw) =%tbar
```

After we're done with all the draws, the following first computes the bootstrapped p -value for the original IPS z statistic by computing the fraction of bootstrapped values which were smaller.² This also computes the empirical density of the z statistics (Figure 11.1). The actual density clearly has a mean below zero and is skewed left, so the asymptotic p value is more significant than it should be, but the test still rejects at conventional significance levels.

```
sstats(mean) 1 ndraws (ipszstat(t)<sampleips)>>ipssignif
density(grid=automatic,smoothing=1.5) ipszstat / zx zf
set n01 = %density(zx)
spgraph(footer="Bootstrapped IPS z-bar statistics")
scatter(style=line,vmin=0.0,hgrid=sampleips) 2
# zx zf
# zx n01
grtext(position=upleft) $
  "Bootstrapped p-value "+%strval(ipssignif,"#.###")
spgraph(done)
```

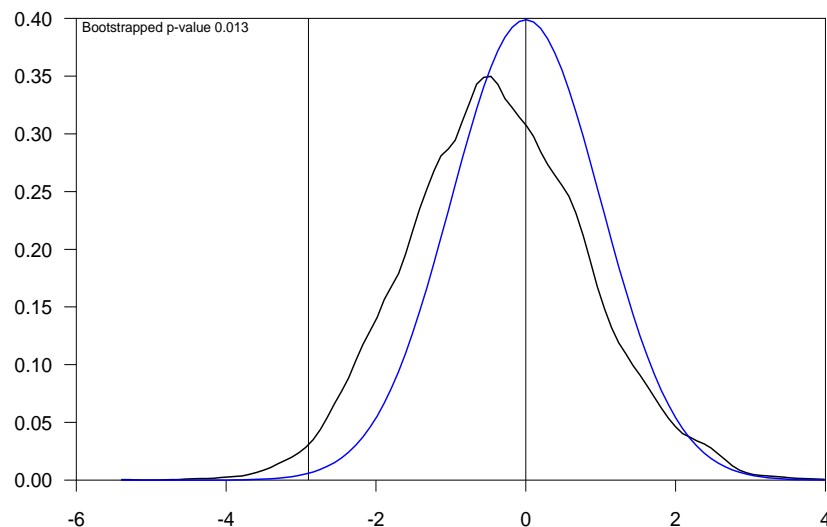


Figure 11.1: Bootstrapped Distribution of Unit Root test

²The IPS test rejects in the left tail

Example 11.1 Random Effects by Gibbs Sampling

This estimates a linear random effects model by Gibbs sampling. The detailed description starts on page 109.

```
open data grunfeld.xls
calendar(panelobs=20,a) 1935
all 10//1954:01
data(format=xls,org=columns) 1//1935:01 10//1954:01 $
    firm year invest value cap
*
preg(method=random,vcomp=ml) invest
# constant value cap
*
compute nindiv=10
*
* We define the equation in terms of <<yadjust>> (which is just the
* original dependent variable at first) since we need to replace the
* dependent variable with a mean-corrected version later on.
*
set yadjust = invest
linreg(define=eqn) yadjust
# constant value cap
*
* Start with OLS (zeros for the individual effects)
*
compute [vector] adraw  =%zeros(nindiv,1)
compute [vector] bdraw  =%xsubvec(%beta,1,%nreg)
compute hdraw  =1.0/%seesq
*
* Prior mean and precision for the regression coefficients
*
compute [vector] bprior=%zeros(1,%nreg)
compute [symm]   hprior=%zeros(%nreg,%nreg)
*
* (Non-informative) prior for equation variance.
*
compute s2prior=0.0
compute nuprior=0.0
*
* Prior for the precision of the common pool for the individual
* intercepts
*
compute s2vprior=50.0^2
compute nuvprior=2.0
compute hadraw  =0.0
*
* This needs a high burn-in because the intercept in the regression is
* highly correlated with the individual effects, but is drawn separately.
*
compute nburn  =10000
compute ndraws=10000
*
```



```

dec series[vect] bgibbs
dec series[vect] agibbs
*
gset bgibbs 1 ndraws = %zeros(%nreg,1)
gset agibbs 1 ndraws = %zeros(nindiv,1)
*
infobox(action=define,progress,lower=-nburn,upper=ndraws) $
  "Random Effects-Gibbs Sampling"
do draw=-nburn,ndraws
  *
  * Compute y's adjusted for the current individual intercepts
  *
  set yadjust = invest-adraw(%indiv(t))
  cmom(equation=eqn)
  *
  * Draw regression coefficients given hdraw
  *
  compute bdraw =%ranmvpostcmom(%cmom,hdraw,hprior,bprior)
  *
  * Draw residual precision conditional on the new coefficients
  *
  compute rssplus=nuprior*s2prior+%rsscmom(%cmom,bdraw)
  compute hdraw =%ranchisqr(nuprior+%nobs)/rssplus
  *
  * Draws individual effects given betas, h and ha
  *
  do i=1,nindiv
    sstats(smpl=%indiv(t)==i) / $
      invest-%eqnvalue(eqn,t,bdraw)>>sumepsi
    compute hmean=hdraw*%nobs+hadraw
    compute adraw(i)=sumepsi*hdraw/hmean+%ran(sqrt(1.0/hmean))
  end do i
  *
  * Draw the precision for the individual effects pool given the
  * current draws. %normsqr(adraw) computes the sum of squares of the
  * alphas.
  *
  compute nuvpost=nindiv+nuvprior
  compute s2vpost=(%normsqr(adraw)+s2vprior*nuvprior)/nuvpost
  compute hadraw=%ranchisqr(nuvpost)/(s2vpost*nuvpost)
  infobox(current=draw)
  if draw<=0
    next
  *
  * Do the bookkeeping here.
  *
  compute bgibbs(draw)=bdraw
  compute agibbs(draw)=adraw
end do draw
infobox(action=remove)
@mcmcpostproc(ndraws=ndraws,mean=bmean,stderrs=bstderrs) bgibbs
@mcmcpostproc(ndraws=ndraws,mean=amean,stderrs=astderrs) agibbs
report(action=define)
report(atrow=1,atcol=1,align=center) "Variable" "Coeff" $

```

```

"Std Error"
do i=1,%nreg
  report(row=new,atcol=1) %eqnreglabels(0)(i) bmean(i) $
    bstderrs(i)
end do i
report(row=new)
do i=1,nindiv
  report(row=new,atcol=1) "Indiv-"+i amean(i) astderrs(i)
end do i
report(action=format,atcol=2,tocol=3,picture="*.###")
report(action=format,atcol=4,picture="*.##")
report(action=show)

```

Example 11.2 Random Effects Probit by Gibbs Sampling

This estimates a random effects probit model using Gibbs sampling. The discussion starts on page 114.

```

open data lfp.dta
calendar(panelobs=5)
data(format=dta) 1//1 5663//5 id period lfp black educ age agesq kids hinc $
  per1 per2 per3 per4 per5 lhinc
*
compute nindiv=%panelsize()
*
* Start with a standard probit model
*
ddv(dist=probit) lfp
# kids lhinc educ black age agesq constant per1 per2 per3 per4
*
* To make it easier to adapt to different data, we'll copy the dependent
* variable to "Y"
*
set y          = lfp
set yadjust = lfp
equation(lastreg) eqn yadjust
compute bdraw=%beta
compute [vector] adraw
compute adraw=%zeros(nindiv,1)
*
* Prior mean and precision for the regression coefficients
*
compute [vector] bprior=%zeros(1,%nreg)
compute [syymm]  hprior=%zeros(%nreg,%nreg)
*
* Prior for the precision of the common pool for the individual
* intercepts.
*
compute s2vprior=.5^2
compute nuvprior=2.0
compute hadraw =0.0

```

```

*
* This needs a high burn-in because the intercept in the regression is
* highly correlated with the individual effects, but is drawn separately.
*
compute nburn =10000
compute ndraws=10000
*
dec series[vect] bgibbs
dec series[vect] sgibbs
*
gset bgibbs 1 ndraws = %zeros(%nreg,1)
gset sgibbs 1 ndraws = %zeros(1,1)
*
infobox(action=define,progress,lower=-nburn,upper=ndraws) $
  "Random Effects-Gibbs Sampling"
do draw=-nburn,ndraws
  *
  * Draw ystar's given the current values for the individual effects
  * and the regression coefficients.
  *
  set ystar = z=%eqnvalue(eqn,t,bdraw)+adraw(%indiv(t)), $
    %if(y,%rantruncate(z,1.0,0.0,%na),%rantruncate(z,1.0,%na,0.0))
  *
  * Compute ystar's adjusted for the current individual intercepts
  *
  set yadjust = ystar-adraw(%indiv(t))
  cmom(equation=eqn)
  *
  * Draw regression coefficients. The regression precision is pegged to 1.
  *
  compute bdraw =%ranmvpostcmom(%cmom,1.0,hprior,bprior)
  *
  * Draws individual effects given betas, h (which is 1) and ha
  *
  set eps = ystar-%eqnvalue(eqn,t,bdraw)
  do i=1,nindiv
    sstats i//1 i//5 eps>>sumepsi
    compute hmean=%nobs+hadraw
    compute adraw(i)=sumepsi/hmean+%ran(sqrt(1.0/hmean))
  end do i
  *
  * Draw the precision for the individual effects pool given the
  * current draws. %normsqr(adraw) computes the sum of squares of the
  * alphas.
  *
  compute nuvpost=nindiv+nuvprior
  compute s2vpost=(%normsqr(adraw)+s2vprior*nuvprior)/nuvpost
  compute hadraw=%ranchisqr(nuvpost)/(s2vpost*nuvpost)
  infobox(current=draw)
  if draw<=0
    next
  *
  * Do the bookkeeping here.
  *

```

```

        compute bgibbs(draw)=bdraw
        compute sgibbs(draw)=1.0/sqrt(hadraw)
end do draw
infobox(action=remove)
@mcmcpstproc(ndraws=ndraws,mean=bmean,stderrs=bstderrs) bgibbs
@mcmcpstproc(ndraws=ndraws,mean=smean,stderrs=sstderrs) sgibbs
report(action=define)
report(atrow=1,atcol=1,align=center) "Variable" "Coeff" $
    "Std Error"
do i=1,%nreg
    report(row=new,atcol=1) %eqnreglabels(0)(i) bmean(i) $
        bstderrs(i)
end do i
report(row=new,atcol=1) "sigma" smean(1) sstderrs(1)
report(action=show)
*
set svalues 1 ndraws = sgibbs(t)(1)
density(grid=automatic,smoothing=1.5) svalues / xs fs
scatter(style=line,vmin=0.0)
# xs fs

```

Example 11.3 Bootstrapping a Unit Root Test

This bootstraps a panel unit root test. The detailed discussion begins on page 119.

```

open data pennxrate.dta
calendar(panelobs=34,a) 1970
data(format=dta) 1//1970:01 151//2003:01 year xrate ppp id $
    capt realxrate lnrxrate oecd g7
*
* Do the Im-Pesaran-Shin test on the G7 data (six countries, since US is
* numeraire). Save the test statistics.
*
@ipshin(smpl=g7,lags=8,crit=aic) lnrxrate
compute samplelips=%cdstat
compute sampletbar=%tbar
*
* Copy the data and first difference to series with standard and shorter
* names.
*
set y = lnrxrate
set dy = lnrxrate-lnrxrate{1}
*
* Get information about the data in the G7 so we can find it easily.
*
panel(smpl=g7,id=vid)
*
* For each individual in the subset, rerun the regressions with the
* chosen number of lags under the null. Save separate equations for
* each. Create a series with the combined residuals.

```

```

*
compute nindiv=%size(vid)
compute ntime =%panelobs()
dec vect[equation] eqns(nindiv)
do i=1,nindiv
    linreg(smpl=%indiv(t)==vid(i),define=eqns(i)) dy
    # constant dy{1 to %%ilags(i)}
    set(smpl=%indiv(t)==vid(i)) u %regstart() %regend() = %resids
end do i
*
* These will be filled in with the generated data series. We are putting
* them into locations at the start of the data series.
*
set yboot 1//1 nindiv//ntime = 0.0
set dyboot 1//1 nindiv//ntime = 0.0
*
* Make a copy of this since %%ilags will get overwritten.
*
compute laglengths=%%ilags
*
* 5000 is excessive in practice. 500 or 1000 is more than enough to
* determine whether a test statistic is significance at conventional
* levels.
*
compute ndraws=5000
*
* We'll keep track of the generated z-bar and t-bar statistics
*
set ipszstat 1 ndraws = 0.0
set ipstbar 1 ndraws = 0.0
infobox(action=define,progress,lower=1,upper=ndraws) $
    "Bootstrapping"
do draw=1,ndraws
    *
    * Resample the individuals
    *
    boot indivshuffle 1 nindiv
    do j=1,nindiv
        compute is=indivshuffle(j)
        compute ii=fix(vid(is))
        compute pj=laglengths(is)
        if pj>0 {
            boot(block=pj) timeshuffle j//2 j//(1+pj) ii//2 ii//ntime
            set(nopanel) dyboot j//2 j//(1+pj) = dy(timeshuffle(t))
        }
        boot timeshuffle j//(2+pj) j//ntime ii//(2+pj) ii//ntime
        set(nopanel) shocks j//(2+pj) j//ntime = u(timeshuffle(t))
        forecast(paths,from=j//(2+pj),to=j//ntime,noprint)
        # eqns(is) dyboot
        # shocks
        set yboot j//1 j//ntime = %if(%period(t)==1,0.0,yboot{1}+dyboot)
    end do j
    @ipshin(lags=8,crit=aic,noprint) yboot
    infobox(current=draw)

```

```

        compute ipszstat(draw)=%cdstat
        compute ipstbar(draw) =%tbar
end do draws
infobox(action=remove)
*
* Graph empirical density of IPS test statistic
*
sstats(mean) 1 ndraws (ipszstat(t)<sampleips)>>ipssignif
density(grid=automatic,smoothing=1.5) ipszstat / zx zf
set n01 = %density(zx)
spgraph(footer="Bootstrapped IPS z-bar statistics")
scatter(style=line,vmin=0.0,hgrid=sampleips) 2
# zx zf
# zx n01
grtext(position=upleft) $
    "Bootstrapped p-value "+%strval(ipssignif,"#.###")
spgraph(done)
*
* Graph empirical density of IPS t-bar statistic
*
sstats(mean) 1 ndraws (ipstbar<sampletbar)>>tbarsignif
density(grid=automatic,smoothing=1.5) ipstbar / tx tf
spgraph(footer="Bootstrapped IPS t-bar statistics")
scatter(style=line,vmin=0.0,hgrid=sampletbar) 1
# tx tf
grtext(position=upleft) $
    "Bootstrapped p-value "+%strval(tbarsignif,"#.###")
spgraph(done)

```

Mean Group and Related Estimators

In the linear regression

$$y_{it} = X_{it}\beta_i + u_{it} \quad (12.1)$$

the pooled least squares estimator can be rewritten in the form

$$\hat{\beta} = \left(\sum_{i=1}^N \mathbf{X}_i' \mathbf{X}_i \right)^{-1} \sum_{i=1}^N (\mathbf{X}_i' \mathbf{X}_i) \hat{\beta}_i \quad (12.2)$$

where \mathbf{X}_i is the matrix of explanatory variables for individual i only and $\hat{\beta}_i$ is the least squares estimate for the data just for individual i . This is a matrix-weighted average of the individual estimates with weights $(\mathbf{X}_i' \mathbf{X}_i)$. If the data were homoscedastic with common variance σ^2 , then $\sigma^{-2} (\mathbf{X}_i' \mathbf{X}_i)$ would be the inverse covariance matrix or *precision* of $\hat{\beta}_i$, so (12.2) can be described as a *precision-weighted average* of the individual estimates.

Weighting estimates by precision would make sense if we are convinced that there is, indeed, a common β since the more precise estimators would give us better information. However, if, instead, we think that the β *aren't* the same, but are “similar”, then precision-weighting might not be the best choice. Also, the pooled estimator weights by the *actual* precision only if the variances are homogeneoneous. If they aren't, then since the true precision is proportional to σ_i^{-2} , pooling overweights individuals with high residual variances. As a result, estimators of a central β will generally use a more conservative weighting method, or will more explicitly model the idea of the β_i being similar but not identical.

12.1 Mean Group Estimator

This can't really be simpler—the overall estimator is just the mean of the individual estimates:

$$\hat{\beta} = \frac{1}{N} \sum_{i=1}^N \hat{\beta}_i \quad (12.3)$$

Under the assumption that these are all estimating a common β , the covariance matrix of the estimator is

$$\text{var}(\hat{\beta}) = \frac{1}{N^2} \sum_{i=1}^N \hat{\sigma}_i^2 (\mathbf{X}_i' \mathbf{X}_i)^{-1} \quad (12.4)$$

that is, N^{-2} times the sum of the individual covariance matrices. If the data set is unbalanced, the calculations can be adjusted slightly to overweight the larger individual samples, with

$$\hat{\beta} = \frac{\sum_{i=1}^N T_i \hat{\beta}_i}{\sum_{i=1}^N T_i}, \text{var}(\hat{\beta}) = \frac{\sum_{i=1}^N T_i^2 \hat{\sigma}_i^2 (\mathbf{X}_i' \mathbf{X}_i)^{-1}}{\left(\sum_{i=1}^N T_i \right)^2} \quad (12.5)$$

though, in practice, with roughly equal-sized samples the simpler equally-weighted estimator is generally used.

There are two ways to handle this type of model with RATS. The first uses the procedure `@MEANGROUP`. Applied to the Grunfeld investment equation used in Example 4.1, this is

```
@meangroup invest
# constant value cap
```

The other way to do this is to use `SWEEP`. Again, `SWEEP` just does the number-crunching without any displayed output. The following gives roughly the same results as `@MEANGROUP`:

```
sweep(variance=heterogeneous,group=%indiv(t),average=simple)
# invest
# constant value cap
```

The only difference is in the scaling of the covariance matrix—`@MEANGROUP` uses degrees of freedom corrected estimates for $\hat{\sigma}_i^2$ while `SWEEP` uses uncorrected (maximum likelihood) estimates. The `VARIANCE=HETEROGENEOUS` option has no effect on the point estimates of the coefficients, but *does* change how the covariance matrix is computed. `SWEEP` defines `%BETA` and `%XX` as the coefficients and covariance matrix. These can be displayed using (for instance) `LINREG` with `CREATE`, which is also what is done in `@MEANGROUP`:

```
linreg(create,coeffs=%beta,covmat=%xx,form=chisquared,$
title="Pooled Mean Group using SWEEP") invest
# constant value cap
```

`FORM=CHISQUARED` is necessary because the `%XX` matrix already includes the scaling by the variances.

12.2 Swamy Random Coefficients Models

Swamy (1970) estimated a model with explicit allowance for coefficient vectors which were “similar”. His model was (12.1) combined with

$$\beta_i = \beta + v_i, v_i \sim N(0, \Delta) \text{ i.i.d.} \quad (12.6)$$

This generalizes the random effects model of Chapter 5—if Δ is a matrix with zeros everywhere but the diagonal element for the intercept, you have random effects. The random coefficients model allows the slope coefficients to differ (usually slightly) from the central β . The random coefficients model not only permits estimation of that central value, but also provides more reliable estimates of the individual β_i than you would get using simple individual by individual estimates—it’s a form of “shrinkage” estimator which in practice gives a lower mean squared error.

For now, we will assume that Δ is known, and that the residuals are Normally distributed with known variances σ_i^2 . If you write out the stacked system substituting out for β_i and shifting the v_i into the error term, you get a GLS model with covariance matrix

$$\Omega = \text{diag} \{ \mathbf{X}_i \Delta \mathbf{X}_i' + \sigma_i^2 I_{T_i} \}$$

a block diagonal matrix with blocks corresponding to each individual. This has an inverse which can be computed block by block, thus requiring no larger than an inverse of a $T_i \times T_i$. However, the entire calculation can be simplified considerably by using results from Bayesian analysis for multivariate Normals in Appendix D. The standard result is that data evidence for individual i can be summarized as:

$$\beta_i \sim N \left(\hat{\beta}_i, \hat{\sigma}_i^2 (\mathbf{X}_i' \mathbf{X}_i)^{-1} \right)$$

$\beta_i \sim N(\beta, \Delta)$ can be switched around to the equivalent $\beta \sim N(\beta_i, \Delta)$. Combining the data information with this gives us

$$\beta \sim N \left(\hat{\beta}_i, \Delta + \hat{\sigma}_i^2 (\mathbf{X}_i' \mathbf{X}_i)^{-1} \right)$$

as the inference for β using data for individual i . Combining this across i gives mean

$$\hat{\beta} = \left(\sum_{i=1}^N \left(\Delta + \hat{\sigma}_i^2 (\mathbf{X}_i' \mathbf{X}_i)^{-1} \right)^{-1} \right)^{-1} \sum_{i=1}^N \left(\Delta + \hat{\sigma}_i^2 (\mathbf{X}_i' \mathbf{X}_i)^{-1} \right)^{-1} \hat{\beta}_i \quad (12.7)$$

and covariance matrix

$$\left(\sum_{i=1}^N \left(\Delta + \hat{\sigma}_i^2 (\mathbf{X}_i' \mathbf{X}_i)^{-1} \right)^{-1} \right)^{-1} \quad (12.8)$$

While apparently rather ugly, this reduces to (12.2) if $\Delta = 0$, as the inner double inverses cancel each other. If we define

$$\mathbf{H}_i = \left(\Delta + \hat{\sigma}_i^2 (\mathbf{X}_i' \mathbf{X}_i)^{-1} \right)^{-1} \quad (12.9)$$

which is the precision of the information about β from individual i 's data, then the estimator can be written as

$$\hat{\beta} = \left(\sum_{i=1}^N \mathbf{H}_i \right)^{-1} \sum_{i=1}^N \mathbf{H}_i \hat{\beta}_i \quad (12.10)$$

which is a different matrix weighted average of the individual least squares estimates than (12.2). In some ways, this is a cross between the pooled and mean group estimators. If $\Delta = 0$, you get pooled least squares, while if $\Delta = \kappa I$ for large κ (thus very little common information about β), the point estimate is effectively mean group.¹

The full-sample estimator for β_i is

$$\tilde{\beta}_i = \left(\sigma_i^{-2} \mathbf{X}_i' \mathbf{X}_i + \Delta^{-1} \right)^{-1} \left(\sigma_i^{-2} \mathbf{X}_i' \mathbf{X}_i \hat{\beta}_i + \Delta^{-1} \hat{\beta} \right) \quad (12.11)$$

which is a matrix-weighted average of the individual-specific least squares estimator $\hat{\beta}_i$ and the GLS estimate for the common β . This “shrinks” the estimate towards $\hat{\beta}$ where the degree of shrinkage depends upon the relative sizes of the variance coming from the data $\sigma_i^2 (\mathbf{X}_i' \mathbf{X}_i)^{-1}$ and from the shrinkage prior Δ .² Note, by the way, that this is the first place where Δ has entered into a formula in inverse form. If Δ is very large, then Δ^{-1} is effectively zero, and (not surprisingly) we don't give much weight to the overall estimate.

This isn't yet a usable estimator because we need σ_i^2 and Δ . The variances can be estimated using the residuals from the individual-specific regressions, since we have to compute those anyway. In some cases, we may have reasonable prior information about the possible spread which we can incorporate into Δ . When prior information is used, Δ will generally be diagonal. We may feel that a 95% confidence range of $\pm .2$ (standard deviation .1 or variance of .01) is reasonable for the coefficient, but have little knowledge of what the covariance might be between that coefficient and another. The diagonal Δ allows the data to determine the relationship among the coefficients.

However, good prior information won't always be available. Swamy proposed an empirical estimator for Δ . Estimate the $\hat{\beta}_i$ and use the sample covariance

¹The point estimates are the same, but the covariance matrix in this case would be very large. There's no value of Δ that gives both the mean and covariance of mean group, even in the limit.

²If there were just one regressor, the resulting estimate would have to be between the two. With more than one and a matrix-weighted average, it's possible for individual coefficients to go outside the range of the two estimates because of negative off-diagonal elements in the weighting matrices.

matrix of these (around the mean group estimator), that is

$$\Delta = \frac{1}{N-1} \sum_{i=1}^N \left(\hat{\beta}_i - \hat{\beta}_{MG} \right) \left(\hat{\beta}_i - \hat{\beta}_{MG} \right)' \quad (12.12)$$

Note that this will be singular if the number of regressors is larger than $N - 1$ and so isn't a good choice for small values of N . You could, however, estimate just the diagonal elements and leave the off-diagonal elements at zero when you have a small- N data set.

You can perform this two-step estimator using the procedure `@SWAMY`. To apply this to the example use:

```
@swamy invest
# constant value cap
```

`@SWAMY` also allows you to input the Δ matrix using the `DELTA` option, which provides a symmetric matrix the size of the regression. For instance,

```
@swamy(delta=%diag(||1600.0, .01, .01||)) invest
# constant value cap
```

which gives standard deviations of 40, .1 and .1 for the spread in the three coefficients. The procedure produces regression output for the β using (12.10). The estimated covariance matrix is the inverse weight matrix

$$\left(\sum_{i=1}^N \mathbf{H}_i \right)^{-1}$$

It also produces estimates of the individual coefficients and their covariance matrices using (12.11).³ The coefficients are in a $k \times N$ matrix `%%IBETAS`, while the covariance matrices are in a `VECTOR[SYMM]`, where the element i of that is the covariance matrix for individual i .

12.3 MCMC Estimation for Random Coefficients

The two-step Swamy estimator from section 12.2 relies upon several approximations from an actual implementation of the random coefficients model, with the use of the empirical estimates for σ_i^2 and Δ . If we knew those values, we would know exactly how to compute the GLS estimator, but what if we don't? A more complicated prior that can handle those additional unknown parameters is analytically intractable. However, with modern Bayesian techniques, we can analyze the posterior density of β and β_i .

The random coefficients assumption (12.6) is an example of a *hierarchical prior*. Instead of a direct prior on β_i , it has a prior for β_i that depends upon the further unknown parameter β . If Δ is also unknown, we need a *second* level of a

³The covariance matrix is the inverse weight matrix from that formula.

hierarchical prior. Since Δ is a covariance matrix, the natural form of prior for it is an inverse Wishart (see Appendix C.7).

If the T dimension is large, the σ_i^2 can be handled using standard methods, which means a diffuse inverse chi-squared prior. But if we apply this to data sets where T isn't large, the independent estimates of the σ_i^2 will be quite variable. Instead, a better approach is to use a hierarchical prior for that as well. For variances, the most convenient prior has

$$\sigma_i^2 = \sigma^2 r_i^2, r_i^{-2} \sim \frac{\chi_\omega^2}{\omega} \text{ i.i.d.}$$

A χ_ω^2 divided by its degrees of freedom has a mean of one and so scales up or down from the common σ^2 . A value for ω on the order of 3 or 4 is generally appropriate. The prior on the common σ^2 is most conveniently a diffuse inverse χ^2 , since it will be estimated using the full data set.

If we start out assuming Δ is known, then we have the following sets of parameters that need to be sampled (in some order):

1. β_i given β , σ_i^2 and the data
2. σ_i^2 given the residuals for individual i and σ^2
3. β given β_i
4. σ^2 given the full set of residuals and the σ_i^2

The first is represented by (12.11). $\tilde{\beta}_i$ is the mean (when we substitute the currently sampled β for $\hat{\beta}$) and $(\sigma_i^{-2} \mathbf{X}_i' \mathbf{X}_i + \Delta^{-1})^{-1}$ the variance. However, this is most conveniently done by saving the cross product matrix from the equation for each individual and using the convenience function `%RANMVPPOSTCMOM`. The “prior” for the coefficients has mean β and precision Δ^{-1} . While we're working on individual i , we can sample σ_i^2 . The posterior for it is an inverse chi-squared with degrees of freedom $T_i + \omega$ and scale equal to the sum of squared residuals plus $\omega\sigma^2$. Before the Gibbs loop, we need to compute and save the cross product matrices and T_i values, which is done with:

```
dec vect[symm] cmom(nindiv)
dec vect inobs(nindiv) irss(nindiv)
do i=1,nindiv
  cmom(smpl=%indiv(t)==i,equation=investeq)
  compute cmom(i)=%cmom
  compute inobs(i)=%nobs
end do i
```

Steps 1 and 2 are done with:

```

do i=1,nindiv
  compute betas(i)=%ranmvpostcmom(cmom(i),1.0/sigmas(i),$
    hdelta,bdraw)
  compute irss(i)=%rsscmmom(cmom(i),betas(i))
  compute rssplus=omega*sigsq+irss(i)
  compute sigmas(i)=rssplus/%ranchisqr(omega+inobs(i))
end do i

```

where `BDRAW` is the draw for the common β and `HDELTA` is the inverse of the current value of Δ . This is also saving the sum of squared residuals for the individual for use in step 4.

Step 3 is described by (12.10). The mean is the $\hat{\beta}$ and the covariance matrix is $\left(\sum_{i=1}^N H_i\right)^{-1}$. A draw for this can be done with

```

compute hbeta=%zeros(%nreg,1)
compute hsum=%zeros(%nreg,%nreg)
do i=1,nindiv
  compute hbeta=hbeta+hdelta*betas(i),hsum=hsum+hdelta
end do i
compute hsum=inv(hsum)
compute bdraw=hsum*hbeta+%ranmvnormal(%decomp(hsum))

```

And finally step 4 is done with:

```

sstats 1 nindiv irss(t)*sigsq/sigmas(i)>>rsscommon inobs(t)>>tnobs
compute sigsq=rsscommon/%ranchisqr(tnobs)

```

This assumes that a non-informative prior is used for σ^2 .

This leaves Δ as the only parameter not yet handled. The only factor in the posterior involving it is from (12.6). The kernel of that is

$$\begin{aligned}
 & |\Delta|^{-N/2} \exp \left(-\frac{1}{2} \sum_{i=1}^N (\beta_i - \beta)' \Delta^{-1} (\beta_i - \beta) \right) \\
 & = |\Delta|^{-N/2} \exp \left(-\frac{1}{2} \text{trace} \Delta^{-1} \sum_{i=1}^N (\beta_i - \beta) (\beta_i - \beta)' \right) \quad (12.13)
 \end{aligned}$$

The standard non-informative prior for Δ is $|\Delta|^{-(k+1)/2}$ which, when combined with (12.13) gives us Δ as an inverse Wishart with degrees of freedom N (where k is the number of regressors) and scale matrix

$$\sum_{i=1}^N (\beta_i - \beta) (\beta_i - \beta)' \quad (12.14)$$

This means that Δ will be centered (roughly) around the calculation in (12.12), except it will be recalculated with each sweep.

There is, however, a problem with implementing this with the non-informative prior on Δ —the Gibbs sampler has an absorbing state at $\Delta = 0$. This is the same problem encountered on page 112. If you get a draw for Δ with small elements, that forces the draws on the next sweep for β_i to be similar to β , which means that (12.14) will be small, etc. A modestly informative prior is needed to prevent collapse. Even a prior with fractional degrees of freedom will work.

We need only Δ^{-1} in all the calculations. It has a Wishart distribution rather than an inverse Wishart. We can draw that using the function %RANWISHARTF. With a prior with NUDELTA degrees of freedom and scale matrix NUDELTA*DDELTA, the draw for HDELTA can be done with:

```
compute delta=nudelta*ddelta
do i=1,nindiv
  compute delta=delta+%outerxx(betas(i)-bdraw)
end do i
compute hdelta=%ranwishartf(%decomp(inv(delta)),nindiv+nudelta)
```

The prior that we are using in this example is:

```
compute nudelta=0.1
compute ddelta =%diag(%fill(%nreg,1,1.0))
```

The sampling process is extremely quick since there are sufficient statistics in the cross product matrices (plus number of observations) for all the data information. The point estimates aren't that different from the two-step Swamy procedure as shown, but the standard errors are quite a bit higher, as Gibbs sampling takes into account that the Δ and σ_i^2 aren't known.

Linear Regression - Estimation by Swamy Random Coefficients				
Dependent Variable INVEST				
Panel(20) of Annual Data From 1//1935:01 To 10//1954:01				
Usable Observations 200				
Degrees of Freedom 197				
Mean of Dependent Variable 145.95825000				
Std Error of Dependent Variable 216.87529623				
Standard Error of Estimate 105.53917534				
Sum of Squared Residuals 2194287.9538				
Durbin-Watson Statistic 0.3258				
Variable	Coeff	Std Error	T-Stat	Signif

1. Constant	-9.62928514	17.03503951	-0.56526	0.57189459
2. VALUE	0.08458734	0.01995591	4.23871	0.00002248
3. CAP	0.19941840	0.05265336	3.78738	0.00015224

Linear Regression - Estimation by Random Coefficients by Gibbs Sampling				
Dependent Variable INVEST				
Panel(20) of Annual Data From 1//1935:01 To 10//1954:01				
Usable Observations 200				
Degrees of Freedom 197				
Mean of Dependent Variable 145.95825000				
Std Error of Dependent Variable 216.87529623				
Standard Error of Estimate 99.84185870				
Sum of Squared Residuals 1963774.1593				
Durbin-Watson Statistic 0.3485				
Variable	Coeff	Std Error	T-Stat	Signif

1. Constant	-9.530411979	7.894169198	-1.20727	0.22732729
2. VALUE	0.095587126	0.067696141	1.41200	0.15794920
3. CAP	0.198038916	0.096717379	2.04760	0.04059879

Example 12.1 Mean Group and Random Coefficients Models

This demonstrates estimation of mean group and random coefficients models using both two-step procedures and Gibbs sampling.

```
cal(panelobs=20) 1935
all 10//1954:1
open data grunfeld.xls
data(format=xls,org=cols)
compute nindiv=10
*
* Mean group done using procedure
*
@meangroup invest
# constant value cap
*
* Mean group done using SWEEP
*
sweep(variance=heterogeneous,group=%indiv(t),average=simple)
# invest
# constant value cap
*
linreg(create,coeffs=%beta,covmat=%xx,form=chisquared,$
  title="Pooled Mean Group using SWEEP") invest
# constant value cap
*
* Swamy GLS random coefficients
*
@swamy invest
# constant value cap
*
* Gibbs sampling for random coefficients model. The Swamy procedure uses
* an empirical estimate for the variance of the idiosyncratic component.
*
compute nburn=1000
compute ndraws=5000
*
* Initialize hdelta as near-zeros (which means almost no shrinkage
* towards the common mean).
*
dec symm hdelta(%nreg,%nreg) delta(%nreg,%nreg) ddelta(%nreg,%nreg)
compute hdelta=%diag(%fill(%nreg,1,.0001))
*
* Prior for DELTA (very low, but not zero degrees of freedom).
*
compute nudelta=0.1
compute ddelta =%diag(%fill(%nreg,1,1.0))
*
* Hierarchical prior for the individual variances.
*
compute omega =3.0
*
* Initialize with results from pooled regression.
```



```

*
dec vect[vect] betas(nindiv)
dec vect      sigmas(nindiv)
*
linreg(define=investeq) invest
# constant value cap
compute bdraw=%beta
ewise sigmas(i)=%seesq
compute sigsq=%seesq
*
dec series[vect] bgibbs
gset bgibbs 1 ndraws = %zeros(%nreg,1)
*
* Compute the cross product matrices of (x,y) for each individual
*
dec vect[symm] cmom(nindiv)
dec vect inobs(nindiv) irss(nindiv)
do i=1,nindiv
    cmom(smpl=%indiv(t)==i,equation=investeq)
    compute cmom(i)=%cmom
    compute inobs(i)=%nobs
end do i
*
infobox(action=define,lower=-nburn,upper=ndraws,progress) "Gibbs Sampling"
do draw=-nburn,ndraws
    *
    * Draw beta(i)'s given beta
    *
    do i=1,nindiv
        compute betas(i)=%ranmvpostcmom(cmom(i),1.0/sigmas(i),$
            hdelta,bdraw)
        compute irss(i)=%rsscmom(cmom(i),betas(i))
        compute rssplus=omega*sigsq+irss(i)
        compute sigmas(i)=rssplus/%ranchisqr(omega+inobs(i))
    end do i
    *
    * Draw beta (called bdraw) given beta(i)'s
    *
    compute hbeta=%zeros(%nreg,1)
    compute hsum =%zeros(%nreg,%nreg)
    do i=1,nindiv
        compute hbeta=hbeta+hdelta*betas(i),hsum=hsum+hdelta
    end do i
    compute hsum=inv(hsum)
    compute bdraw=hsum*hbeta+%ranmvnormal(%decomp(hsum))
    *
    * Draw the common sigsq given the sums of squared individual
    * residuals and the current r(i) (ratio sigmas(i) to sigsq).
    *
    sstats 1 nindiv irss(t)*sigsq/sigmas(i)>>rsscommon inobs(t)>>tnobs
    compute sigsq=rsscommon/%ranchisqr(tnobs)
    *
    * Draw the inverse of delta
    *

```

```

compute delta=nudelta*ddelta
do i=1,nindiv
  compute delta=delta+%outerxx(betas(i)-bdraw)
end do i
*
compute hdelta=%ranwishartf(%decomp(inv(delta)),nindiv+nudelta)
infobox(current=draw)
if draw<=0
  next
compute bgibbs(draw)=bdraw
end do draw
infobox(action=remove)
*
@mcmcpstproc(means=bmeans,cv=bcv,ndraws=ndraws) bgibbs
*
linreg(create,coeffs=bmeans,covmat=bcv,form=chisquared,$
  title="Random Coefficients by Gibbs Sampling") invest
# constant value cap

```

Vector Autoregressions

A Vector Autoregression is a close relative of the dynamic panel model examined in Chapter 7. And, as with those models, issues arise when a VAR is applied to panel data with a time dimension substantially smaller than would typically be used in standard time series analysis.

Before we talk about that, we need to first determine if a standard VAR is the appropriate choice for work with panel data. After all, a VAR isn't a structural model—it's a reduced form. As such, what do we think will be homogeneous and what will be heterogeneous? The dynamics of a VAR system are complicated functions of the full set of parameters. If the lag parameters are homogeneous, the responses to shocks will be identical for each cross section. Is that too strong an assumption? Perhaps they can be assumed to be similar, but not identical. That would require a more complicated estimation process, such as those in Chapter 12. We'll look at those in Sections 13.2 and 13.3. In a large N -small T data set, there is one other possibility, which is lag systems which are homogeneous across individuals, but differ across time periods.

One thing to note is that many panel VAR techniques ignore the bias problem that was the emphasis in Chapter 7. Any method designed to apply to data sets which have a reasonably large T dimension will have a negligible bias problem.

13.1 Instrumental Variables Estimators

Holtz-Eakin et al. (1988) describe several estimators for VAR's on large N -small T data with heterogeneous intercepts. The lag parameters are homogeneous across individuals, but they allow for them to be heterogeneous across time. The first step is again to difference the data to eliminate the individual effects. Now, however, it's not just the dependent variable of the equation that requires instrumenting, but *all* the endogenous variables in the system. The sets of instruments can be generated using the `@ABLAGS` procedure applied separately to each endogenous variable.¹

Among the estimators described are equation by equation instrumental variables. This is effectively the same as Arellano-Bond, just with more variables. In addition, with a VAR, it's possible to do systems instrumental variables estimation. The one thing to watch for there is the rank of the GMM weight

¹This paper pre-dated Arellano & Bond (1991).

matrix. As with the Arellano-Bond estimation procedure, you can allow for general serial correlation in the weight matrix by using the `LWINDOW=PANEL` option. However, the size of the weight matrix is number of equations \times number of instruments. That can very quickly easily exceed N , which is the maximum rank of a matrix computed with `LWINDOW=PANEL`, particularly since the number of potential instruments also goes up with the number of equations. Even before the weight matrix becomes singular, its inverse becomes unstable creating problems with the estimates. As a result, panel VAR's are usually estimated using a much smaller set of lag instruments than would be typical in the Arellano-Bond estimator.

The data set used in Example 13.1 was provided by Holtz-Eakin, though it's not clear that it was ever used in a published paper. It's derived from a data set used in the companion piece Holtz-Eakin et al. (1989), which studies fiscal behavior of a set of US cities. This has data for 161 municipalities over the period 1973-1980. The endogenous variables are their full- and part-time workforces (per capita) (variables `FULL` and `PART`), conditional on current and lagged wage rates for full- and part-time workers (`FWAGE` and `PWAGE`). One lag is used for all the variables. The wage rates are assumed to be pre-determined, but not exogenous. After first differencing to eliminate (the rather obvious) individual effects, the second and higher lags of the two endogenous variables are available as instruments, and the first and higher lags are available for the wage rates. The model also includes time period dummies. This reads the data and does the differences of the variables:

```
open data labor.xls
calendar(panelobs=8) 1973
data(format=xls,org=columns) 1//1973:01 161//1980:01 govid year $
  ftpay full ptpay part pay employ wage pwage fwage pop $
  me ma nh nj ny ri tn va grant $
  own comp debtey assetey fcomp tcomp netass null capital
*
set dpart      = part-part{1}
set dpwage     = pwage-pwage{1}
set dfwage     = fwage-fwage{1}
set dfull      = full-full{1}
```

This generates the time period dummies for the six periods per individual that can be included. We lose one due to differencing and one due to the lag in the model.

```
dec vect[series] pdummy(6)
do i=1,6
  set pdummy(i) = %period(t)==i+2
end do i
```

The instruments are created with:

```
@ablags(minlag=2,maxlag=2) full abfull
@ablags(minlag=2,maxlag=2) part abpart
@ablags(minlag=1,maxlag=2) fwage abfwage 3 *
@ablags(minlag=1,maxlag=2) pwage abpwage 3 *
instruments pdummy abfull abpart abfwage abpwage
```

This is a fairly minimal set of instruments, but for the full systems GMM estimator, you can't really go higher than this. The 3 * on the two pre-determined variables prevents them from generating a redundant instrument for time period 2. Since those are generated with `MINLAG=1`, instruments for time period 2 *can* be created, but those won't enter the regression since it is limited to observations 3 and up.

The simplest estimator is 2SLS without allowance for the serial correlation in the residuals. This defines equations which will be used in the systems estimators:

```
linreg(inst,define=parteq) dpart
# pdummy dfwage{0} dpwage{0} dfull{1} dpart{1} dfwage{1} dpwage{1}
linreg(inst,define=fulleq) dfull
# pdummy dfwage{0} dpwage{0} dfull{1} dpart{1} dfwage{1} dpwage{1}
```

This does equation by equation GMM, allowing for general serial correlation within each individual. Since first differencing should induce rather substantial negative serial correlation in the residuals, this should be more efficient than simple 2SLS:

```
linreg(inst,equation=parteq,optimal,lwindow=panel)
linreg(inst,equation=fulleq,optimal,lwindow=panel)
```

This does systems estimation, with weight matrix computed allowing only for heteroscedasticity:

```
group pvar parteq fulleq
sur(model=pvar,inst,zudep,update=continuous)
```

And this does systems estimation allowing for general serial correlation.

```
sur(model=pvar,inst,zudep,update=continuous,lwindow=panel)
```

This will be more efficient than the simpler systems estimator. However, this is where you need to be careful about running out of observations to compute the weight matrix. With the set of instruments that we are using (which again is fairly minimal), this requires a weight matrix of size 84, computed with 161 cross section observations. If the instruments weren't created with so many orthogonal to each other, that would probably be too many for reliable GMM estimates. In this situation, that's probably close to the maximum that you would want to use. This is the header output from the final systems estimator:

Linear Systems - Estimation by System Instrumental Variables			
Iterations Taken		2	
Panel(8) of Annual Data From	1//1974:01 To	161//1980:01	
Usable Observations		966	
Skipped/Missing (from 1287)		321	
J-Specification(60)		75.0153	
Significance Level of J		0.0917035	

If you don't want to add up the number of orthogonality conditions, you can back this out by adding the degrees of freedom of the J statistic (60) to the number of estimated coefficients (24).

13.2 Shrinkage Estimators, Univariate Autoregressions

The VECM models of Chapter 10 are a special case of (one equation out of) a VAR. The Pesaran-Shin-Smith and Mark-Sul estimators both assumed homogeneity in the cointegrating vector with heterogeneity in all remaining parameters. And in those and the Pedroni estimation procedures, the short-run parameters were considered to be “nuisance” parameters. We can't do that if we're interested in shorter-run behavior, such as for forecasting. However, assuming full homogeneity across individuals, as is done in section 13.1, is likely to produce poor results—different countries have different monetary and fiscal policies and thus will very likely have different short-run dynamics. However, it's likely that countries with somewhat similar economic structures (for instance, the G7 countries) will have at least similar dynamics.

An obvious way to handle this is with an extension of the shrinkage estimators from Chapter 12 to VAR'S. However, the Swamy estimator with unknown Δ won't be feasible in practice. As pointed out there, if the number of coefficients in the regression exceeds N , the covariance matrix of the changes is singular. For a VAR, the number of coefficients covered by this is the total number in the system across equations—typically *many* times larger than the number of individuals. If M is the number of variables and L the number of lags on each, the total size of the stacked system is M^2L . Even restricting the calculation to the diagonals of Δ may not work well since N is often not large enough even for *those* numbers to be computed well. However, unlike the typical regression equation, we generally have fairly good information about the relative sizes for the lag coefficients in a VAR, so we can use that to treat Δ as known.

To start, let's look at a shrinkage estimator for the simpler case of a univariate autoregression. This is Example 13.2. The data set is from Canova & Ciccarelli (2009). This has quarterly data on the G7 countries for 1980:1 to 2000:4. For the univariate case, we will work with real GDP growth. Univariate autoregressions are fairly easy to handle because their lag coefficients are independent of the scale of the variable—the coefficients don't change if you rescale the data. The intercept and variance will change but the lag coefficients won't. For seasonally adjusted data, we would expect that the most important coefficients will be the first few, and that no coefficient is likely to be much above 1. Based

upon the literature on Bayesian VAR'S, a prior on the difference between β_i and the common β with a standard deviation on the order of .1 or .2 seems reasonable. We'll do this without a further shrinkage with increasing lag, though that would be easy to incorporate.

This is similar to the Gibbs sampling part of Example 12.1 except that `HDELTA` will be set once outside the loop (rather than being treated as a parameter) and we will also not bother with the hierarchical prior for σ^2 since we have a large enough T dimension. The data are read with

```
compute nindiv=7          ;*Number of individuals
compute nlags=4          ;*Number of lags
*
dec vect[strings] country(nindiv)
input country
United States
Japan
Germany
United Kingdom
France
Italy
Canada
*
cal(q,panel=84) 1980
*
open data g7panel.xls
data(org=columns,format=xls) / inf infr eg gdpq
```

The series of interest for now is `GDPG`. The `HDELTA` (precision version of Δ) is set up with

```
compute tight=.1
*
dec symm hdelta(nlags+1,nlags+1)
ewise hdelta(i,j)=%if(i==j.and.i>1,1.0/tight^2,0.0)
```

This has a “flat” prior on the intercept. The value of `TIGHT` is the standard deviation, so the precision is its reciprocal squared. As in the previous chapter, we'll initialize based upon the pooled estimate:

```
dec vect[vect] betas(nindiv)
dec vect      sigmas(nindiv)
*
linreg(define=areq) gdpq
# constant gdpq{1 to nlags}
*
compute bdraw=%beta
ewise sigmas(i)=%seesq
```

For efficiency, we compute individual-by-individual cross product matrices and save those and the number of observations.

```
dec vect[symm] cmom(nindiv)
dec vect inobs(nindiv) irss(nindiv)
do i=1,nindiv
  cmom(smpl=%indiv(t)==i,equation=areq)
  compute cmom(i)=%cmom
  compute inobs(i)=%nobs
  compute betas(i)=bdraw
end do i
```

The following sets up a `PARMSET` to organize the coefficients that we want to save. That will be the common mean coefficient vector and the individual coefficient vectors.

```
nonlin(parmset=gibbsparms) bdraw betas
dec series[vect] bgibbs
gset bgibbs 1 ndraws = %zeros(%size(%parmspeek(gibbsparms)),1)
```

Inside the draw loop, we first draw the individual coefficients given the common coefficient vector and a new value of σ_i^2 . Because we aren't doing a hierarchical prior for the latter, this doesn't weight in a value of σ^2 .

```
do i=1,nindiv
  compute betas(i)=%ranmvpostcmom(cmom(i),1.0/sigmas(i), $
    hdelta,bdraw)
  compute irss(i)=%rsscmom(cmom(i),betas(i))
  compute rssplus=irss(i)
  compute sigmas(i)=rssplus/%ranchisqr(inobs(i))
end do i
```

The draw for the mean coefficient vector is again quite similar to the previous chapter. The one difference is that `%GINV` (generalized inverse) is used rather than `INV` to “invert” `HSUM`. Because we used a non-informative prior on the intercept, `HSUM` will have all zeros in the first row and column and thus won't be invertible. Given the structure of the model, we don't really have any good information on the intercept in the mean vector—as this is computed, it will always be zero. This isn't a problem because it also doesn't have any effect on the draws for β_i since it gets multiplied by its precision of zero in computing the mean for that.

```
compute hbeta=%zeros(%nreg,1)
compute hsum=%zeros(%nreg,%nreg)
do i=1,nindiv
  compute hbeta=hbeta+hdelta*betas(i),hsum=hsum+hdelta
end do i
compute hsum=%ginv(hsum)
compute bdraw=hsum*hbeta+%ranmvnormal(%decomp(hsum))
```


Since we don't need a draw for `HDELTA`, we are done with a Gibbs sweep. We just need the end-of-pass bookkeeping:

```
infobox(current=draw)
if draw<=0
  next
compute bgibbs(draw)=%parmspeek(gibbsparms)
```

We now have `NDRAWS` simulations from the posterior for the coefficients. What can we do with this? The following will compute the means of those into `BMEANS`.

```
@mcmcpostproc(means=bmeans, cv=bcv, ndraws=ndraws) bgibbs
```

The following will do comparisons of the responses to unit shocks at the mean estimates for each country. It resets the coefficients for the equation to the proper subvector of the vector of Gibbs means, shocks the equation and saves them into a different slot in the `GDPPIRF` vector of series.

```
dec vect[series] gdpirf(nindiv)
dec vect[series] gdpirfi(nindiv)
do i=1,nindiv
  compute coeffsi=%xsubvec(bmeans,i*%nreg+1,(i+1)*%nreg)
  compute %eqnsetcoeffs(areq,coeffsi)
  impulse(shock=||1.0||,steps=24)
  # areq gdpirf(i)
end do i
list iser = 1 to nindiv
graph(number=0,nodates,klabls=country,key=below,$
  footer="Shrinkage IRF's") nindiv
cards gdpirf(iser)
```

This produces (subject to slight Monte Carlo variation) Figure 13.1.

The responses are broadly similar, with the response in Germany decaying quickest and the UK slowest. If you make the value of `TIGHT` smaller, this spread will get tighter as well; if you make it larger, this will expand a bit. However, there is relatively little difference for values in the standard “Minnesota prior” range of .1 to .2.

We can also compare the dynamics from country-by-country OLS estimates with the shrinkage estimates:

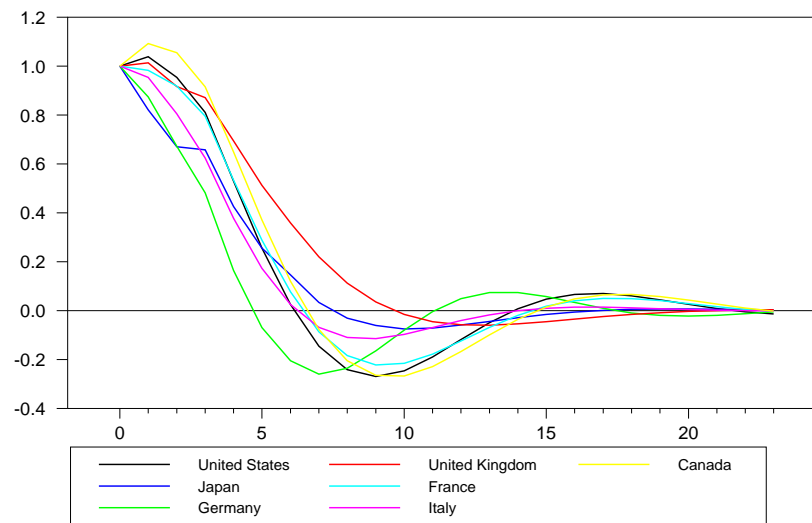


Figure 13.1: Univariate IRF Comparison

```
do i=1,nindiv
  linreg(noprint,equation=areq,smpl=%indiv(t)==i)
  impulse(shock=||1.0||,steps=24)
  # areq gdpirfi(i)
  graph(nodates,number=0,klabels=||"Individual","Shrinkage"||,$
    footer="Individual vs Shrinkage IRF's-"+country(i)) 2
  # gdpirfi(i)
  # gdpirf(i)
end do i
```

Most of these are relatively similar between the two estimates. The one where the prior has the greatest effect is Japan: Figure 13.2. The OLS estimates show a rather odd pattern of a quick decline followed by an uptick in the third period out, a pattern not seen in any other country. The shrinkage estimator largely straightens that out, and that's true even with a looser .2 prior. That would seem to indicate the particular shape really isn't that important for fitting the Japanese data.

13.3 Shrinkage Estimators, VAR's

We now look at how we extend the shrinkage estimator to the full VAR. The choice for Δ isn't as simple now because the coefficients are now scale-dependent—if you change the scale of one of the variables, but not the others, the cross effects coefficients will rescale. Again, the use of a Minnesota prior for the spread among the individuals seems reasonable.

To adjust for scale, the Minnesota prior starts with an (OLS) univariate autoregression (including any deterministic components) on each of the dependent variables. The standard errors of estimate of these regressions (called s_i for

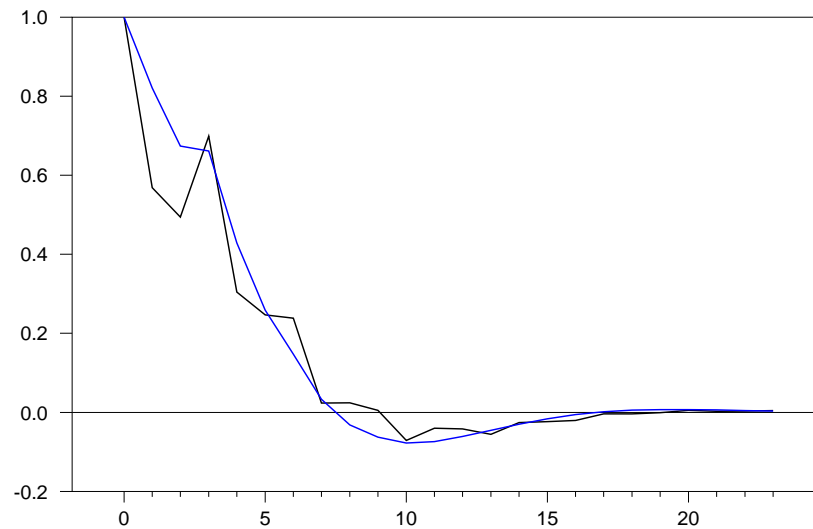


Figure 13.2: Individual vs Shrinkage IRF's-Japan

equation i) are used to rescale the standard deviation of the prior for a coefficient on a lag of variable j in equation i by s_i/s_j . While somewhat arbitrary, this has stood the test of time. It handles issues of scale correctly (you'll get exactly the same results if you rescale variables), it doesn't add any extra hyperparameters to the structure and it seems to work fine in practice. In the RATS manual, the standard error function takes the general form:

$$S(i, j, l) = \frac{\gamma l^{-d} f(i, j) s_i}{s_j}$$

$f(i, j)$ is the relative tightness of variable j in equation i . This could be a general $M \times M$ function, but the standard here is to make

$$f(i, i) = 1; f(i, j) = w \text{ if } i \neq j$$

where w is a new hyperparameter.

Since the s_i are computed from the data, we need to specify the overall tightness γ , the lag decay parameter d and the "other" relative tightness w . For **Example 13.3**, we will be using the same data set, including in the VAR the series INF (CPI inflation), INFR (rental inflation), EG (employment growth) as well as GDPG (GDP growth). The VAR is set up with

```
system(model=varmodel)
variables inf infr eg gdp
lags 1 to nlags
det constant
end(system)
```

The settings for the hyperparameters governing Δ are fairly standard from BVAR literature, except we will be using them for the spread among the coefficients, rather than deviations from a fixed mean:

```

compute tight =.1
compute other =.5
compute decay =.5

```

The following computes the s_i :

```

compute dvlist=%modeldepvars(varmodel)
compute nvar  =%size(dvlist)
dec vect see(nvar)
do j=1,nvar
  linreg(noprint) dvlist(j)
  # constant dvlist(j){1 to nlags}
  compute see(j)=sqrt(%seesq)
end do j

```

This sets up the prior with zero precision for any coefficient which is not a lag of one of the endogenous variables. HDIAG is a vector of diagonal elements across the entire model, which is converted into the precision matrix.

```

compute dvtable =%tablefromr1(dvlist)
compute sampleeq=%modeleqn(varmodel,1)
compute nsize  =%eqnsize(sampleeq)
compute etable =%eqntable(sampleeq)
dec vect hdiag(nsize*nvar)
do j=1,nvar
  compute jvar=dvlist(j)
  do i=1,nsize
    compute ivar =etable(1,i)
    compute ilag =etable(2,i)
    compute ipos =%tablefind(dvtable,ivar,0)
    compute slot =(j-1)*nsize+i
    if ipos==0
      compute hdiag(slot)=0.0
    else
      compute hdiag(slot)=ilag^(2*decay)*%if(ivar==jvar,$
        1.0/tight,see(ipos)/(tight*other*see(j)))^2
    end do i
  end do j
compute [symm] hdelta=%diag(hdiag)

```

We will again start with pooled least squares estimates of the regressions. Now, however, the SIGMAS are SYMMETRIC matrices rather than simple real numbers. The BDRAW and BETAS elements will be system coefficients stacked into a vector.

```

dec vect[vect] betas(nindiv)
dec vect[symm] sigmas(nindiv)
*
estimate
*
compute bdraw=%betasys
ewise sigmas(i)=%sigma

```

We again need the cross product matrix for each individual. When this is done using a MODEL as input, it includes both the explanatory variables (which come first in the structure of the matrix) and all the endogenous variables.

```

dec vect[symm] cmom(nindiv)
dec vect inobs(nindiv)
do i=1,nindiv
  cmom(smpl=%indiv(t)==i,model=varmodel)
  compute cmom(i)=%cmom
  compute inobs(i)=%nobs
  compute betas(i)=bdraw
end do i

```

The setup for saving the results is almost identical:

```

compute varsize=%size(bdraw)
nonlin(parmset=gibbsparms) bdraw betas
dec series[vect] bgibbs
gset bgibbs 1 ndraws = %zeros(%size(%parmspeek(gibbsparms)),1)

```

Inside the draw loop is similar to before, but with the use of the functions for draws from systems of equations. First, drawing the individual coefficients given the common value:

```

do i=1,nindiv
  compute betas(i)=%ranmvkroncmom(cmom(i),inv(sigmas(i)),$
    hdelta,bdraw)
  compute rssmat=%sigmacmom(%cmom,%reshape(betas(i),nsize,nvar))
  compute sigmas(i)=%ranwisharti(%decomp(inv(rssmat)),inobs(i))
end do i

```

The draw for the common BETA is effectively the same, other than the size of the matrices:

```

compute hbeta=%zeros(varsize,1)
compute hsum=%zeros(varsize,varsize)
do i=1,nindiv
  compute hbeta=hbeta+hdelta*betas(i),hsum=hsum+hdelta
end do i
compute hsum=%ginv(hsum)
compute bdraw=hsum*hbeta+%ranmvnormal(%decomp(hsum))

```

IRF graphs are trickier than they are for the univariate processes. Different countries have different patterns of contemporaneous correlations, so picking one standard Σ matrix isn't possible. However, using a different one for each country won't make the IRF's comparable. Here, this is done by doing unit shocks to all variables, although that also has the problem that independent shocks are unrealistic. However, that's the best we can do for comparing the behavior of the lag coefficients alone.

```

dec vect[rect[series]] irf(nindiv)
do i=1,nindiv
  compute coeffsi=%xsubvec(bmeans,i*varsize+1,(i+1)*varsize)
  compute %modelsetcoeffs(varmodel,coeffsi)
  impulse(model=varmodel,results=irf(i),$
    cv=%identity(nvar),steps=24)
end do i
list iser = 1 to nindiv
spgraph(vfields=nvar,hfields=nvar,footer="Shrinkage IRF's",
  xlabel=longvars,ylabel=longvars)
do i=1,nvar
  do j=1,nvar
    graph(number=0,nodates) nindiv
    cards irf(iser)(i,j)
  end do j
end do i
spgraph(done)

```

This generates Figure 13.3. The variable shocked is in the column and the target is in the row. You can see that the patterns are largely consistent across countries except for anything involving the rental inflation variable.

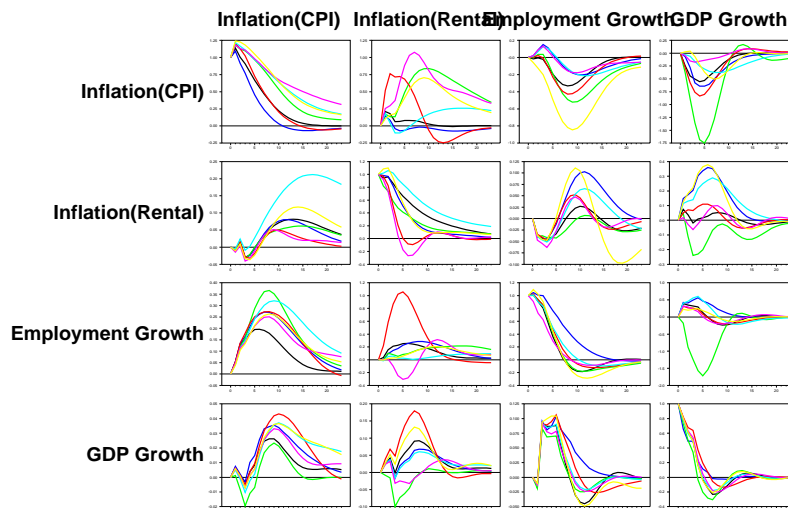


Figure 13.3: Shrinkage IRF's-VAR

13.4 Causality Tests

A Granger causality test is, in a bivariate setting, simply a restriction on the coefficients in one equation in a VAR. In the tightly parameterized world of Section 13.1, this is straightforward: estimate the unrestricted model and do a Wald test on the lags. This possibility is mentioned in the original paper.

More interesting is the application to the small N -big T data sets, where homogeneity is both unreasonable and unnecessary. As has been the case from the beginning of the workbook, the testing strategy will depend upon what we see as being homogeneous and what is heterogeneous. For a VAR on macroeconomic data, everything being heterogeneous would seem to be the right choice. With heterogeneity under the null and the alternative, the test would reject (theoretically) if there is Granger causality in *any* of the individual regressions, that is, the null is that there is no Granger causality in any individual, while the alternative is that there is Granger causality in at least one. Of course, the power of the test will be fairly low if there is only causality in a few individuals.

Allowing for full heterogeneity (plus independence across individuals), a likelihood ratio test is quite easy to compute. It is, in fact, just the sum of the individual likelihood ratios for Granger causality (with degrees of freedom summing as well). Since rejection of the null of non-causality means that causality is found in some (though not necessarily all) of the individuals, an obvious way to display the results is with both the joint test and the individual tests. Note, by the way, that it is possible for all the individual tests to be insignificant at conventional significance levels while the joint test is strongly significant. That's not unexpected, since the whole point of the joint test is to give us more and better information than is available in the individual samples.

Example 13.4 uses the data set from Mark and Sul used in Example 10.2. We do a test of real money causing real income, with both variables in log differences. Use of panel data doesn't change the fact that causality tests are subject to the problems described in Sims et al. (1990). If the series have unit roots, a causality hypothesis is one of the types of tests that has a non-standard distribution since non-causality implies non-cointegration.² By first differencing, we eliminate the unit roots. Of course, if the two series are cointegrated, the first-differenced specification is misspecified, which is a different issue. We'll assume here that this is a reasonable situation for causality testing.

You can do a joint test directly with

²The simplest summary of the results of the paper are that a restriction which does not alter the potential unit root/cointegration behavior of the data has standard asymptotics, and a restriction which *does* alter it has non-standard asymptotics.

```

sweep(group=%indiv(t),var=hetero)
# dy
# constant dy{1 to p} dm{1 to p}
compute loglunr=%logl,nregunr=%nregsystem
sweep(group=%indiv(t),var=hetero)
# dy
# constant dy{1 to p}
compute loglres=%logl,nregres=%nregsystem
cdf(title="Heterogeneous Panel Causality Test") chisqr $
    2.0*(loglunr-loglres) nregunr-nregres
compute jointtest=%cdstat,jointsignif=%signif

```

This uses **SWEEP** to do unrestricted, then restricted regressions, in each case saving the log likelihood and total coefficients. The likelihood ratio test is constructed using those.

The individual likelihood ratio tests can be done with (for individual i)

```

linreg(noprint,smpl=%indiv(t)==i) dy
# constant dy{1 to p} dm{1 to p}
exclude(noprint)
# dm{1 to p}
compute lr=log((1+p*%cdstat/%ndf))*%nobs

```

EXCLUDE produces an F -test statistic (%CDSTAT), which we convert into a likelihood ratio test to be compatible with the joint test. These are inserted into a **REPORT** along with the overall result.

The output from the original joint test is

Heterogeneous Panel Causality Test	
Chi-Squared(57)=	202.357431 with Significance Level 0.00000000

which obviously shows overwhelming rejection. The individual tests are in Table 13.1. As you can see from that, the strongest rejections tend to be in the smaller countries. However, even if the test were restricted to get six G7 countries included (Italy isn't in the data set), it would be 36.369 with 18 degrees of freedom which is still very significant (.006 significance level) even though none of those six countries has an individual significance below France's .028.

Table 13.1: Money-Income Causality Tests

Austria	6.142	0.10488
Belgium	16.859	0.00076
Denmark	8.136	0.04327
Finland	13.771	0.00323
France	9.121	0.02772
Germany	1.806	0.61368
Iceland	35.762	0.00000
Ireland	12.283	0.00647
Netherlands	3.896	0.27290
Norway	23.017	0.00004
Portugal	6.063	0.10858
Spain	7.043	0.07055
Switzerland	7.288	0.06325
UK	7.980	0.04642
Japan	7.263	0.06396
Australia	11.205	0.01067
New Zealand	14.521	0.00227
Canada	8.234	0.04142
USA	1.965	0.57966
OVERALL	202.357	0.00000

Example 13.1 Panel VAR with instrumental variables

This estimates a VAR for full- and part-time employment using the techniques from Holtz-Eakin et al. (1988). The details begin on page 142.

```

open data labor.xls
calendar(panelobs=8) 1973
data(format=xls,org=columns) 1//1973:01 161//1980:01 govid year $
  ftpay full ptpay part pay employ wage pwage fwage pop $
  me ma nh nj ny ri tn va grant $
  own comp debtey assetey fcomp tcomp netass null capital
*
set dpart      = part-part{1}
set dpwage     = pwage-pwage{1}
set dfwage     = fwage-fwage{1}
set dfull      = full-full{1}
*
* Generate time period dummies. We need only for the last 6 periods,
* since the first two are lost in the transformations.
*
dec vect[series] pdummy(6)
do i=1,6
  set pdummy(i) = %period(t)==i+2
end do i
*
* Generate Arellano-Bond instruments
*
@ablags(minlag=2,maxlag=2) full  abfull
@ablags(minlag=2,maxlag=2) part  abpart
@ablags(minlag=1,maxlag=2) fwage abfwage 3 *
```

```

@ablags(minlag=1,maxlag=2) pwage abpwage 3 *
instruments pdummy abfull abpart abfwage abpwage
*
* Simple 2SLS
*
linreg(inst,define=parteq) dpart
# pdummy dfwage{0} dpwage{0} dfull{1} dpart{1} dfwage{1} dpwage{1}
linreg(inst,define=fulleq) dfull
# pdummy dfwage{0} dpwage{0} dfull{1} dpart{1} dfwage{1} dpwage{1}
*
* Equation by equation GMM
*
linreg(inst,equation=parteq,optimal,lwindow=panel)
linreg(inst,equation=fulleq,optimal,lwindow=panel)
*
* 3SLS
*
group pvar parteq fulleq
sur(model=pvar,inst,zudep,update=continuous)
*
* 3SLS with GMM weight matrix
*
sur(model=pvar,inst,zudep,update=continuous,lwindow=panel)

```

Example 13.2 Univariate Autoregression: Shrinkage Estimator

This estimates a univariate autoregression using a shrinkage estimator, allowing for different, but similar, autoregressive dynamics. See the discussion beginning on page 144 for details.

```

compute nindiv=7          ;*Number of individuals
compute nlags=4           ;*Number of lags
*
dec vect[strings] country(nindiv)
input country
United States
Japan
Germany
United Kingdom
France
Italy
Canada
*
cal(q,panel=84) 1980
*
open data g7panel.xls
data(org=columns,format=xls) / inf infr eg gdpq
*
* Shrink towards common mean. The precision is treated as known.
*
compute tight=.1

```

```

*
dec symm hdelta(nlags+1,nlags+1)
ewise hdelta(i,j)=%if(i==j.and.i>1,1.0/tight^2,0.0)
*
* Gibbs sampling for random coefficients model.
*
compute nburn=1000
compute ndraws=5000
*
* Initialize with results from pooled regression.
*
dec vect[vect] betas(nindiv)
dec vect      sigmas(nindiv)
*
linreg(define=areq) gdpq
# constant gdpq{1 to nlags}
*
compute bdraw=%beta
ewise sigmas(i)=%seesq
*
dec vect[symm] cmom(nindiv)
dec vect inobs(nindiv) irss(nindiv)
do i=1,nindiv
  cmom(smpl=%indiv(t)==i,equation=areq)
  compute cmom(i)=%cmom
  compute inobs(i)=%nobs
  compute betas(i)=bdraw
end do i
*
nonlin(parmset=gibbsparms) bdraw betas
dec series[vect] bgibbs
gset bgibbs 1 ndraws = %zeros(%size(%parmspeek(gibbsparms)),1)
*
infobox(action=define,lower=-nburn,upper=ndraws,progress) "Gibbs Sampling"
do draw=-nburn,ndraws
  *
  * Draw beta(i)'s given beta
  *
  do i=1,nindiv
    compute betas(i)=%ranmvpostcmom(cmom(i),1.0/sigmas(i),$
      hdelta,bdraw)
    compute irss(i)=%rsscmom(cmom(i),betas(i))
    compute rssplus=irss(i)
    compute sigmas(i)=rssplus/%ranchisqr(inobs(i))
  end do i
  *
  * Draw beta (called bdraw) given beta(i)'s
  *
  compute hbeta=%zeros(%nreg,1)
  compute hsum=%zeros(%nreg,%nreg)
  do i=1,nindiv
    compute hbeta=hbeta+hdelta*betas(i),hsum=hsum+hdelta
  end do i
  compute hsum=%ginv(hsum)

```

```

        compute bdraw=hsum*hbeta+%ranmvnormal(%decomp(hsum))
        infobox(current=draw)
        if draw<=0
            next
        compute bgibbs(draw)=%parmspeek(gibbsparms)
    end do draw
    infobox(action=remove)
    *
    @mcmcpostproc(means=bmeans, cv=bcv, ndraws=ndraws) bgibbs
    *
    dec vect[series] gdpirf(nindiv)
    dec vect[series] gdpirfi(nindiv)
    do i=1,nindiv
        compute coeffsi=%xsubvec(bmeans,i*%nreg+1,(i+1)*%nreg)
        compute %eqnsetcoeffs(areq,coeffsi)
        impulse(shock=||1.0||, steps=24)
        # areq gdpirf(i)
    end do i
    list iser = 1 to nindiv
    graph(number=0, nodates, klabels=country, key=below, $
        footer="Shrinkage IRF's") nindiv
    cards gdpirf(iser)
    *
    do i=1,nindiv
        linreg(noprint, equation=areq, smpl=%indiv(t)==i)
        impulse(shock=||1.0||, steps=24)
        # areq gdpirfi(i)
        graph(nodates, number=0, klabels=||"Individual", "Shrinkage"||, $
            footer="Individual vs Shrinkage IRF's-"+country(i)) 2
        # gdpirfi(i)
        # gdpirf(i)
    end do i

```

Example 13.3 Vector Autoregression: Shrinkage Estimator

This estimates an autoregression using a shrinkage estimator, allowing for different, but similar, dynamics. The details start on page 148.

```

compute nindiv=7                ;*Number of individuals
compute nlags=4                 ;*Number of lags
*
dec vect[strings] country(nindiv)
input country
United States
Japan
Germany
United Kingdom
France
Italy
Canada
*

```

```

cal(q,panel=84) 1980
*
open data g7panel.xls
data(org=columns,format=xls) / inf infr eg gdpq
dec vect[strings] longvars
compute longvars=|"Inflation(CPI)","Inflation(Rental)","$
    "Employment Growth","GDP Growth"||
*
* Set up the model
*
system(model=varmodel)
variables inf infr eg gdpq
lags 1 to nlags
det constant
end(system)
*
* Shrinkage based upon Minnesota prior. Same standard errors but no mean.
*
compute tight =.1
compute other =.5
compute decay =.5
*
* Compute the standard errors from OLS autoregression for use in scaling
* the prior.
*
compute dvlist=%modeldepvars(varmodel)
compute nvar  =%size(dvlist)
dec vect see(nvar)
do j=1,nvar
    linreg(noprint) dvlist(j)
    # constant dvlist(j){1 to nlags}
    compute see(j)=sqrt(%seesq)
end do j
*
* Set up the precision form for a Minnesota prior
*
compute dvtable =%tablefromrl(dvlist)
compute sampleeq=%modeleqn(varmodel,1)
compute nsize  =%eqnsize(sampleeq)
compute etable =%eqntable(sampleeq)
dec vect hdiag(nsize*nvar)
do j=1,nvar
    compute jvar=dvlist(j)
    do i=1,nsize
        compute ivar =etable(1,i)
        compute ilag =etable(2,i)
        compute ipos =%tablefind(dvtable,ivar,0)
        compute slot =(j-1)*nsize+i
        if ipos==0
            compute hdiag(slot)=0.0
        else
            compute hdiag(slot)=ilag^(2*decay)*%if(ivar==jvar,$
                1.0/tight,see(ipos)/(tight*other*see(j)))^2
        end do i
    end do j
end do i

```

```

end do j
*
compute [symm] hdelta=%diag(hdiag)
*
* Gibbs sampling for random coefficients model.
*
compute nburn =1000
compute ndraws=5000
*
* Initialize with results from pooled least squares
*
dec vect[vect] betas(nindiv)
dec vect[symm] sigmas(nindiv)
*
estimate
*
compute bdraw=%betasys
ewise sigmas(i)=%sigma
*
dec vect[symm] cmom(nindiv)
dec vect inobs(nindiv)
do i=1,nindiv
  cmom(smpl=%indiv(t)==i,model=varmodel)
  compute cmom(i)=%cmom
  compute inobs(i)=%nobs
  compute betas(i)=bdraw
end do i
*
compute varsize=%size(bdraw)
nonlin(parmset=gibbsparms) bdraw betas
dec series[vect] bgibbs
gset bgibbs 1 ndraws = %zeros(%size(%parmspeek(gibbsparms)),1)
*
infobox(action=define,lower=-nburn,upper=ndraws,progress) "Gibbs Sampling"
do draw=-nburn,ndraws
  *
  * Draw beta(i)'s given beta and sigma(i)'s given beta(i)
  *
  do i=1,nindiv
    compute betas(i)=%ranmvkroncmom(cmom(i),inv(sigmas(i)),$
      hdelta,bdraw)
    compute rssmat=%sigmacmom(%cmom,%reshape(betas(i),nsize,nvar))
    compute sigmas(i)=%ranwisharti(%decomp(inv(rssmat)),inobs(i))
  end do i
  *
  * Draw beta (called bdraw) given beta(i)'s
  *
  compute hbeta=%zeros(varsize,1)
  compute hsum=%zeros(varsize,varsize)
  do i=1,nindiv
    compute hbeta=hbeta+hdelta*betas(i),hsum=hsum+hdelta
  end do i
  compute hsum=%ginv(hsum)
  compute bdraw=hsum*hbeta+%ranmvnormal(%decomp(hsum))

```

```

    infobox(current=draw)
    if draw<=0
        next
    compute bgibbs(draw)=%parmspeek(gibbsparms)
end do draw
infobox(action=remove)
*
@mcmcpstproc(means=bmeans, cv=bcv, ndraws=ndraws) bgibbs
*
dec vect[rect[series]] irf(nindiv)
do i=1,nindiv
    compute coeffsi=%xsubvec(bmeans,i*varsize+1,(i+1)*varsize)
    compute %modelsetcoeffs(varmodel,coeffsi)
    impulse(model=varmodel,results=irf(i),$
        cv=%identity(nvar),steps=24)
end do i
list iser = 1 to nindiv
spgraph(vfields=nvar,hfields=nvar,footer="Shrinkage IRF's",
    xlabel=longvars,ylabel=longvars)
do i=1,nvar
    do j=1,nvar
        graph(number=0,nodates) nindiv
        cards irf(iser)(i,j)
    end do j
end do i
spgraph(done)

```

Example 13.4 Causality Test

This is an example of a causality test. The discussion is in Section 13.4.

```

cal(panelobs=40)
open data panelmoney.xls
data(org=obs,format=xls) 1//1 19//40 realm reayl
dec vect[strings] countries(19)
input countries
Austria
Belgium
Denmark
Finland
France
Germany
Iceland
Ireland
Netherlands
Norway
Portugal
Spain
Switzerland
UK
Japan

```

```

Australia
New Zealand
Canada
USA
*
* Number of lags
*
compute p=3
*
set dy = realy-realy{1}
set dm = realm-realm{1}
*
* Joint test.
*
sweep(group=%indiv(t),var=hetero)
# dy
# constant dy{1 to p} dm{1 to p}
compute loglunr=%logl,nregunr=%nregsystem
sweep(group=%indiv(t),var=hetero)
# dy
# constant dy{1 to p}
compute loglres=%logl,nregres=%nregsystem
cdf(title="Heterogeneous Panel Causality Test") chisqr $
  2.0*(loglunr-loglres) nregunr-nregres
compute jointtest=%cdstat,jointsignif=%signif
*
* Individual causality tests. The individual log likelihood ratios sum
* to the joint test.
*
report(action=define,title="Panel Causality Test")
do i=1,19
  linreg(noprint,smpl=%indiv(t)==i) dy
  # constant dy{1 to p} dm{1 to p}
  exclude(noprint)
  # dm{1 to p}
  compute lr=log((1+p*%cdstat/%ndf))*%nobs
  report(row=new,atcol=1) countries(i) lr %chisqr(lr,p)
end do i
report(row=new,atcol=1) "OVERALL" jointtest jointsignif
report(atcol=2,tocol=2,action=format,picture="*.###")
report(atcol=3,tocol=3,action=format,picture="*.#####")
report(action=show)

```


Random Effects Component Estimators

This works through the details for the most commonly used “method of moments” estimators for the variances of random components, allowing for one- and two-way effects with possibly unbalanced samples. There are N individuals and T time periods. The total number of data points is n , which will be NT for a balanced sample, but less than that for an unbalanced one. The number of time periods for individual i is denoted T_i , and the number of individuals for time period t is N_t .

We’ll also use the following notation: \mathcal{J} is the block diagonal matrix $\text{diag}(\mathbf{J}_{T_1}, \dots, \mathbf{J}_{T_N})$, which replaces each element $X_{i,t}$ of a data vector by the sum of X across individual i and $\bar{\mathcal{J}} = \text{diag}(\frac{1}{T_1}\mathbf{J}_{T_1}, \dots, \frac{1}{T_N}\mathbf{J}_{T_N})$ which averages rather than sums. The corresponding matrices for summing and averaging across individuals for a given time period are \mathcal{T} and $\bar{\mathcal{T}}$. \mathcal{T} will be an $n \times n$ matrix with 0’s everywhere except 1’s in the entries where the row and column represent the same time period.

We’ll use \mathbf{Q} (with different subscripts) for various residual projection matrices and similarly use \mathbf{P} for projection matrices themselves. The calculations here will make repeated use of the results:

- $\text{trace } \mathbf{AB} = \text{trace } \mathbf{BA}$ if both products make sense
- projection matrices are symmetric, idempotent ($\mathbf{AA} = \mathbf{A}$) and have trace equal to their rank.

A.1 Individual Effects Only

The model for the true residuals is

$$u_{it} = \mu_i + \eta_{it}$$

We’ll first compute these assuming that we can observe these, which means:

$$Eu_{it}^2 = \sigma_\mu^2 + \sigma_\eta^2$$

This was proposed in Wallace & Hussain (1969). Summing over the data set gives one equation in the two variances:

$$E \sum_{i,t} u_{it}^2 = \sigma_\mu^2 \left(\sum_i T_i \right) + \sigma_\eta^2 \left(\sum_i T_i \right) \quad (\text{A.1})$$

Almost any distinct (from the identity) quadratic form in the u vector can yield a second equation. What will be (in the end) the simplest is:

$$\begin{aligned}
 E \sum_i \frac{1}{T_i} \left(\sum_t u_{it} \right)^2 &= \sum_i \frac{1}{T_i} E \left(\sum_t (\mu_i + \eta_{it}) \right)^2 \\
 &= \sum_i \frac{1}{T_i} E \left(T_i \mu_i + \sum_t \eta_{it} \right)^2 \\
 &= \sigma_\mu^2 \left(\sum_i T_i \right) + \sigma_\eta^2 N
 \end{aligned} \tag{A.2}$$

Summing across t before squaring reduces the relative variance from the random part thus giving different multipliers. Replacing expectations with sample values and solving (A.1) and (A.2) simultaneously gives an estimate for the two components.

An approach which gives identical results uses the fact that the within variation eliminates the individual effect:

$$E(u_{it} - u_{i\bullet})^2 = E(\eta_{it} - \eta_{i\bullet})^2 = \sigma_\eta^2 \frac{(T_i - 1)}{T_i}$$

Summing across i and t gives the condition:

$$\sum_i \sum_t E(u_{it} - u_{i\bullet})^2 = \sum_i \sum_t \sigma_\eta^2 \frac{(T_i - 1)}{T_i} = \sigma_\eta^2 \sum_i (T_i - 1) \tag{A.3}$$

Solving any combination of two of the sample analogues of (A.1), (A.2) and (A.3) will give identical results since

$$\sum_t (u_{it} - u_{i\bullet})^2 = \sum_t u_{it}^2 - T_i u_{i\bullet}^2 = \sum_t u_{it}^2 - \frac{1}{T_i} \left(\sum_s u_{is} \right)^2$$

so summing across i connects the three by an identity.

A.1.1 Wallace-Hussain (OLS-based)

These calculations haven't taken into account the fact that the residuals have additional correlation due to sampling error. In general, the full sample covariance matrix of the least squares residuals takes the form:

$$\mathbf{Q}_X \mathbf{\Omega} \mathbf{Q}_X$$

where $\mathbf{Q}_X = \mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is the standard residual projection matrix and $\mathbf{\Omega}$ is the full sample covariance matrix of the (true) residuals. $\mathbf{\Omega}$ depends additively upon terms in the component variances: in the case of individual effects only,

it will be $\sigma_\eta^2 \mathbf{I} + \sigma_\mu^2 \mathcal{J}$.

$$\begin{aligned} E \sum_{i,t} \hat{u}_{it}^2 &= \text{trace}(\mathbf{Q}_X(\sigma_\eta^2 \mathbf{I} + \sigma_\mu^2 \mathcal{J})\mathbf{Q}_X) \\ &= \sigma_\eta^2 \text{trace}(\mathbf{Q}_X \mathbf{Q}_X) + \sigma_\mu^2 \text{trace}(\mathbf{Q}_X \mathcal{J} \mathbf{Q}_X) \end{aligned} \quad (\text{A.4})$$

By the well-known properties of \mathbf{Q}_X , the σ_η^2 term simplifies to

$$\sigma_\eta^2 \text{trace}(\mathbf{Q}_X) = \sigma_\eta^2 (\dim(\mathbf{Q}_X) - K)$$

where K is the number of regressors (including the constant). Thus the random component will use the standard degrees of freedom correction, regardless of the behavior of the \mathbf{X} matrix (as long as it has full column rank). The σ_μ^2 term does not simplify to that extent unless the regressors are all time-invariant and the sample is balanced. While the direct calculation based upon (A.4) will work, the matrices are often of a size that makes this infeasible. Instead, we can rearrange the calculation to get in general:

$$\begin{aligned} \sigma_\mu^2 \text{trace}(\mathbf{Q}_X \mathcal{J} \mathbf{Q}_X) &= \sigma_\mu^2 \text{trace}(\mathcal{J} \mathbf{Q}_X \mathbf{Q}_X) \\ &= \sigma_\mu^2 \text{trace}\left(\mathcal{J} \left(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)\right) \\ &= \sigma_\mu^2 \left\{ \text{trace}(\mathcal{J}) - \text{trace}\left(\mathcal{J}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right) \right\} \\ &= \sigma_\mu^2 \left\{ \text{trace}(\mathcal{J}) - \text{trace}\left((\mathbf{X}'\mathcal{J}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1}\right) \right\} \end{aligned}$$

This requires the additional calculation of $\mathbf{X}'\mathcal{J}\mathbf{X}$, but that's just a $K \times K$ matrix. In the special case of a balanced sample with time-invariant regressors, $\mathbf{X}'\mathcal{J}\mathbf{X} = T\mathbf{X}'\mathbf{X}$, so this will simplify to:

$$\sigma_\mu^2 \{NT - TK\}$$

Note that the degrees of freedom correction on this term is TK , not K alone.

To compute (A.2) allowing for sampling variation, we observe that this is $\mathbf{u}'\bar{\mathcal{J}}\mathbf{u}$. Applying this to the residuals, we get

$$\begin{aligned} E\hat{\mathbf{u}}'\bar{\mathcal{J}}\hat{\mathbf{u}} &= \text{trace}(E\bar{\mathcal{J}}\hat{\mathbf{u}}\hat{\mathbf{u}}') \\ &= \text{trace}(\bar{\mathcal{J}}\mathbf{Q}_X\mathbf{Q}_X) \\ &= \sigma_\eta^2 \text{trace}(\bar{\mathcal{J}}\mathbf{Q}_X) + \sigma_\mu^2 \text{trace}(\bar{\mathcal{J}}\mathbf{Q}_X\mathcal{J}\mathbf{Q}_X) \end{aligned}$$

For the σ_η^2 term

$$\begin{aligned} \text{trace}(\bar{\mathcal{J}}\mathbf{Q}_X) &= \text{trace}(\bar{\mathcal{J}}) - \text{trace}(\bar{\mathcal{J}}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') \\ &= \text{trace}(\bar{\mathcal{J}}) - \text{trace}((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\bar{\mathcal{J}}\mathbf{X}) \end{aligned}$$

$\text{trace}(\bar{\mathcal{J}})$ is just N . For the balanced, time-invariant case, $\mathbf{X}'\bar{\mathcal{J}}\mathbf{X} = \mathbf{X}'\mathbf{X}$ so the full expression reduces to $N - K$. For σ_μ^2 ,

$$\begin{aligned} \text{trace}(\bar{\mathcal{J}}\mathbf{Q}_X\mathcal{J}\mathbf{Q}_X) &= \text{trace}(\bar{\mathcal{J}}\mathcal{J}) - 2\text{trace}(\bar{\mathcal{J}}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathcal{J}) \\ &\quad + \text{trace}(\bar{\mathcal{J}}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathcal{J}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') \end{aligned}$$

$\mathcal{J}\bar{\mathcal{J}} = \mathcal{J}$, so the trace in the center term can be simplified to

$$\text{trace}((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathcal{J}\mathbf{X}) \quad (\text{A.5})$$

which we've already seen. This evaluates to TK in the balanced time-invariant case. The final term can be rearranged to

$$\text{trace}(\mathbf{X}'\bar{\mathcal{J}}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathcal{J}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}) \quad (\text{A.6})$$

While apparently rather ugly, this is just the product of four $K \times K$ matrices, all of which we've needed for other parts of the calculation. For the special case, this will simplify to TK .

A.1.2 Amemiya or Wansbeek-Kapteyn (Fixed Effects-based)

This was proposed originally in Amemiya (1971) and extended to unbalanced samples by Wansbeek & Kapteyn (1989). In RATS, we call it WK, after the second reference.

Let $\hat{\beta}_{FE}$ be the slope coefficients from fixed effects estimation:

$$\hat{\beta}_{FE} = (\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\mathbf{y} \quad (\text{A.7})$$

The full sample covariance matrix of the residuals is now

$$\tilde{\mathbf{Q}}_X \Omega \tilde{\mathbf{Q}}_X' \quad (\text{A.8})$$

where $\tilde{\mathbf{Q}}_X = \mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}$ which, unlike \mathbf{Q}_X , is not symmetric. The first condition will come from analyzing the residuals from fixed effects, including the individual effects. This can be written $\mathbf{Q}\hat{\mathbf{u}}$. So we need

$$E\hat{\mathbf{u}}'\mathbf{Q}\hat{\mathbf{u}} = \text{trace}(\mathbf{Q}E\hat{\mathbf{u}}\hat{\mathbf{u}}') = \text{trace}(\mathbf{Q}\tilde{\mathbf{Q}}_X\Omega\tilde{\mathbf{M}}_X') \quad (\text{A.9})$$

The σ_μ^2 term in this vanishes because $\mathbf{Q}\mathcal{J} = 0$. For σ_η^2 , we get:

$$\begin{aligned} \text{trace}(\mathbf{Q}\tilde{\mathbf{M}}_X\tilde{\mathbf{M}}_X') &= \text{trace}\left(\mathbf{Q}\left(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\right)\left(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\right)'\right) \\ &= \text{trace}\left(\mathbf{Q}\left(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\right)\left(\mathbf{I} - \mathbf{Q}\mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\right)\right) \\ &= \text{trace}(\mathbf{Q} - \text{trace} \mathbf{Q}\mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}) \\ &\quad - \text{trace}(\mathbf{Q}\mathbf{Q}\mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}') \\ &\quad + \text{trace}(\mathbf{Q}\mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\mathbf{Q}\mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}') \\ &= \text{trace}\mathbf{Q} - \text{trace}((\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\mathbf{X}) \end{aligned}$$

where the final reduction uses the fact that $\mathbf{Q}\mathbf{Q} = \mathbf{Q}$. $\text{trace}(\mathbf{Q}) = n - N$ and $\text{trace}((\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\mathbf{X})$ is the number of time-varying regressors, so the coefficient on σ_η^2 will be the degrees of freedom from the fixed effects regression.

The second condition will come from analyzing squares of the individual averages of the raw residuals, the $\hat{\mathbf{u}} = \mathbf{y} - \mathbf{X}\hat{\beta}_{FE}$ formed *without* subtracting off

individual effects. Now, there is nothing forcing this to be mean zero in sample, so we first apply the operator Q_n which subtracts off the (full) sample mean. This has the properties that $Q_n Q_n = Q_n$ and $Q_n Q = Q$. Its orthogonal complement P_n satisfies $P_n \bar{J} = P_n$.

$$E\hat{u}'Q_n\bar{J}Q_n\hat{u} = \text{trace} \left(Q_n\bar{J}Q_n \left(I - X(X'QX)^{-1}X'Q \right) \Omega \left(I - X(X'QX)^{-1}X'Q \right)' \right) \quad (\text{A.10})$$

The σ_μ^2 term again simplifies quite a bit because $QJ = JQ = 0$ which eliminates the terms with the Q in (A.10), leaving only $\text{trace}(Q_n\bar{J}Q_nJ)$ which is $n - \frac{1}{n} \sum_i T_i^2$.

While the matrix multiplies in the σ_η^2 term don't produce zero matrices, the traces on the cross terms drop out because they can be rearranged to include a factor of $\bar{J}Q_nQ = \bar{J}Q = 0$. Thus, we are left with

$$\text{trace}(Q_n\bar{J}Q_n) + \text{trace}((X'QX)^{-1}X'Q_n\bar{J}Q_nX)$$

where

$$Q_n\bar{J}Q_n = (I - P_n)\bar{J}Q_n = (\bar{J} - P_n)Q_n = \bar{J}Q_n = \bar{J} - P_n$$

The trace of this is $N - 1$. The second trace requires the calculation of the matrices $X'\bar{J}X$ and $X'P_nX$.

A.1.3 Swamy-Arora (Fixed and Between)

This was proposed by Swamy & Arora (1972). It uses a combination of the fixed effects estimator and the between estimator to provide the two conditions. The FE condition is just (A.8) as before. The residuals from the between estimator have covariance matrix:

$$Q_X^{(P)} \Omega Q_X^{(P)'}$$

where

$$Q_X^{(P)} = I - X(X'PX)^{-1}X'P$$

We can now use the simpler $E\hat{u}'P\hat{u}$ since the between residuals will be mean zero. The σ_η^2 term requires calculation of:

$$\begin{aligned} \text{trace}(PQ_X^{(P)}Q_X^{(P)'}) &= \text{trace}(P(I - X(X'PX)^{-1}X'P)(I - PX(X'PX)^{-1}X')) \\ &= \text{trace}(P) - \text{trace}(PX(X'PX)^{-1}X'P) - \text{trace}(PPX(X'PX)^{-1}X') \\ &\quad + \text{trace}(PX(X'PX)^{-1}X'PPX(X'PX)^{-1}X') \end{aligned}$$

Since $PP = P$, the last three terms all simplify to $\text{trace}((X'PX)^{-1}X'PX)$ which is equal to K . Combined with $\text{trace}(P) = N$, we get $N - K$ as the coefficient on σ_η^2 .

For σ_μ^2 , we need to compute

$$\begin{aligned} & \text{trace}(\mathbf{P}\mathcal{J}) - \text{trace}(\mathbf{P}\mathbf{X}(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathbf{P}\mathcal{J}) - \text{trace}(\mathbf{P}\mathcal{J}\mathbf{P}\mathbf{X}(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}') \\ & + \text{trace}(\mathbf{P}\mathbf{X}(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathbf{P}\mathcal{J}\mathbf{P}\mathbf{X}(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}') \end{aligned}$$

The three matrix multiplies can all be rearranged into

$$\text{trace}((\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathbf{P}\mathcal{J}\mathbf{P}\mathbf{X})$$

$\mathbf{P}\mathcal{J}\mathbf{P}$ simplifies to \mathcal{J} . Since $\text{trace}(\mathbf{P}\mathcal{J}) = n$, the multiplier on σ_μ^2 reduces to $n - \text{trace}((\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathcal{J}\mathbf{X})$.

A.2 Two-Way Effects

Again, we'll compute these assuming that we can observe the true residuals:

$$u_{it} = \mu_i + \lambda_t + \eta_{it}$$

so

$$Eu_{it}^2 = \sigma_\mu^2 + \sigma_\lambda^2 + \sigma_\eta^2$$

Summing over the data set gives one equation in the three variances:

$$E \sum_{i,t} u_{it}^2 = \sigma_\mu^2 \left(\sum_i T_i \right) + \sigma_\lambda^2 \left(\sum_i T_i \right) + \sigma_\eta^2 \left(\sum_i T_i \right) \quad (\text{A.11})$$

The second condition comes from the squares of individual sums:

$$\begin{aligned} E \sum_i \frac{1}{T_i} \left(\sum_t u_{it} \right)^2 &= \sum_i \frac{1}{T_i} E \left(\sum_t (\mu_i + \lambda_t + \eta_{it}) \right)^2 \\ &= \sum_i \frac{1}{T_i} E \left(T_i \mu_i + \sum_t \lambda_t + \sum_t \eta_{it} \right)^2 \\ &= \sigma_\mu^2 \left(\sum_i T_i \right) + \sigma_\lambda^2 N + \sigma_\eta^2 N \end{aligned} \quad (\text{A.12})$$

The third condition comes for the squares of the time sums:

$$\begin{aligned} E \sum_t \frac{1}{N_t} \left(\sum_i u_{it} \right)^2 &= \sum_t \frac{1}{N_t} E \left(\sum_i (\mu_i + \lambda_t + \eta_{it}) \right)^2 \\ &= \sum_t \frac{1}{N_t} E \left(\sum_i \mu_i + N_t \lambda_t + \sum_i \eta_{it} \right)^2 \\ &= \sigma_\mu^2 T + \sigma_\lambda^2 \left(\sum_t N_t \right) + \sigma_\eta^2 T \end{aligned}$$

Solving these three gives the desired estimates.

A.2.1 Wallace-Hussain

When we decompose the sum of squares from OLS, the weights on σ_μ^2 and σ_η^2 are exactly as they were in the one-way model. By a similar calculation, the weight on σ_λ^2 is

$$\text{trace}(\mathcal{T}) - \text{trace}\left((\mathbf{X}'\mathcal{T}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1}\right)$$

where the first term will again just be n .

The two other conditions will come from $\mathbf{u}'\bar{\mathcal{J}}\mathbf{u}$ and $\mathbf{u}'\bar{\mathcal{T}}\mathbf{u}$. Again, the weights on σ_μ^2 and σ_η^2 in the first of these are exactly as they are in the one-way model, and those on σ_λ^2 and σ_η^2 in the second are similar with \mathcal{T} replacing \mathcal{J} . Getting σ_λ^2 in the individual average equation requires:

$$\begin{aligned} \text{trace}(\bar{\mathcal{J}}\mathbf{Q}_\mathbf{x}\mathcal{T}\mathbf{Q}_\mathbf{x}) &= \text{trace}(\bar{\mathcal{J}}\mathcal{T}) - \text{trace}(\bar{\mathcal{J}}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathcal{T}) \\ &\quad - \text{trace}(\bar{\mathcal{J}}\mathcal{T}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') \\ &\quad + \text{trace}(\bar{\mathcal{J}}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathcal{T}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') \end{aligned}$$

The final term rearranges to:

$$\text{trace}\left(\mathbf{X}'\bar{\mathcal{J}}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathcal{T}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\right)$$

which isn't particularly difficult to compute since it just requires grouped crossproducts. The center terms can be arranged and combined to make:

$$\text{trace}\left((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\bar{\mathcal{J}}\mathcal{T} + \mathcal{T}\bar{\mathcal{J}})\mathbf{X}\right)$$

For a balanced sample, $\bar{\mathcal{J}}\mathcal{T} = \mathcal{T}\bar{\mathcal{J}} = NP_n$. For an unbalanced sample, $\bar{\mathcal{J}}\mathcal{T}$ and $\mathcal{T}\bar{\mathcal{J}}$ aren't equal and aren't symmetric (though their sum is).

A.2.2 Amemiya (Fixed Effects-based)

Let $\hat{\beta}_{FE}$ be the slope coefficients from two-way fixed effects estimation (where, for simplicity, we'll now use \mathbf{Q} for the two-way fixed effects within operator):

$$\hat{\beta}_{FE} = (\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\mathbf{y}$$

The full sample covariance matrix of the residuals is now

$$\tilde{\mathbf{Q}}_\mathbf{x}\Omega\tilde{\mathbf{Q}}_\mathbf{x}'$$

where $\tilde{\mathbf{Q}}_\mathbf{x} = \mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}$. The first condition will come from analyzing the residuals from fixed effects. This can be written $\mathbf{Q}\hat{\mathbf{u}}$. So we need

$$E\hat{\mathbf{u}}'\mathbf{Q}\hat{\mathbf{u}} = \text{trace}(\mathbf{Q}E\hat{\mathbf{u}}\hat{\mathbf{u}}') = \text{trace}(\mathbf{Q}\tilde{\mathbf{Q}}_\mathbf{x}\Omega\tilde{\mathbf{Q}}_\mathbf{x}')$$

The σ_μ^2 and σ_λ^2 terms both vanish because $\mathbf{Q}\mathcal{J} = 0$ and $\mathbf{Q}\mathcal{T} = 0$. For σ_η^2 , as with the one-way analysis, we get a coefficient of $\text{trace}(\mathbf{Q}) - \text{trace}((\mathbf{X}'\mathbf{Q}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Q}\mathbf{X})$, which will be the degrees of freedom of the two-way fixed effects estimator.

The second and third conditions come from analyzing squares of the individual averages of the (centered) raw residuals, and from the squares of their time averages. When doing the individual averages, the coefficients on σ_μ^2 and σ_η^2 are computed exactly as above, and the analogous calculations are used for the coefficients on σ_λ^2 and σ_η^2 when analyzing the time averages. The only new calculation is for the cross effects: time with individual averages and individuals with time average. However, since $\mathbf{Q}\mathcal{T} = 0$ as well, the only term which isn't zeroed when computing the σ_λ^2 in the individual averages is the "constant", leaving

$$\text{trace}(\mathbf{Q}_n \bar{\mathcal{T}} \mathbf{Q}_n \mathcal{T}) = \text{trace}((\bar{\mathcal{T}} - \mathbf{P}_n) \mathcal{T}) \quad (\text{A.13})$$

$\text{trace}(\mathbf{P}_n \mathcal{T}) = \frac{1}{n} \sum_t N_t^2$, which simplifies to N in the balanced sample case. $\text{trace}(\bar{\mathcal{T}} \mathcal{T})$ will be N as long as the data set is an actual "panel" data set with at most one of each time period per individual.¹ So in the balanced case, (A.13) will be zero. Similarly, the coefficient on σ_μ^2 in the time average equation will be

$$\text{trace}(\mathbf{Q}_n \bar{\mathcal{T}} \mathbf{Q}_n \mathcal{T}) = \text{trace}((\bar{\mathcal{T}} - \mathbf{P}_n) \mathcal{T}) = \frac{1}{n} \sum_i T_i^2 - \text{trace}(\bar{\mathcal{T}} \mathcal{T})$$

which will be $T - \frac{1}{n} \sum_i T_i^2$, evaluating to zero in the balanced case.

A.2.3 Swamy-Arora (Fixed and Between)

Swamy-Arora uses the fixed effects estimator and between estimators in each direction to provide the conditions. The FE condition is the same as in Amemiya.

The coefficients on σ_η^2 and σ_μ^2 when working with the between individuals estimator are as in the one-way case, and on σ_η^2 and σ_λ^2 for the between time estimator are the direct analogues, so it's again only the cross effects that need to be analyzed. For σ_λ^2 in the individuals regression, we need to compute

$$\begin{aligned} &\text{trace}(\mathbf{P}\mathcal{T}) - \text{trace}(\mathbf{P}\mathbf{X}(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathbf{P}\mathcal{T}) - \text{trace}(\mathbf{P}\mathcal{T}\mathbf{P}\mathbf{X}(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}') \\ &\quad + \text{trace}(\mathbf{P}\mathbf{X}(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathbf{P}\mathcal{T}\mathbf{P}\mathbf{X}(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}') \end{aligned}$$

The three matrix multiplies can all be rearranged into

$$\text{trace}((\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathbf{P}\mathcal{T}\mathbf{P}\mathbf{X})$$

For a balanced sample, $\mathbf{P}\mathcal{T}\mathbf{P}$ is an $NT \times NT$ matrix of 1's. For an unbalanced sample, it has a structure which isn't easy to describe, and $\mathbf{X}'\mathbf{P}\mathcal{T}\mathbf{P}\mathbf{X}$ is most easily computed by first doing $\mathbf{P}\mathbf{X}$, then applying the \mathcal{T} cross product operation to the result.

¹A general two-way grouped sample might not satisfy this.

Corrected Covariance Matrices

In some cases, it won't be possible to do a GLS correction for serial correlation or cross section dependence, either because of a problem with pre-determined but not exogenous regressors (for instance, in multi-step predictability regressions) or simply because of a lack of data in one dimension. The following assumes that we fall back on using OLS, though the same ideas apply to other forms of regressions.

If we run an OLS regression on a panel data set, we get the following as the standard asymptotic breakdown:

$$\sqrt{n}(\hat{\beta} - \beta) = \left(\frac{1}{n} \sum_{it} X'_{it} X_{it} \right)^{-1} \left(\frac{1}{\sqrt{n}} \sum_{it} X_{it} u_{it} \right)$$

where n is the total number of data points. As usual, the first factor will be assumed to converge in probability to a fixed matrix, and the sample average serves as an estimator for it. It's the second factor which doesn't fit into any of the forms given in Hansen (1982) since, with panel data, it involves N separate time series, rather than just 1. However, the same ideas used there can be applied.

In general, we need to compute:

$$\text{cov} \left\{ \sum_{i,t} \mathbf{X}_{it} u_{it} \right\} = E \sum_{i,j,t,s} \mathbf{X}'_{it} u_{it} u_{js} \mathbf{X}_{js}$$

The sample analogue of this isn't a feasible estimator since it's a rank one matrix as the outer product of the single sum

$$\sum_{i,t} \mathbf{X}_{it} u_{it}$$

We need to make some assumptions which will reduce the freedom of the relationship.

Time Clustered Standard Errors

If we assume (conditional on \mathbf{X}) that:

$$E u_{it} u_{js} = 0 \text{ if } t \neq s$$

that is, no time dependence (which rules out individual effects, since they create correlation between u_{it} and u_{is} for all t and s), then

$$E \sum_t \sum_{i,j} \mathbf{X}'_{it} u_{it} u_{jt} \mathbf{X}_{jt} = E \sum_t \left(\sum_i \mathbf{X}_{it} u_{it} \right)' \left(\sum_i \mathbf{X}_{it} u_{it} \right)$$

which is the covariance matrix *clustered by time*. With RATS, you get these using the option `CLUSTER=%PERIOD (T)` for PANEL-dated data, or `CLUSTER=time id series` for other data. Note that this will be singular if the number of time periods is less than the number of regressors since it's the sum of T rank one matrices.

Panel Corrected Standard Errors

With the similar, but stronger assumption that:

$$Eu_{it}u_{js} = \begin{cases} 0 & \text{if } t \neq s \\ \sigma_{ij} & \text{if } t = s \end{cases}$$

we get

$$E \sum_t \sum_{i,j} \mathbf{X}'_{it} u_{it} u_{jt} \mathbf{X}_{jt} = E \sum_t \sum_{i,j} \sigma_{ij} \mathbf{X}'_{it} \mathbf{X}_{jt}$$

These give rise to *Panel Corrected Standard Errors* (Beck & Katz (1995)). This is an alternative to using a feasible GLS estimator. If Σ were known, GLS would be more efficient than OLS. However, if Σ isn't known, and if T isn't much larger than N , then feasible GLS can be less accurate than OLS because it weights observations using the inverse of a poorly estimated Σ matrix. While the PCSE use that same matrix, they don't invert it, and so aren't subject to the same degree of sampling error. After you estimate the equation by **LINREG**, you can use the procedure `@REGPCSE` to redo the regression with the corrected covariance matrix.

Individual Clustered Standard Errors

If instead, we assume:

$$Eu_{it}u_{js} = 0 \text{ if } i \neq j$$

that is, no correlation between individuals, we get

$$E \sum_i \sum_{t,s} \mathbf{X}'_{it} u_{it} u_{is} \mathbf{X}_{is} = E \sum_i \left(\sum_t \mathbf{X}_{it} u_{it} \right)' \left(\sum_t \mathbf{X}_{it} u_{it} \right)$$

which are (individual) clustered standard errors. These allow for arbitrary patterns of serial correlation within an individual. These can be done directly on the **LINREG** using `LWINDOW=PANEL` or `CLUSTER=%INDIV (T)` for PANEL-dated data, or `CLUSTER=individual id series` for other data. This will be singular if the number of individuals is less than the number of regressors since it's the sum of N rank one matrices.

Individual HAC Standard Errors

A somewhat sharper assumption is:

$$Eu_{it}u_{js} = 0 \text{ if } i \neq j \text{ or } |t - s| > L$$

which limits the serial correlation within an individual record to no more than L lags or leads (which again rules out individual effects). There is no correlation between individuals. A feasible estimator for this requires some type of window to prevent the covariance matrix from going non-positive definite.

$$\sum_i \sum_t \sum_{l=-L}^L w_l u_{it} u_{i,t-l} \mathbf{X}'_{it} \mathbf{X}_{i,t-l}$$

where the w_l are lag weights. A particular form of this would be Newey-West, which is applied individual by individual. This can be done directly on the **LINREG** using the **LAGS** and **LWINDOW** options (like **LWINDOW=NEWKEY**).

Full Panel HAC Standard Errors

The most general form that can be handled is:

$$Eu_{it}u_{js} = 0 \text{ if } |t - s| > L$$

This allows for serial correlation both within and between individuals. However, again, this rules out individual effects.

$$\sum_{i,j} \sum_t \sum_{l=-L}^L w_l u_{it} u_{j,t-l} \mathbf{X}'_{it} \mathbf{X}_{j,t-l}$$

which is a full panel HAC correction. After you estimate the equation by **LINREG**, you can use the procedure **@REGPCSE** with the option **METHOD=PHAC** and **LAGS=value of L** to redo the regression with the corrected covariance matrix.

Probability Distributions

C.1 Chi-Squared Distribution

Parameters	Degrees of freedom (ν).
Kernel	$x^{(\nu-2)/2} \exp(-x/2)$
Range	$[0, \infty)$
Mean	ν
Variance	2ν
Main uses	Asymptotic distribution. Prior, exact and approximate posterior for the precision (reciprocal of variance) of residuals or other shocks in a model.
Density function	<code>%CHISQDENSITY(x, nu)</code> is the (non-logged) density with <code>nu</code> degrees of freedom at <code>x</code> .
Tail Probability	<code>%CHISQR(x, nu)</code> returns the probability that a chi-squared with <code>nu</code> degrees of freedom exceeds <code>x</code> .
Inverse Tail Probability	<code>%INVCHISQR(p, nu)</code> returns the critical value for probability p .
Random Draws	<code>%RANCHISQR(nu)</code> draws one or more (depending upon the target) independent chi-squareds with <code>NU</code> degrees of freedom.

C.2 (Scaled) Inverse Chi-Squared Distribution

Parameters	Degrees of freedom (ν) and scale (τ^2). An inverse chi-squared is the reciprocal of a chi-squared combined with scaling factor which represents a “target” variance that the distribution is intended to represent. (Note that the mean is roughly τ^2 for large degrees of freedom.)
Kernel	$x^{-(a+1)} \exp(-bx^{-1})$
Integrating Constant	$b^a / \Gamma(a)$
Support	$[0, \infty)$
Mean	$\frac{\nu\tau^2}{\nu-2}$ if $\nu > 2$
Variance	$\frac{2\nu^2\tau^4}{(\nu-2)^2(\nu-4)}$ if $\nu > 4$
Main uses	Prior, exact and approximate posterior for the variance of residuals or other shocks in a model. The closely-related inverse gamma (Appendix C.6) can be used for that as well, but the scaled inverse chi-squared tends to be more intuitive.
DensityFunction	<code>%LOGINVCHISQRDENSITY(x, nu, tausq)</code> . Added with RATS 9.1.
Moment Matching	<code>%InvChisqrParms(mean, sd)</code> (external function) returns the 2-vector of parameters ((ν, τ^2) in that order) for the parameters of an inverse chi-squared with the given mean and standard deviation. If <code>sd</code> is the missing value, this will return $\nu = 4$, which is the largest value of ν which gives an infinite variance.
Random draws	You can use <code>(nu*tausq)/%ranchisqr(nu)</code> . Note that you divide by the random chi-squared.

C.3 Univariate Normal

Parameters	Mean (μ), Variance (σ^2)
Kernel	$\sigma^{-1} \exp \left(-\frac{(x - \mu)^2}{2\sigma^2} \right)$
Support	$(-\infty, \infty)$
Mean	μ
Variance	σ^2
Main uses	Distribution of error terms in univariate processes. Asymptotic distributions. Prior, exact and approximate posteriors for parameters with unlimited ranges.
Density Function	<code>%DENSITY(x)</code> is the non-logged standard Normal density. More generally, <code>%LOGDENSITY(variance,u)</code> . Use <code>%LOGDENSITY(sigmasq,x-mu)</code> to compute $\log f(x \mu, \sigma^2)$.
CDF	<code>%CDF(x)</code> is the standard Normal CDF (running from 0 in the left tail to 1 in the right). To get $F(x \mu, \sigma^2)$, use <code>%CDF((x-mu)/sigma)</code> . <code>%ZTEST(z)</code> gives the two-tailed tail probability (probability a $N(0,1)$ exceeds z in absolute value).
Inverse CDF	<code>%INVNORMAL(p)</code> is the standard Normal inverse CDF.
Random Draws	<code>%RAN(s)</code> draws one or more (depending upon the target) independent $N(0, s^2)$. <code>%RANMAT(m,n)</code> draws a matrix of independent $N(0, 1)$.

C.4 Multivariate Normal

Parameters	Mean (μ), Covariance matrix (Σ) or precision (H)
Kernel	$ \Sigma ^{-1/2} \exp \left(-\frac{1}{2} (x - \mu)' \Sigma^{-1} (x - \mu) \right)$ or $ H ^{1/2} \exp \left(-\frac{1}{2} (x - \mu)' H (x - \mu) \right)$
Support	\mathbb{R}^n
Mean	μ
Variance	Σ or H^{-1}
Main uses	Distribution of multivariate error processes. Asymptotic distributions. Prior, exact and approximate posteriors for a collection of parameters with unlimited ranges.
Density Function	<code>%LOGDENSITY(sigma,u)</code> . To compute $\log f(x \mu, \Sigma)$ use <code>%LOGDENSITY(sigma,x-mu)</code> . (The same function works for univariate and multivariate Normals).
Distribution Function	<code>%BICDF(x,y,rho)</code> returns $P(X \leq x, Y \leq y)$ for a bivariate Normal with mean zero, variance 1 in each component and correlation rho.
Random Draws	<code>%RANMAT(m,n)</code> draws a matrix of independent $N(0,1)$. <code>%RANMVNORMAL(F)</code> draws an n -vector from a $N(0, FF')$, where F is any factor of the covariance matrix. This setup is used (rather than taking the covariance matrix itself as the input) so you can do the factor just once if it's fixed across a set of draws. To get a single draw from a $N(\mu, \Sigma)$, use <code>MU+%RANMVNORMAL(%DECOMP(SIGMA))</code> <code>%RANMVPOST</code> , <code>%RANMVPOSTCMOM</code> , <code>%RANMVKRON</code> and <code>%RANMVKRONCMOM</code> are specialized functions which draw multivariate Normals with calculations of the mean and covariance matrix from other matrices.

C.5 Gamma Distribution

Parameters	shape (a) and scale (b), alternatively, degrees of freedom (ν) and mean (μ). The RATS functions use the first of these. The relationship between them is $a = \nu/2$ and $b = \frac{2\mu}{\nu}$. The chi-squared distribution with ν degrees of freedom is a special case with $\mu = \nu$.
Kernel	$x^{a-1} \exp\left(-\frac{x}{b}\right)$ or $x^{(\nu/2)-1} \exp\left(-\frac{x\nu}{2\mu}\right)$
Support	$[0, \infty)$
Mean	ba or μ
Variance	b^2a or $\frac{2\mu^2}{\nu}$
Main uses	Prior, exact and approximate posterior for the precision (reciprocal of variance) of residuals or other shocks in a model
Density function	<code>%LOGGAMMADENSITY(x, a, b)</code> . For the $\{\nu, \mu\}$ parameterization, use <code>%LOGGAMMADENSITY(x, .5*nu, 2.0*mu/nu)</code>
Random Draws	<code>%RANGAMMA(a)</code> draws one or more (depending upon the target) independent Gammas with unit scale factor. Use <code>b*%RANGAMMA(a)</code> to get a draw from $Gamma(a, b)$. If you are using the $\{\nu, \mu\}$ parameterization, use <code>2.0*mu*%RANGAMMA(.5*nu)/nu</code> . You can also use <code>mu*%RANCHISQR(nu)/nu</code> .
Moment Matching	<code>%GammaParms(mean, sd)</code> (external function) returns the 2-vector of parameters ((a, b) parameterization) for a gamma with the given mean and standard deviation.

C.6 Inverse Gamma Distribution

Parameters	shape (a) and scale (b). An inverse gamma is the reciprocal of a gamma. The special case is the scaled inverse chi-squared (Appendix C.2 with parameters ν (degrees of freedom) and τ^2 (scale parameter) which has $a = \nu/2$ and $b = \nu\tau^2/2$.
Kernel	$x^{-(a+1)} \exp(-bx^{-1})$
Integrating Constant	$b^a/\Gamma(a)$
Support	$[0, \infty)$
Mean	$\frac{b}{(a-1)}$ if $a > 1$
Variance	$\frac{b^2}{(a-1)^2(a-2)}$ if $a > 2$
Main uses	Prior, exact and approximate posterior for the variance of residuals or other shocks in a model. For these purposes, it's usually simpler to directly use the scaled inverse chi-squared variation.
Density Function	<code>%LOGINVGAMMADENSITY(x, a, b)</code> . Added with RATS 9.1.
Moment Matching	<code>%InvGammaParms(mean, sd)</code> (external function) returns the 2-vector of parameters ((a, b) parameterization) for the parameters of an inverse gamma with the given mean and standard deviation. If <code>sd</code> is the missing value, this will return $a = 2$, which is the largest value of a which gives an infinite variance.
Random draws	You can use <code>b/%rangamma(a)</code> . A draw from a scaled inverse chi-squared is typically done with <code>nu*tausqr/%ranchisqr(nu)</code> .

C.7 Wishart Distribution

Parameters	Scaling \mathbf{A} (symmetric $n \times n$ matrix) and degrees of freedom (ν). This only has a proper density if $\nu > n - 1$ and \mathbf{A} is positive definite.
Kernel	$\exp\left(-\frac{1}{2}\text{trace}(\mathbf{A}^{-1}\mathbf{X})\right) \mathbf{X} ^{\frac{1}{2}(\nu-n-1)}$
Support	Positive definite symmetric matrices
Mean	$\nu\mathbf{A}$
Main uses	Prior, exact and approximate posterior for the precision matrix (inverse of covariance matrix) of residuals in a multivariate regression, though that is mainly in the inverse form (Appendix C.8) since that would be the distribution of the covariance matrix itself.
Random Draws	<p><code>%RANWISHART (n, nu)</code> draws a single $n \times n$ Wishart matrix with $\mathbf{A} = \mathbf{I}$ and degrees of freedom ν.</p> <p><code>%RANWISHARTF (F, nu)</code> draws a single $n \times n$ Wishart matrix with $\mathbf{A} = \mathbf{F}\mathbf{F}'$ and degrees of freedom ν. \mathbf{F} can be any factor of \mathbf{A}, but would typically be computed as the Cholesky factor using <code>%DECOMP</code>.</p>

C.8 Inverse Wishart Distribution

Parameters	Scaling Ψ (symmetric $n \times n$ matrix) and degrees of freedom (ν). This only has a proper density if $\nu > n - 1$ and Ψ is positive definite.
Kernel	$\exp\left(-\frac{1}{2}\text{trace}(\Psi\mathbf{X}^{-1})\right) \mathbf{X} ^{-\frac{1}{2}(\nu+n+1)}$
Support	Positive definite symmetric matrices
Mean	$\frac{1}{\nu-n-1}\Psi$
Main uses	Prior, exact and approximate posterior for the covariance matrix of residuals in a multivariate regression with Gaussian residuals.
Diffuse versions	The density function is improper if $\nu < n - 1$, but the improper prior with $\nu = 0$ and $\Psi = 0$ has kernel $ \mathbf{X} ^{-\frac{1}{2}(n+1)}$ which forms the Jeffrey's prior for inference on the covariance matrix.
Combining Densities	If $\mathbf{X} \sim IW(\Psi_1, \nu_1)$ and $\mathbf{X} \sim IW(\Psi_2, \nu_2)$ are inverse Wishart densities for \mathbf{X} , then the posterior from combining them has $\mathbf{X} \sim IW(\Psi_1 + \Psi_2, \nu_1 + \nu_2 + n + 1)$.
Random Draws	<code>%RANWISHARTI(F, nu)</code> draws a single $n \times n$ inverse Wishart matrix with $\mathbf{F}\mathbf{F}' = \Psi^{-1}$ and degrees of freedom ν . Note that \mathbf{F} needs to be a factor of the <i>inverse</i> . \mathbf{F} can be any factor matrix, but is typically the Cholesky factor, computed using <code>%DECOMP</code> .
Notes	The basic result has the data evidence on the covariance matrix of Gaussian residuals summarized as an inverse Wishart with $\Psi = T\hat{\Sigma}$, where T is the number of observations and $\hat{\Sigma}$ the sample covariance matrix of residuals (thus Ψ itself is the sum of the outer products of the residuals). The degrees of freedom for the inverse Wishart from the data itself are typically $T - (n + 1)$ (sometimes less some additional adjustments for regressors, depending upon the form of conditioning). The $n + 1$ is needed because you don't really have the ability to estimate a covariance matrix until you have that many observations. Combining data with the Jeffrey's prior "corrects" the degrees of freedom so the posterior value of $\nu = T$.

An informative prior is generally based upon a prior belief on the value of \mathbf{X} . Because \mathbf{X} is typically the covariance matrix Σ , call this Σ_0 . The corresponding value of Ψ for this is $\alpha\Sigma_0$ where the prior degrees of freedom are $\alpha + n + 1$. Combined with data, this gives an inverse Wishart with $T + \alpha$ degrees of freedom and Ψ matrix which is $T\hat{\Sigma} + \alpha\Sigma_0$, thus (roughly) sample and non-sample information on Σ weighted by the number of actual and “dummy” observations.

Properties of Multivariate Normals

The density function for a jointly Normal random n -vector \mathbf{x} with mean μ and covariance matrix Σ is

$$(2\pi)^{-n/2} |\Sigma|^{-1/2} \exp \left(-\frac{1}{2} (\mathbf{x} - \mu)' \Sigma^{-1} (\mathbf{x} - \mu) \right)$$

Manipulations with this can quickly get very messy. Fortunately, the exponent in this (which is just a scalar – the result of the quadratic form) contains enough information that we can just read from it μ and Σ . In particular, if we can determine that the kernel of the density of \mathbf{x} (the part of the density which includes all occurrences of \mathbf{x}) takes the form

$$\exp \left(-\frac{1}{2} Q(\mathbf{x}) \right) \quad , \text{where} \quad Q(\mathbf{x}) = \mathbf{x}' \mathbf{A} \mathbf{x} - 2\mathbf{x}' \mathbf{b} + c$$

then, by completing the square, we can turn this into

$$Q(\mathbf{x}) = (\mathbf{x} - \mathbf{A}^{-1}\mathbf{b})' \mathbf{A} (\mathbf{x} - \mathbf{A}^{-1}\mathbf{b}) + (c - \mathbf{b}' \mathbf{A}^{-1} \mathbf{b})$$

Thus $\Sigma = \mathbf{A}^{-1}$ and $\mu = \mathbf{A}^{-1}\mathbf{b}$. The final part of this, $(c - \mathbf{b}' \mathbf{A}^{-1} \mathbf{b})$, doesn't involve \mathbf{x} and will just wash into the constant of integration for the Normal. We might need to retain it for analyzing other parameters (such as the residual variance), but it has no effect on the conditional distribution of \mathbf{x} itself.

One standard manipulation of multivariate Normals comes in the Bayesian technique of combining a prior and a likelihood to produce a posterior density. In the standard Normal linear model, the data have density

$$f(\mathbf{Y}|\beta) \sim N(\mathbf{X}\beta, \sigma^2 \mathbf{I})$$

The (log) kernel of the density is

$$-\frac{1}{2\sigma^2} (\mathbf{Y} - \mathbf{X}\beta)' (\mathbf{Y} - \mathbf{X}\beta) = -\frac{1}{2} (\beta' (\sigma^{-2} \mathbf{X}' \mathbf{X}) \beta - 2\beta' \sigma^{-2} \mathbf{X}' \mathbf{Y} + \sigma^{-2} \mathbf{Y}' \mathbf{Y})$$

Looking at this as a function of β , we read off

$$\beta|\mathbf{Y} \sim N \left((\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{Y}, \sigma^2 (\mathbf{X}' \mathbf{X})^{-1} \right) = N \left(\hat{\beta}, \sigma^2 (\mathbf{X}' \mathbf{X})^{-1} \right)$$

Now, suppose that, in addition to the data, we have the prior

$$\beta \sim N(\beta^*, \mathbf{H}^{-1})$$

and we write $\hat{\mathbf{H}} = \sigma^{-2} (\mathbf{X}'\mathbf{X})$ (the inverse of the least squares covariance matrix). If we multiply the densities, the only parts that include β will be the two quadratic parts, which we can add in log form to get

$$Q(\beta) = (\beta - \hat{\beta})' \hat{\mathbf{H}} (\beta - \hat{\beta}) + (\beta - \beta^*)' \mathbf{H} (\beta - \beta^*) \quad (\text{D.1})$$

Expanding this gives us

$$Q(\beta) = \beta' (\hat{\mathbf{H}} + \mathbf{H}) \beta - 2\beta' (\hat{\mathbf{H}}\hat{\beta} + \mathbf{H}\beta^*) + \dots$$

where the extra terms don't involve β . Thus, the posterior for β is

$$\beta \sim N \left((\hat{\mathbf{H}} + \mathbf{H})^{-1} (\hat{\mathbf{H}}\hat{\beta} + \mathbf{H}\beta^*), (\hat{\mathbf{H}} + \mathbf{H})^{-1} \right)$$

Notice that the posterior mean is a matrix weighted average of the two input means, where the weights are the inverses of the variances. The inverse of the variance (of a Normal) is known as the *precision*. Notice that precision is additive: the precision of the posterior is the sum of the precisions of the data information and prior.

If we need to keep track of the extra terms, there's a relatively simple way to evaluate this. If we write the posterior mean and precision as:

$$\bar{\beta} = (\hat{\mathbf{H}} + \mathbf{H})^{-1} (\hat{\mathbf{H}}\hat{\beta} + \mathbf{H}\beta^*), \bar{\mathbf{H}} = \hat{\mathbf{H}} + \mathbf{H}$$

then we have

$$Q(\beta) = (\beta - \bar{\beta})' \bar{\mathbf{H}} (\beta - \bar{\beta}) + Q(\bar{\beta}) \quad (\text{D.2})$$

The first term in (D.2) has value 0 at $\bar{\beta}$, so using this and (D.1) gives

$$Q(\bar{\beta}) = (\bar{\beta} - \hat{\beta})' \hat{\mathbf{H}} (\bar{\beta} - \hat{\beta}) + (\bar{\beta} - \beta^*)' \mathbf{H} (\bar{\beta} - \beta^*)$$

As another example, consider the partitioned process

$$\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix} \sim N \left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma'_{12} & \Sigma_{22} \end{bmatrix} \right)$$

The Q function takes the form

$$\begin{bmatrix} \mathbf{Y}_1 - \mu_1 \\ \mathbf{Y}_2 - \mu_2 \end{bmatrix}' \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma'_{12} & \Sigma_{22} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Y}_1 - \mu_1 \\ \mathbf{Y}_2 - \mu_2 \end{bmatrix}$$

For now, let's just write the inverse in partitioned form without solving it out, thus

$$\begin{bmatrix} \mathbf{Y}_1 - \mu_1 \\ \mathbf{Y}_2 - \mu_2 \end{bmatrix}' \begin{bmatrix} \Sigma^{11} & \Sigma^{12} \\ \Sigma^{12'} & \Sigma^{22} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 - \mu_1 \\ \mathbf{Y}_2 - \mu_2 \end{bmatrix}$$

This expands to

$$(\mathbf{Y}_1 - \mu_1)' \Sigma^{11} (\mathbf{Y}_1 - \mu_1) + 2(\mathbf{Y}_1 - \mu_1)' \Sigma^{12} (\mathbf{Y}_2 - \mu_2) + (\mathbf{Y}_2 - \mu_2)' \Sigma^{22} (\mathbf{Y}_2 - \mu_2)$$

where the cross terms are scalars which are transposes of each other, and hence equal, hence the 2 multiplier. If we now want to look at $\mathbf{Y}_1|\mathbf{Y}_2$, we get immediately that this has covariance matrix

$$(\Sigma^{11})^{-1}$$

and mean

$$\mu_1 - (\Sigma^{11})^{-1} \Sigma^{12} (\mathbf{Y}_2 - \mu_2)$$

If we want to reduce this to a formula in the original matrices, we can use partitioned inverse formulas to get

$$(\Sigma^{11})^{-1} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}$$

and

$$(\Sigma^{11})^{-1} \Sigma^{12} = -\Sigma_{12} \Sigma_{22}^{-1}$$

thus

$$\mathbf{Y}_1|\mathbf{Y}_2 \sim N(\mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (\mathbf{Y}_2 - \mu_2), \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21})$$

Note that the covariance matrix of the conditional distribution satisfies $\Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21} \leq \Sigma_{11}$ and that it doesn't depend upon the actual data observed, even if those data are seriously in conflict with the assumed distribution.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

`<https://fsf.org/>`

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated

herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or

XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that

carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together

with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a

translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document

does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

Bibliography

- Amemiya T 1971 *International Economic Review* **12**(1), 1–13.
- Anderson T W & Hsiao C 1982 *Journal of Econometrics* **18**(1), 47–82.
- Andrews D W K & Monahan C J 1992 *Econometrica* **60**(4), 953–966.
- Arellano M & Bond S 1991 *Review of Economic Studies* **58**(2), 277–297.
- Baltagi B 2008 *Econometric Analysis of Panel Data* 4th edn West Sussex: Wiley.
- Beck N & Katz J 1995 *American Political Science Review* **89**(3), 634–647.
- Berkowitz J & Kilian L 2000 *Econometric Reviews* **19**(1), 1–48.
- Breitung J 2000 in B Baltagi, T Fomby & R. C Hill, eds, ‘Nonstationary Panels, Panel Cointegration and Dynamic Panels’ *Advances in Econometrics* Emerald Group Publishing Limited pp. 161–177.
- Canova F & Ciccarelli M 2009 *International Economic Review* **50**(3), 929–959.
- Enders W 2010 *Applied Econometric Time Series* 3rd edn Wiley.
- Greene W 2007 *Econometric Analysis* 5th edn Prentice-Hall.
- Hadri K 2000 *Econometrics Journal* **3**(2), 148–161.
- Hansen B 1992 *Journal of Econometrics* **53**(1), 87–121.
- Hansen B 1999 *Journal of Econometrics* **93**(2), 345–368.
- Hansen L 1982 *Econometrica* **50**(4), 1029–1054.
- Harris R D F & Tzavalis E 1999 *Journal of Econometrics* **91**(2), 201–226.
- Hayashi F & Sims C 1983 *Econometrica* **51**(3), 783–798.
- Holtz-Eakin D, Newey W & Rosen H 1988 *Econometrica* **56**(6), 1371–95.
- Holtz-Eakin D, Newey W & Rosen H 1989 *International Economic Review* **30**(2), 415–429.
- Im K S, Pesaran M H & Shin Y 2003 *Journal of Econometrics* **115**(1), 53–74.
- Kiviet J F 1995 *Journal of Econometrics* **68**(1), 53–78.
- Koop G 2003 *Bayesian Econometrics* West Sussex: Wiley.

- Kwiatkowski D, Phillips P, Schmidt P & Shin Y 1992 *Journal of Econometrics* **54**(1-3), 159–178.
- Levin A, Lin C F & Chu C S 2002 *Journal of Econometrics* **108**(1), 1–24.
- Mark N & Sul D 2003 *Oxford Bulletin of Economics and Statistics* **65**(5), 655–680.
- Nickell S 1981 *Econometrica* **49**(6), 1417–26.
- Pedroni P 2001 *Review of Economics and Statistics* **83**(4), 727–731.
- Pedroni P 2007 *Journal of Applied Econometrics* **22**(2), 429–451.
- Pesaran M H, Shin Y & Smith R P 1999 *Journal of American Statistical Association* **94**(446), 621–634.
- Phillips P & Hansen B 1990 *Review of Economic Studies* **57**(1), 99–125.
- Phillips P & Ouliaris S 1990 *Econometrica* **58**(1), 165–193.
- Sims C, Stock J & Watson M 1990 *Econometrica* **58**(1), 113–144.
- Stock J & Watson M 1993 *Econometrica* **61**(4), 783–820.
- Swamy P 1970 *Econometrica* **38**(2), 311–323.
- Swamy P & Arora S 1972 *Econometrica* **40**(2), 261–275.
- Wallace T D & Hussain A 1969 *Econometrica* **37**(1), 55–72.
- Wansbeek T & Kapteyn A 1989 *Journal of Econometrics* **41**(3), 341–61.
- Wooldridge J 2010 *Econometric Analysis of Cross Section and Panel Data* 2nd edn MIT Press.

Index

- A-B instruments, 51
- ABDATA.DTA data set, 48
- @ABLAGS** procedure, 51, 143
 - COLLAPSE option, 52
 - MINLAG option, 53
- Absorbing state, 112
- AH estimator, 50
- AH-D estimator, 50
- AH-L estimator, 50
- Amemiya estimator, 166
- Amemiya, T., 166
- Anderson, T.W., 49
- Anderson-Hsiao estimator, 50
- Andrews, D., 97
- Arellano, M., 48, 51
- Arellano-Bond estimator, 52
- Arora, S., 167
- Balanced panel, 3
- Baltagi, B., 8, 19, 42, 60
- Beck, N., 172
- Berkowitz, J., 118
- Between estimates, 25
- %BICDF** function, 177
- Bond, S., 48, 51
- BOOT** instruction, 11
- Bootstrapping, 117
- @BREITUNG** procedure, 84
- Breitung, J., 82
- Brownian bridge, 85
- CALENDAR** instruction
 - PANELOBS option, 3
 - YPP option, 49
- Causality test, 153
- %CDF** function, 176
- Chi-squared distribution, 174, 175
- Chow test
 - for heterogeneity, 23
- Chu, C.-S., 28
- CLUSTER option, 14
- Clustered standard errors, 13
 - by time, 172
- %CMOM** matrix, 52
- Conditional logit, 65
- CORNWELL&RUPERT.XLS data set, 13, 33
- %DENSITY** function, 176
- DFC option, 18
- Double-blind trial, 16
- Dynamic OLS, 95, 96
- Enders, W., 6
- FELogitElement** user function, 67
- FIND** instruction, 63
- Frisch-Waugh theorem, 18
- Fuller-Battese estimator, 32
- Fully modified least squares, 92
- G7PANEL.XLS data set, 144
- Gamma distribution, 178, 179
- GASOLINE.XLS data set, 42
- Gauss-Hermite integration, 69, 116
- @GaussHermite** procedure, 69
- Greene, W., 11, 19, 32
- Grouped data, 2
- Grunfeld data set, 19
- GRUNFELD.XLS data set, 19
- GSET** instruction, 9
- @HADRI** procedure, 85
- Hadri test, 85
- Hadri, K., 85
- Hansen, B., 3, 92
- Hansen, L., 51, 52
- Harris, R.D.F., 81
- Harris-Tzavalis test, 81
- Hausman test, 34
 - two-way effects, 44
- Heterogeneous parameters, 2
- Hierarchical prior, 133

- Holtz-Eakin, D., 7, 51, 141
- Homogeneous parameters, 2
- Hsiao, C., 49
- @HTUNIT** procedure, 81
- Hussain, A., 163
- ι vector, 29
- %%IBETAS matrix, 94, 133
- Im, K., 81
- Im-Pesaran-Shin test, 81
- Incidental parameters problem, 59
- Index model, 64
- %INDIV function, 3
- Inverse Wishart distribution, 181
- INVEST.TXT data set, 3
- %INVNORMAL function, 176
- @IPSHIN** procedure, 82
- %%ISTDERRS matrix, 94
- %%ITSTATS matrix, 94
- %%IXX matrix, 94
- J_T matrix, 29
- Kapteyn, A., 166
- Katz, J., 172
- Kilian, L., 118
- Kiviet, J., 49, 53
- Koop, G., 19
- LABOR.DAT data set, 7
- LABOR.XLS data set, 142
- Least Squares Dummy Variable, 17
- Levin, A., 78
- Levin-Lin-Chu test, 78
- @LEVINLIN** procedure, 79
- LFP.DTA data set, 64
- Lin, C.-F., 78
- LINREG** instruction
 - CREATE option, 130
- %LOGDENSITY function, 176, 177
- Logit model
 - conditional, 65
- LSDV, 21
- LSDV estimation, 17
- LSDVC estimator, 54
- @LSDVC** procedure, 55
- LWINDOW= PANEL option, 13, 142
- Mark, N., 10, 96
- MCOV** instruction, 36
 - LWFORM option, 52
- Mean Group estimator, 129
- @MEANGROUP** procedure, 130
- Minnesota prior, 148
- Monahan, C., 97
- Multivariate Normal distribution, 177, 183
- Newey, W., 7, 51, 141
- %NGROUP variable, 8
- Nickell, S., 46
- NLLS** instruction
 - START option, 61
- Non-linear least squares, 59
- NOPANEL option, 9
- Normal distribution, 176
 - multivariate, 177, 183
- %NREGSYSTEM variable, 24
- Numerical integration
 - Gauss-Hermite, 69
- Ouliaris, S., 89
- @PANCOINT** procedure, 88
- Panel Corrected Standard Errors, 172
- PANEL** instruction
 - ENTRY option, 9
 - GLS option, 10
 - GROUP option, 9
 - INDIV option, 9
 - SMPL option, 9
 - DUMMIES option, 21
 - ID option, 10, 67
 - IDENTRIES option, 10, 67
 - MEAN option, 44
 - SMPL option, 23
 - TIME option, 44
 - VINDIV option, 37
 - VRANDOM option, 37
 - VTIME option, 44
- PANEL-dated data, 3
- Panel-robust standard errors, 13

- PANEL.XLS data set, 6
- @PANELDOLS** procedure, 95
- @PANELFM** procedure, 93
- PATENTDATA.TXT data set, 8
- Pedroni, P., 88, 90, 92
- PEDRONI-DATA.TXT data set, 90
- PENNGROW.TXT data set, 49
- PENNXRATE.DTA data set, 78
- %PERIOD function, 3
- Pesaran, M.H., 81, 98
- PFORM** instruction, 6
 - with INDIV/TIME options, 7
 - INPUT option, 8
- Phillips, P.C.B., 89, 92
- Pooled estimation, 19
- Precision, 184
- Precision-weighted average, 129
- PREGRESS** instruction
 - EFFECTS=BOTH option, 42
 - EFFECTS=TIME option, 42
 - HAUSMAN option, 34
 - METHOD=BETWEEN, 25
 - METHOD=FIXED, 20
 - METHOD=POOLED, 19
 - METHOD=RANDOM, 30
 - VCOMP option, 33
 - VINDIV option, 33
 - VRANDOM option, 33
- Prior
 - hierarchical, 133
- Probability distributions
 - chi-squared, 174
 - gamma, 178
 - inverse chi-squared, 175
 - inverse gamma, 179
 - inverse Wishart, 181
 - multivariate normal, 177, 183
 - normal, 176
 - Wishart, 180
- Probit model
 - random effects, 68
- PRODUC.XLS data set, 60
- PSID, 1
- PSTATS** instruction, 11, 31
- %RAN function, 176
- Random coefficients model, 131
- Random effects, 28
- Random effects probit, 68
- %RANMAT function, 176, 177
- %RANMVNORMAL function, 177
- %RANMVPOSTCMOM function, 134
- %RANWISHART function, 180
- %RANWISHARTF function, 180
- %RANWISHARTI function, 181
- Recoloring, 97
- @REGPCSE** procedure, 172
- REProbitIntegral** user function, 70
- Rosen, H., 51, 141
- Rosen, S., 7
- SET** instruction
 - NOPANEL option, 8
 - with lags, 4
- Shin, Y., 81, 98
- %SIGMA matrix, 24
- %SLIKE function, 8
- Smith, R., 98
- Smooth transition regression, 61
- Spectral decomposition, 29
- SSTATS** instruction, 9, 35
- STAR model, 61
- Stock, J., 95
- Sul, D., 10, 96
- @SWAMY** procedure, 133
- Swamy, P.A.V.B., 131, 167
- Swamy-Arora estimator, 32
- SWEEP** instruction, 10, 24
- Threshold model, 61
- time-invariant data, 8
- Time-invariant regressors, 28
- Tzavalis, E., 81
- Unbalanced panel, 3
- %VINDIV variable, 31, 34
- %VRANDOM variable, 31, 34
- %VTIME variable, 42
- Wallace, T.D., 163

- Wallace-Hussain estimator, 31, 35, 163
- Wansbeek, T., 166
- Wansbeek-Kapteyn estimator, 32, 166
- Watson, M., 95
- Wishart distribution, 180
 - inverse, 181
- Within regression, 22
- Wooldridge, J., 31, 42, 64
- Zero standard errors in output, 21
- `%ZTEST` function, 176