

A Gentle Introduction
to
Panel Data Modeling Using R

M. S. Legrand

© *Draft date August 9, 2013*

Contents

Contents	2
1 Panel Data Basics	5
1.1 What is Panel Data?	5
1.2 Panel Data in R	5
2 The General Form of the Linear Model	9
3 Loading/Saving Panel Data	11
3.1 Reading and Writing csv Panel Data Files	11
3.2 Foreign Reading and Writing	12
3.3 Reading EViews wfl files	12
3.4 Data from Multiple Sources	13
3.5 Cleaning Data	15
3.6 Panel Data Simultions	16
3.7 The Shape of Data	17
3.8 Reading and Combining Single Year Files	19
4 Exploring Panel Data	23
4.1 Basic Statistics	23
4.2 Plotting	24
4.3 Plotting Time Series	27
4.4 Heterogeneity Across Countries	28
4.5 Heterogeneity Across Years	29
5 The Pooled Model	31
6 The Between Model	33
6.0.1 The Between Estimator	34
7 Fixed Effects Model	39
7.1 Dummy Variable Estimator (LSDV)	39
7.1.1 Using Factors as Dummy variables	42
7.2 The Within Estimator (FE)	45
7.3 Fixed vs Pooling Models	49

7.4	FE Panel Data Generation	50
8	The Random Effects Model	51
8.1	The RE Model	51
8.2	Random Effects Estimation	53
8.3	Lagrange Multiplier Test	55
8.4	Hausman Test	56
8.5	Eliminating Additional Explanatory Variables	57
9	Instrument Variables	59
9.1	Bias Arrises From Endogeneity of X	59
9.2	Two Stage Least Squares	60
9.3	Combining the 2 Stage Regression Calculations	61
10	Guidelines for Model Selection	63
	Appendices	69
.1	Ordinary Least Square Regression (OLS)	69
.1.1	A Simple Example	69
.2	Gauss Markov Theorem	72

Chapter 1

Panel Data Basics

1.1 What is Panel Data?

Panel Data¹, is a multi-dimensional data set, consisting of repeated measurements ($\vec{X}_{i,t}$) of individuals or countries (i)) spanning over time (t). More precisely,

Definition 1: Panel Data

Panel data is a mapping from a subset of a product space $I \times T$ into another product space $\Pi_1^k M_k$ called the measurements. I represents the set of individuals (states, or countries) and the T represent observation times. When the domain of the panel data is equal to $I \times T$ we say the panel data is *balanced*. Panel data is said to be *unbalanced* provided it is not balanced.

Thus each panel data measurement consists of K many values associated with an individual and a time t . For example consider table 1.1

Here, each row represents a single observation together its associated index: $(i, t) = (year, country)$. That is, the observation times appear in the first column, the individuals in the second column. Thus $T = (2010, 2011, 2012)$, $I = (A, B, C)$, Moreover, the first row is interpreted as $t = 2010$, $i = A$, $X_{i,t} = (-144.70, 0.77, 36.48)$ Since the observation times for each country are the same, this panel is *balanced*.

1.2 Panel Data in R

Panel data in R is represented as a *data.frame* object. A *data.frame* object is similar to a matrix having named columns but unlike matrix, the types of columns may differ. Thus one column may be an integer while another may be a string. In R, we can inspect the first 6 lines of our panel (data frame) using the *head* command

¹ Also known as longitudinal data or repeated measures

	year	country	y	x1	x2
1	2010	A	-144.70	0.77	156.45
2	2011	A	14.78	36.48	78.40
3	2012	A	194.56	89.71	15.04
4	2010	A	-139.39	1.17	181.94
5	2011	B	24.81	38.47	62.52
6	2012	B	-52.23	44.96	162.54
7	2010	B	136.50	87.94	69.78
8	2011	B	47.66	95.48	183.70
9	2012	C	122.45	77.08	42.30
10	2010	C	136.39	91.03	66.27
11	2011	C	38.85	97.39	186.54
12	2012	C	-87.00	28.60	184.82

Table 1.1: Panel Data

```
head(panel)
```

```
##   year country      y      x1      x2
## 1 2010      A -144.70  0.7713 156.45
## 2 2011      A   14.78 36.4832  78.40
## 3 2012      A  194.56 89.7059  15.04
## 4 2010      A -139.39  1.1708 181.94
## 5 2011      B   24.81 38.4662  62.52
## 6 2012      B  -52.23 44.9631 162.54
```

To inspect the entire panel, issue `print(panel)` command.

```
print(panel)
```

```
##   year country      y      x1      x2
## 1 2010      A -144.70  0.7713 156.45
## 2 2011      A   14.78 36.4832  78.40
## 3 2012      A  194.56 89.7059  15.04
## 4 2010      A -139.39  1.1708 181.94
## 5 2011      B   24.81 38.4662  62.52
## 6 2012      B  -52.23 44.9631 162.54
## 7 2010      B  136.50 87.9412  69.78
## 8 2011      B   47.66 95.4751 183.70
## 9 2012      C  122.45 77.0806  42.30
## 10 2010      C  136.39 91.0296  66.27
## 11 2011      C   38.85 97.3930 186.54
## 12 2012      C  -87.00 28.6029 184.82
```

The format we use in R to represent panel data is called the *long format*. This format consists one column representing the time, one column representing the indi-

viduals, and remaining columns representing the data. Thus one row represent a single observation $X_{i,t}$.

Chapter 2

The General Form of the Linear Model

Linear modeling assumes a model of the form

$$y_{i,t} = \vec{\beta}^T x_{i,t} + Error_{i,t} \quad (2.1)$$

which is traditionally calibrated using linear regression. The error term $Error_{i,t}$ is a random term which may be decomposed into three components: individual specific effects: δ_i , period specific effects γ_t and residuals $\epsilon_{i,t}$. The δ_i accommodates for heterogeneity across individuals, and the γ_t accommodates for heterogeneity across time.

The general form of linear model for panel data is:

The diagram shows the equation $Y_{i,t} = \alpha + X_{i,t}^T \beta + \delta_i + \gamma_t + \epsilon_{i,t}$ with various terms highlighted in colored boxes. Annotations with arrows point from descriptive text to these terms:

- Explanatory Variable (points to $X_{i,t}^T$)
- Dependent Variable (points to $Y_{i,t}$)
- global constant (points to α)
- explanatory coefficients (points to β)
- cross sectional effect (points to δ_i)
- period effect (points to γ_t)
- residual (points to $\epsilon_{i,t}$)

$Y_{i,t} = \alpha + X_{i,t}^T \beta + \delta_i + \gamma_t + \epsilon_{i,t} \quad (2.2)$

With the assumptions:

- The residual $\epsilon_{i,t}$ satisfies $\epsilon_{i,t} \sim IDD(0, \sigma_\epsilon^2)$

- δ_i is uncorrelated with $\epsilon_{i,t}$, that is $\rho(\delta_i, \epsilon_{i,t}) = 0$
- γ_i is uncorrelated with $\epsilon_{i,t}$, that is $\rho(\gamma_i, \epsilon_{i,t}) = 0$
- $X_{i,t}$ is uncorrelated with $\epsilon_{i,t}$, that is $\rho(X_{i,t}, \epsilon_{i,t}) = 0$

By varying the assumptions on the individual parts, we obtain several models, the simplest being the *pooled model*, where we assume the cross sectional and period effects to be null.

$$y_{i,t} = \alpha + x_{i,t}^T \beta + \epsilon_{i,t} \quad (2.3)$$

By allowing individual cross sectional effects two prominent models can be obtained: the *Fixed Effects Model* and the *Random Effects Model*

These models are distinguished by whether or not the cross-sectional effects are correlated with the explanatory variables $X_{i,t}$. See table 2 below. The Fixed Effects model and the Random Effects Model are discussed in chapter 7 and chapter 8 respectively.

Table 2.1: Fixed vs Random Effect

	Fixed Effects	Random Effects
Correlation	$\rho(X_{i,t}, \delta_i) \neq 0$	$\rho(X_{i,t}, \delta_i) = 0$
Form	$y_{i,t} = (\alpha + \delta_i) + X_{i,t}^T + \epsilon_{i,t}$	$y_{i,t} = \alpha + X_{i,t}^T + (\delta_i + \epsilon_{i,t})$
Intercepts	varies across individuals (or time)	Constant
Error Variances	Constant	Varies across individuals
Slope	constant	constant
Test	Incremental F	Breusch-Pagen LM Test
Estimator	LSDV or Within	GLS, FGLS

Chapter 3

Loading/Saving Panel Data

The simplest example of reading and writing is that of a plain text file.

3.1 Reading and Writing csv Panel Data Files

A csv file is a plain text file that uses comma's to separate fields, with each data entry on a separate line. For example, it may look like

```
"year", "country", "y", "x1", "x2"
2010, "A", 50.3429206006974, 44.062630017288, 48.0893397703767
2011, "A", 114.641023960132, 48.4834742499515, 2.6708984747529
2012, "A", -98.9665030501783, 13.5961908847094, 156.396966427565
2010, "B", -36.9658716348, 34.5664419233799, 116.402363497764
2011, "B", 155.295395891704, 67.5066618714482, 0.21973354741931
2012, "B", 99.3606661572028, 50.4071495495737, 31.8780469708145
2010, "C", -150.150092165442, 13.8569770613685, 188.616755511612
2011, "C", 115.937315194709, 70.0987725518644, 45.0018198695034
2012, "C", -87.6802043890115, 33.1616186769679, 184.856367809698
```

Then to read this file, we simply issue the *read.csv* command:

```
df <- read.csv("./Data/country.csv")
head(df)
```

##	year	country	y	x1	x2
## 1	2010	A	50.34	44.06	48.0893
## 2	2011	A	114.64	48.48	2.6709
## 3	2012	A	-98.97	13.60	156.3970
## 4	2010	B	-36.97	34.57	116.4024
## 5	2011	B	155.30	67.51	0.2197
## 6	2012	B	99.36	50.41	31.8780

As usual, we use the *head* command to display the first 6 lines

To write the panel data back to a text file named "*temp.csv*" issue the *write.csv* command.

```
write.csv(df, "./Data/temp.csv", row.names = FALSE)
```

Note: we set *row.names=false* to prevent writing the row names.

Both *read.csv* and *write.csv* are special cases of *write.table* from the package *Utils*. For greater detail issue the commands *help(read.table)* and *help(write.table)*

3.2 Foreign Reading and Writing

Using the *foreign* package, we can read and write in many different formats: Stata, SAS, SPSS, Systat, ... For example, we can load stata files as follows:

```
library(foreign)
paneldata <- read.dta("http://dss.princeton.edu/training/Panel101.dta")
head(paneldata)
```

##	country	year	y	y_bin	x1	x2	x3
## 1	F	1990	1.343e+09	1	-0.56757	-0.3467	1.25842
## 2	B	1990	-5.935e+09	0	-0.08185	1.4251	0.02343
## 3	E	1990	1.343e+09	1	0.45287	1.7284	0.59706
## 4	D	1990	1.883e+09	1	-0.31391	1.7367	0.64664
## 5	G	1990	1.343e+09	1	0.94488	-1.5150	1.45266
## 6	C	1990	-1.292e+09	0	1.31256	-1.2931	0.20408

First the package *foreign* is imported using *library(foreign)* command. Then the panel is read in using the *read.dta* command. Finally we display the first 6 lines of the panel data using the *head(Panel)* command. To write back the stata file, we use the *write.dta* command. For more information, type *help(package=foreign)*

3.3 Reading EViews wf1 files

Similarly eviews files can be loaded, but using a different package, *hexView*. For example:

```
library(hexView)
# download from
# 'http://www.principlesofeconometrics.com/eviews/bond.wf1')
Panel <- readEViews("./Data/nls_panel.wf1")

## Skipping boilerplate variable
## Skipping boilerplate variable
```

```
head(Panel)
```

```
##      AGE BLACK C_CITY COLLGRAD DATEID EDUC  EXPER  EXPER2 HOURS ID  LWAGE  MSP
## 1   30     1      1         0 723545   12   7.667   58.78    38  1  1.808    1
## 2   31     1      1         0 723910   12   8.583   73.67    38  1  1.863    1
## 3   33     1      1         0 724641   12  10.179  103.62    38  1  1.789    0
## 4   35     1      1         0 725371   12  12.179  148.34    40  1  1.847    0
## 5   37     1      1         0 725736   12  13.622  185.55    40  1  1.856    0
## 6   36     0      0         1 723545   17   7.577   57.41    48  2  1.281    1
##      NEV_MAR NOT_SMSA SOUTH  TENURE  TENURE2  UNION  YEAR
## 1           0         0      0   7.667   58.778      1    82
## 2           0         0      0   8.583   73.674      1    83
## 3           0         0      0   1.833    3.361      1    85
## 4           0         0      0   3.750   14.062      1    87
## 5           0         0      0   5.250   27.562      1    88
## 6           0         0      0   2.417    5.840      0    82
```

For more information type `help(hexView)`

3.4 Data from Multiple Sources

Sometimes our data comes from multiple sources and so to do our analysis, we may need to combine them. This can be accomplished by *merging* the corresponding data.frames. For example: Consider the following pair of data sets:

```
panel1 <- read.csv("../Data/panel1.csv")
print(panel1)

##   year country      y      x1      x2
## 1 2010        A -4.284 25.608 66.03
## 2 2011        A 47.677 72.424 118.61
## 3 2012        A -48.110 38.688 154.70
## 4 2010        B -104.150 40.096 192.87
## 5 2011        B 59.659 28.249 17.18
## 6 2012        B -80.145 5.726 121.77
## 7 2010        C 123.112 90.130 65.75
## 8 2011        C 93.281 51.376 30.86
## 9 2012        C 15.322 16.322 47.06
```

```
panel2 <- read.csv("../Data/panel2.csv")
print(panel2)

##   year country      z1      z2
## 1 2010        A 64.63 56.369
## 2 2011        A 22.32 106.185
```

```
## 3 2012      A 28.59    4.234
## 4 2010      B 31.90    35.882
## 5 2011      B 60.47    43.064
## 6 2012      B 15.47  111.384
## 7 2010      C 90.17    46.027
## 8 2011      C 52.82  176.952
## 9 2012      C 93.38  164.077
```

Note: The countries and dates of `panel1` and `panel2` match, but the columns do not.

What we want to do is to combine these into a single panel. This may be accomplished by the `merge` command as follows:

```
panelM <- merge(panel1, panel2, by = c("year", "country"))
print(panelM)
```

##	year	country	y	x1	x2	z1	z2
## 1	2010	A	-4.284	25.608	66.03	64.63	56.369
## 2	2010	B	-104.150	40.096	192.87	31.90	35.882
## 3	2010	C	123.112	90.130	65.75	90.17	46.027
## 4	2011	A	47.677	72.424	118.61	22.32	106.185
## 5	2011	B	59.659	28.249	17.18	60.47	43.064
## 6	2011	C	93.281	51.376	30.86	52.82	176.952
## 7	2012	A	-48.110	38.688	154.70	28.59	4.234
## 8	2012	B	-80.145	5.726	121.77	15.47	111.384
## 9	2012	C	15.322	16.322	47.06	93.38	164.077

We may still use `merge` when the year and country don't align, as shown in the next example.

```
panel3 <- read.csv("../Data/panel3.csv")
print(panel3)
```

##	year	country	z1	z2
## 1	2011	A	64.63	56.369
## 2	2012	A	22.32	106.185
## 3	2013	A	28.59	4.234
## 4	2011	B	31.90	35.882
## 5	2012	B	60.47	43.064
## 6	2013	B	15.47	111.384
## 7	2011	C	90.17	46.027
## 8	2012	C	52.82	176.952
## 9	2013	C	93.38	164.077

Note: *Panel3* begins at 2011 and ends at 2013. Thus when merging, we get

```
panelM <- merge(panel1, panel3, by = c("year", "country"))
print(panelM)
```

##	year	country	y	x1	x2	z1	z2
## 1	2011	A	47.68	72.424	118.61	64.63	56.37
## 2	2011	B	59.66	28.249	17.18	31.90	35.88
## 3	2011	C	93.28	51.376	30.86	90.17	46.03
## 4	2012	A	-48.11	38.688	154.70	22.32	106.18
## 5	2012	B	-80.15	5.726	121.77	60.47	43.06
## 6	2012	C	15.32	16.322	47.06	52.82	176.95

By default, the rows for 2010 and 2013 are omitted since *Panel1* is missing 2013 and *Panel3* is missing 2010. To have both rows included in our results, we simply add the options `all.x=TRUE` and `all.y=TRUE`

```
panelM <- merge(panel1, panel3, by = c("year", "country"), all.x = TRUE,
               all.y = TRUE)
print(panelM)
```

##	year	country	y	x1	x2	z1	z2
## 1	2010	A	-4.284	25.608	66.03	NA	NA
## 2	2010	B	-104.150	40.096	192.87	NA	NA
## 3	2010	C	123.112	90.130	65.75	NA	NA
## 4	2011	A	47.677	72.424	118.61	64.63	56.369
## 5	2011	B	59.659	28.249	17.18	31.90	35.882
## 6	2011	C	93.281	51.376	30.86	90.17	46.027
## 7	2012	A	-48.110	38.688	154.70	22.32	106.185
## 8	2012	B	-80.145	5.726	121.77	60.47	43.064
## 9	2012	C	15.322	16.322	47.06	52.82	176.952
## 10	2013	A	NA	NA	NA	28.59	4.234
## 11	2013	B	NA	NA	NA	15.47	111.384
## 12	2013	C	NA	NA	NA	93.38	164.077

3.5 Cleaning Data

Data is not always clean. For example, consider the following data:

```
panelDirty <- read.csv("../Data/pDirty.csv")
print(panelDirty)
```

##	year	country	y	x1	x2	z1	z2
## 1	2010	A	-4.284	25.608	66.03	64.63	56.37
## 2	2010	B	-104.150	40.096	192.87	31.90	35.88
## 3	2010	C	123.112	90.130	65.75	90.17	NA

```
## 4 2011      A    47.677 72.424 118.61      NA 106.18
## 5 2011      B    59.659 28.249  17.18      NA      NA
## 6 2011      C    93.281 51.376  30.86 52.82      NA
## 7 2012      A   -48.110 38.688 154.70 28.59      NA
## 8 2012      B   -80.145  5.726 121.77 15.47      NA
## 9 2012      C    15.322 16.322  47.06 93.38 164.08
```

One approach is to remove rows with NA's using *na.omit*:

```
panelClean1 <- na.omit(panelDirty)
print(panelClean1)

##   year country      y      x1      x2      z1      z2
## 1 2010      A   -4.284 25.61  66.03 64.63  56.37
## 2 2010      B -104.150 40.10 192.87 31.90  35.88
## 9 2012      C   15.322 16.32  47.06 93.38 164.08
```

If the *z2* column is not relevant to our calculations, we might first delete it by simply using *panelDirty\$z2 <- NULL*, before removing rows with NA's

```
panelDirty$z2 <- NULL
panelClean2 <- na.omit(panelDirty)
print(panelClean2)

##   year country      y      x1      x2      z1
## 1 2010      A   -4.284 25.608  66.03 64.63
## 2 2010      B -104.150 40.096 192.87 31.90
## 3 2010      C  123.112 90.130  65.75 90.17
## 6 2011      C    93.281 51.376  30.86 52.82
## 7 2012      A   -48.110 38.688 154.70 28.59
## 8 2012      B   -80.145  5.726 121.77 15.47
## 9 2012      C    15.322 16.322  47.06 93.38
```

3.6 Panel Data Simulations

Sometimes it is useful to generate some artificial panel data, to test our analysis algorithms. We can easily generate panel data in R using the built random number generators and a wonderful little function called *expand.grid*.

```
country <- c("alpha", "beta", "gamma")
years <- c(2008, 2009, 2011)
panel <- expand.grid(country = country, year = years)
n <- nrow(panel)
panel$x <- rnorm(n, mean = 10 * (1:n), sd = 3)
panel$y <- panel$x * 2 + rnorm(n, mean = 0, sd = 2)
```


The result is

```
head(panel)

##   country year      x      y
## 1   alpha 2008 10.03 19.44
## 2    beta 2008 20.17 42.23
## 3   gamma 2008 27.25 53.41
## 4   alpha 2009 37.35 74.19
## 5    beta 2009 49.53 97.33
## 6   gamma 2009 59.75 121.40
```

and can be saved using

```
write.csv(df, "./Data/simulation.csv", row.names = FALSE)
```

3.7 The Shape of Data

The data we have considered so far has consisted of rows of the form *country, year, x1, x2, ... ,y*. However sometimes our data is not in that form. For example, consider the data in table 3.1:

	state	var	2008	2009	2011
1	dc	x	11.02	40.72	68.13
2	dc	y	21.11	82.58	134.04
3	virginia	x	19.38	53.06	82.66
4	virginia	y	39.83	108.91	163.39
5	maryland	x	29.26	59.04	96.74
6	maryland	y	55.82	116.00	197.80

Table 3.1: Panel Data

Here, each row represents a time series, that columns are used to present data for different years. In order to transform this into a more usable format we transform it into the *long* data format, but first we read in the data.

```
fat.data <- read.csv("./Data/fatPanel.csv")
head(fat.data)

##      state var X2008 X2009 X2011
## 1      dc  x 11.02  40.72  68.13
## 2      dc  y 21.11  82.58 134.04
## 3 virginia x 19.38  53.06  82.66
## 4 virginia y 39.83 108.91 163.39
## 5 maryland x 29.26  59.04  96.74
## 6 maryland y 55.82 116.00 197.80
```

Note: the extra X appearing in front of the year. This is because, by default, R converts the column names using the *make.names* function to "valid" names. To prevent this from occurring, we must set the *check.names=F* option.

```
fat.data <- read.csv("./Data/fatPanel.csv", check.names = F)
head(fat.data)
```

	state	var	2008	2009	2011
## 1	dc	x	11.02	40.72	68.13
## 2	dc	y	21.11	82.58	134.04
## 3	virginia	x	19.38	53.06	82.66
## 4	virginia	y	39.83	108.91	163.39
## 5	maryland	x	29.26	59.04	96.74
## 6	maryland	y	55.82	116.00	197.80

Next we convert *fat.data* into the "long data format". This is done using the *melt* function from the *reshape* package.

```
library("reshape2")
long.data <- melt(fat.data, id.vars = c("state", "var"))
head(long.data)
```

	state	var	variable	value
## 1	dc	x	2008	11.02
## 2	dc	y	2008	21.11
## 3	virginia	x	2008	19.38
## 4	virginia	y	2008	39.83
## 5	maryland	x	2008	29.26
## 6	maryland	y	2008	55.82

We rename the column for readability

```
names(long.data)[3] <- "year"
head(long.data)
```

	state	var	year	value
## 1	dc	x	2008	11.02
## 2	dc	y	2008	21.11
## 3	virginia	x	2008	19.38
## 4	virginia	y	2008	39.83
## 5	maryland	x	2008	29.26
## 6	maryland	y	2008	55.82

And finally we use *dcast* to reshape into our more familiar format.

```

panel.data <- dcast(long.data, state + year ~ var,
  value.var = "value")
panel.data

##      state year      x      y
## 1      dc 2008 11.02 21.11
## 2      dc 2009 40.72 82.58
## 3      dc 2011 68.13 134.04
## 4 maryland 2008 29.26 55.82
## 5 maryland 2009 59.04 116.00
## 6 maryland 2011 96.74 197.80
## 7 virginia 2008 19.38 39.83
## 8 virginia 2009 53.06 108.91
## 9 virginia 2011 82.66 163.39

```

3.8 Reading and Combining Single Year Files

Occasionally, we are presented with a situation where each file represents the data for all individuals for a single given year. To show how we might handle this, we consider a simple example:

In this example all files located in a subdirectory called *Data/Yearly*. Furthermore, all file are of the *csv format* and have names of the form "*dddd.csv*" The files are

	country	x	y
1	Aland	13.47	23.67
2	Bland	11.59	25.00
3	Cland	11.42	24.65

Table 3.2: Data/Yearly/1926.csv

	country	x	y
1	Aland	11.87	25.93
2	Bland	10.20	27.94
3	Cland	11.32	22.54

Table 3.3: Data/Yearly/1927.csv

To see a listing we would normally type

```

dir("Data/Yearly")

## [1] "1926.csv" "1927.csv" "1928.csv"

```

Using this mechanism, we can collect all the files as a single R vector, called *files*

	country	x	y
1	Aland	13.37	29.05
2	Bland	11.19	27.96
3	Cland	11.37	25.91

Table 3.4: Data/Yearly/1928.csv

```
files <- dir("Data/Yearly")
```

The idea is for each file Name, *fName* in files we want to read the data.frame, *df* using *read.csv* and then combine them back into a single data.frame using some form of *plyr*. The only complication,are

- Before, recombining the data.frames back into a single data.frame we want to add a year-column containing the year corresponding to that file. So we need to extract the year from the name. That is what the function *toYear* does below.
- To read the file, we must include the path to file in addition to the file name. That is what the function *toPath* does.

Once this we have the two helper functions, all we need to do is to use *ldply*¹ from the *plyr* package.

The code is actually quite short!

```
library("plyr")
files <- dir("Data/Yearly/")
toYear <- function(fName) {
  as.integer(substr(fName, 1, 4))
}
toPath <- function(fName) {
  paste("Data/Yearly", fName, sep = "/")
}
panel.data <- ldply(files, function(fName) {
  df <- read.csv(toPath(fName))
  df$year <- toYear(fName)
  df
})
panel.data

##   country      x      y year
## 1   Aland 13.47 23.67 1926
## 2   Bland 11.59 25.00 1926
## 3   Cland 11.42 24.65 1926
## 4   Aland 11.87 25.93 1927
## 5   Bland 10.20 27.94 1927
```

¹The ld of *ldply* stands for apply to a list to produce a data.frame.

##	6	Cland	11.32	22.54	1927
##	7	Aland	13.37	29.05	1928
##	8	Bland	11.19	27.96	1928
##	9	Cland	11.37	25.91	1928

Chapter 4

Exploring Panel Data

4.1 Basic Statistics

The `summary` command provides a brief summary of the statistics of our panel data as seen below.

```
library(foreign)
panel <- read.dta("http://dss.princeton.edu/training/Panel101.dta")
panel$y_bin <- NULL
head(panel)
```

##	country	year	y	x1	x2	x3
## 1	F	1990	1.343e+09	-0.56757	-0.3467	1.25842
## 2	B	1990	-5.935e+09	-0.08185	1.4251	0.02343
## 3	E	1990	1.343e+09	0.45287	1.7284	0.59706
## 4	D	1990	1.883e+09	-0.31391	1.7367	0.64664
## 5	G	1990	1.343e+09	0.94488	-1.5150	1.45266
## 6	C	1990	-1.292e+09	1.31256	-1.2931	0.20408

```
summary(panel)
```

##	country	year	y	x1
## A:10	Min.	:1990	Min. :-7.86e+09	Min. :-0.568
## B:10	1st Qu.:	:1992	1st Qu.: 2.47e+08	1st Qu.: 0.329
## C:10	Median :	:1994	Median : 1.90e+09	Median : 0.641
## D:10	Mean :	:1994	Mean : 1.85e+09	Mean : 0.648
## E:10	3rd Qu.:	:1997	3rd Qu.: 3.37e+09	3rd Qu.: 1.096
## F:10	Max. :	:1999	Max. : 8.94e+09	Max. : 1.446
## G:10				
##	x2		x3	
##	Min. :-1.622		Min. :-1.165	
##	1st Qu.:-1.216		1st Qu.: -0.079	

```
## Median :-0.462 Median : 0.514
## Mean : 0.134 Mean : 0.762
## 3rd Qu.: 1.608 3rd Qu.: 1.155
## Max. : 2.530 Max. : 7.169
##
```

However, these statistics tell only part of the story, in particular we might be interested in the mean of Y on a per country basis.

That is,

```
aggregate(panel$y, by = list(panel$country), mean)

## Group.1      x
## 1      A 1.728e+09
## 2      B 2.139e+08
## 3      C 1.345e+09
## 4      D 3.641e+09
## 5      E 7.231e+08
## 6      F 3.559e+09
## 7      G 1.706e+09
```

Or we can find them all at once using the *plyr* package.

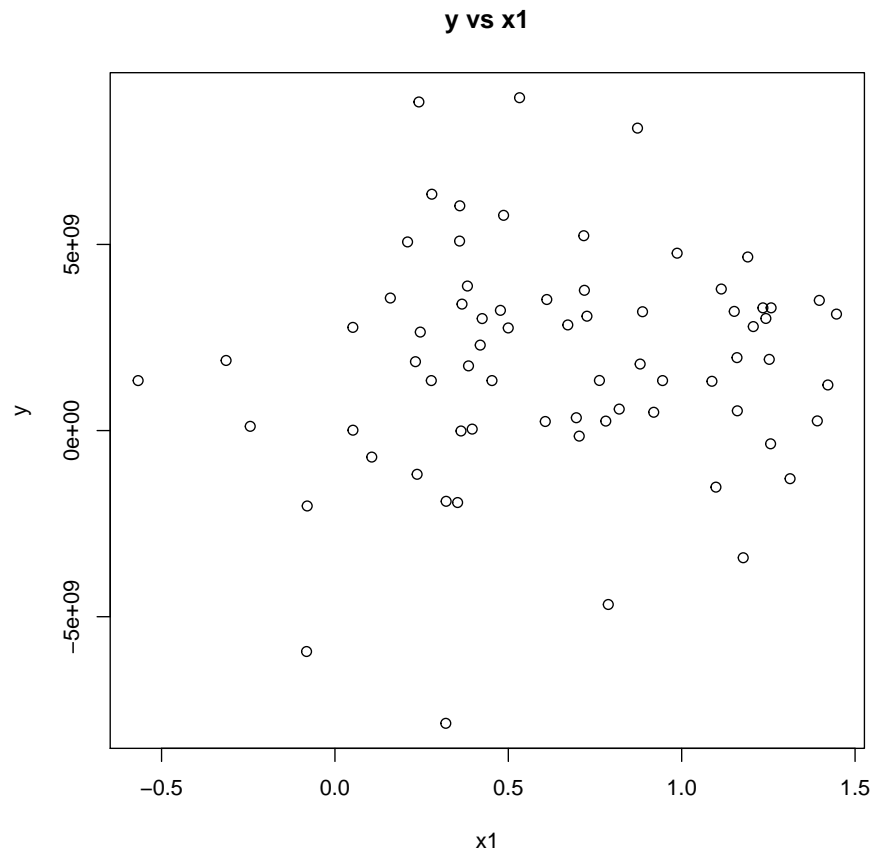
```
library(plyr)
ddply(panel, .(country), function(x) c(meanY = mean(x$y),
    meanX1 = mean(x$x1), meanX2 = mean(x$x2), meanX3 = mean(x$x3),
    varY = var(x$y)))

## country      meanY meanX1 meanX2 meanX3      varY
## 1      A 1.728e+09 0.3424 -0.8013 0.92101 3.206e+18
## 2      B 2.139e+08 0.5137 1.5723 -0.03205 8.403e+18
## 3      C 1.345e+09 1.2391 -1.2608 0.52809 6.361e+18
## 4      D 3.641e+09 0.1930 1.7191 0.18357 1.033e+19
## 5      E 7.231e+08 0.5355 1.6841 0.04465 1.186e+19
## 6      F 3.559e+09 0.6255 -0.5164 0.34887 1.569e+19
## 7      G 1.706e+09 1.0869 -1.4600 3.33880 2.323e+18
```

4.2 Plotting

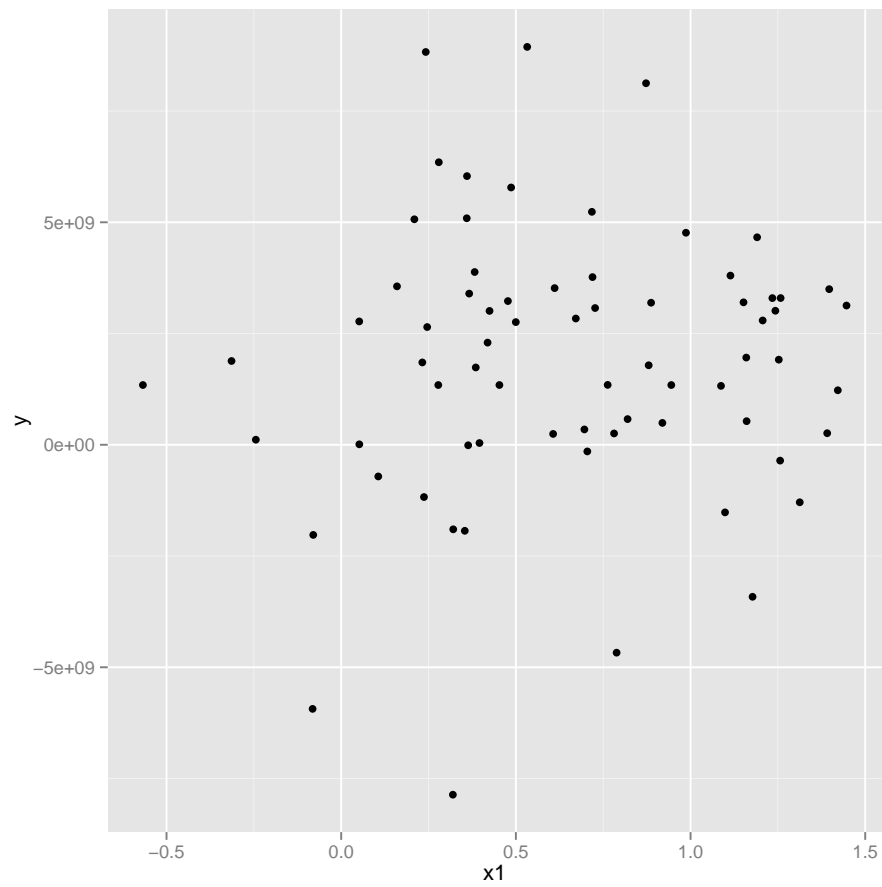
R contains a built in plotting commands *Plot*

```
plot(y ~ x1, data = panel, main = "y vs x1")
```

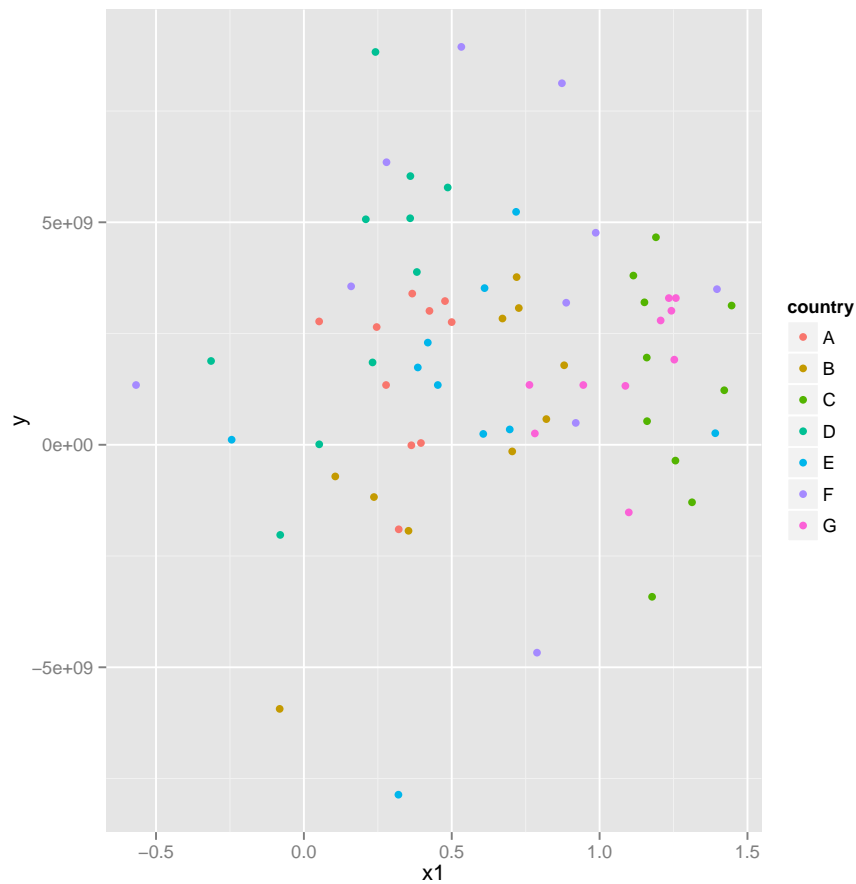
However the package *ggplot* provides additional features and so we will be using *ggplot*.

```
library(ggplot2)
ggplot(panel) + geom_point(aes(x = x1, y = y))
```



Finer detail may be obtained coloring each plot according to the country.

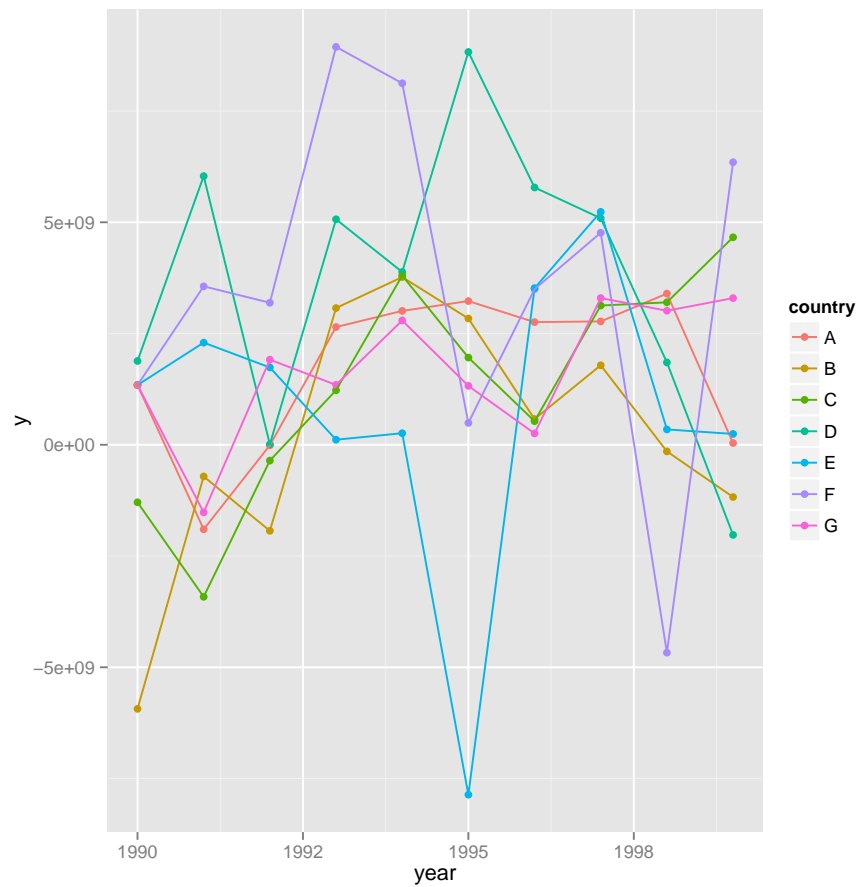
```
library(ggplot2)
ggplot(panel) + geom_point(aes(x = x1, y = y, group = country,
  colour = country))
```



4.3 Plotting Time Series

To plot y as a time series, replace the role of $x1$ by $year$ in the above and add lines to join the points between consecutive time intervals for each country.

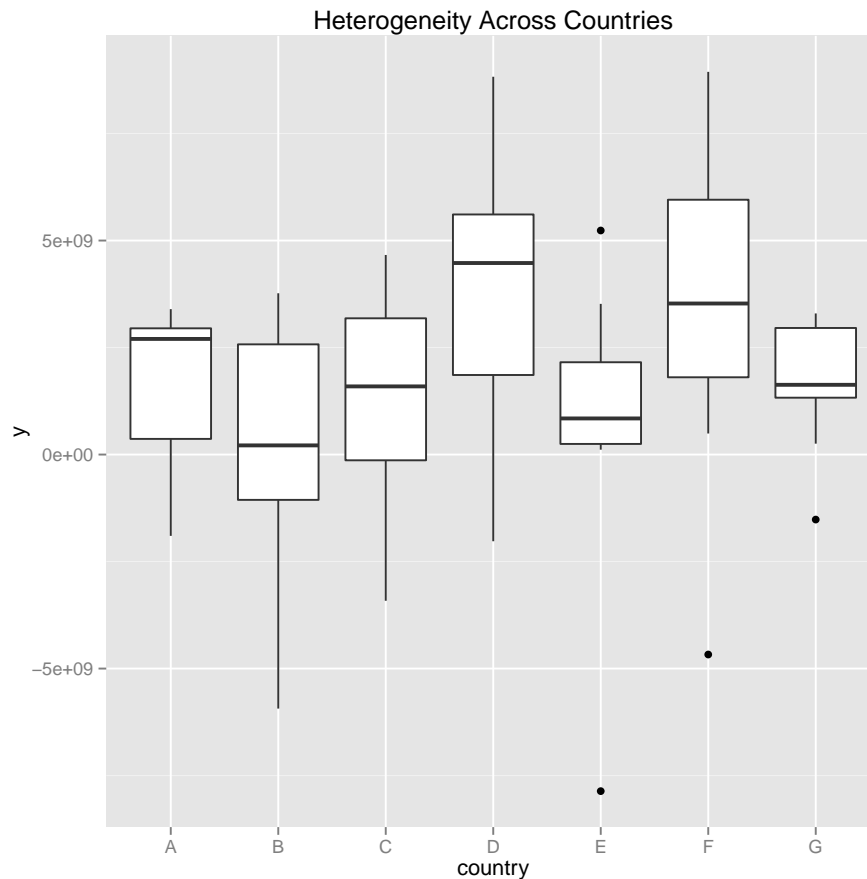
```
library(ggplot2)
ggplot(panel) + geom_point(aes(x = year, y = y, group = country,
  colour = country)) + geom_line(aes(x = year, y = y,
  group = country, colour = country))
```



4.4 Heterogeneity Across Countries

We can demonstrate the Heterogeneity across countries using boxplot as follows.

```
p <- ggplot(panel, aes(country, y))
p <- p + geom_boxplot()
p + labs(title = "Heterogeneity Across Countries")
```



The parameters of each part of a boxplot are determined by various statistics. The middle bar is the 50% percentile, the bottom and top of the box are the 25% and 75% percentiles, etc.

4.5 Heterogeneity Across Years

We can also observe the Heterogeneity across years with the following stat plot

```
p <- ggplot(panel, aes(year, y))
# p + stat_smooth(geom =
# 'point')+stat_smooth(geom = 'errorbar')
p <- p + geom_point() + stat_smooth(level = 0.95)
p + labs(title = "Heterogeneity Across Years")

## geom_smooth: method="auto" and size of largest group is
<1000, so using loess. Use 'method = x' to change the smoothing
method.
```

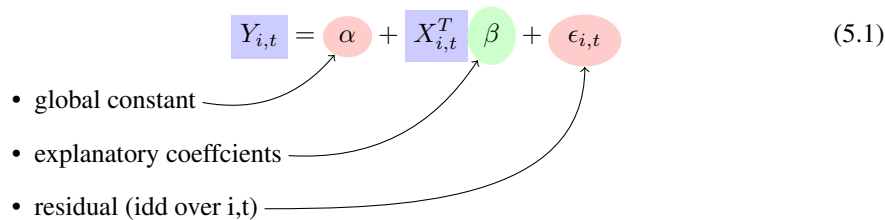


Here the shade area corresponds to a 95% confidence level.

Chapter 5

The Pooled Model

The most restrictive model is the *pooled model*. It assumes constant coefficients and that $\delta_i = 0$ and $\gamma_t = 0$. Thus we have

$$Y_{i,t} = \alpha + X_{i,t}^T \beta + \epsilon_{i,t} \quad (5.1)$$


- global constant — points to α
- explanatory coefficients — points to β
- residual (idd over i,t) — points to $\epsilon_{i,t}$

where, as in eqn 2.2, we assume:

- The residual satisfies $\epsilon_{i,t}$ satisfies $E[\epsilon_{i,t}] = 0$
- $X_{i,t}$ is uncorrelated with $\epsilon_{i,t}$, that is $\rho(X_{i,t}, \epsilon_{i,t}) = 0$

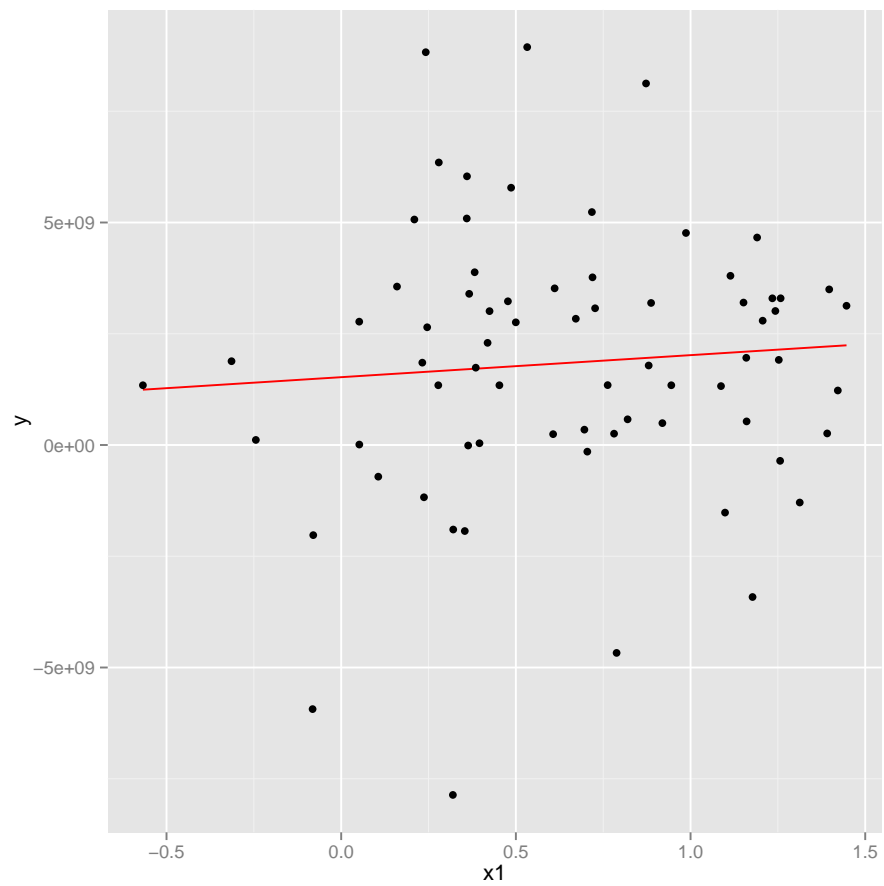
In the preceding example, we did not have a time component, however we can still apply OLS by pooling all the observations together and then perform a least squares regression. Again, the form for the general equation for pooled panel data becomes

$$y_{i,t} = \alpha + X_{i,t}^T \vec{\beta} + \epsilon_{i,t} \quad (5.2)$$

We can solve this in exactly the same way as in the preceding section, or alternatively, we may fit and plot in a single step

```
library(foreign)
panelData <- read.dta("http://dss.princeton.edu/training/Panel101.dta")
p <- ggplot(data = panelData, aes(x = x1,
  y = y))
p <- p + geom_smooth(method = "lm",
  se = FALSE, color = "red",
```

```
formula = y ~ x)  
p <- p + geom_point()  
p
```



Chapter 6

The Between Model

The *Between Model* uses just the cross-section variation to estimate the value of β by averaging across time. Thus the between model has the form:

$$\begin{aligned} & \bullet \frac{1}{T} \sum_t X_{i,t} \\ & \bullet \frac{1}{T} \sum_t Y_{i,t} \\ & \bar{Y}_{i,*} = \alpha + \bar{X}_{i,*}^T \beta + \eta_i \end{aligned} \quad (6.1)$$

• global constant
 • explanatory coefficients
 • residual

Note: In this model, t has been "averaged" out. Thus we used η_i rather than the usual $\epsilon_{i,t}$. Moreover, if we introduce an individual effect, δ_i , then we are averaging t over

$$Y_{i,t} = \alpha + X_{i,t}^T \beta + \delta_i + \epsilon_{i,t} \quad (6.2)$$

which produces a residual of

$$\eta_i = \delta_i + \frac{1}{T} \sum_t \epsilon_{i,t} = \delta_i + \bar{\epsilon}_{i,*} \quad (6.3)$$

The *between* estimator is the ordinary least squares estimator of the regression of \bar{Y}_i on an intercept α and \bar{x}_i . Hence, the *between* estimator is consistent if the regressors \bar{X}_i are independent of the composite error $\delta_i + \bar{\epsilon}_i$.

6.0.1 The Between Estimator

To perform *between* estimation we use the *plm* package, as follows:

```
library(foreign)
panelData <- read.dta("http://dss.princeton.edu/training/Panel101.dta")
library(plm)

## Loading required package: bdsmatrix
##
## Attaching package: 'bdsmatrix'
## The following object is masked from 'package:base':
##
##      backsolve
## Loading required package: nlme
## Loading required package: Formula
## Loading required package: MASS
## Loading required package: sandwich
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following object is masked from 'package:base':
##
##      as.Date, as.Date.numeric

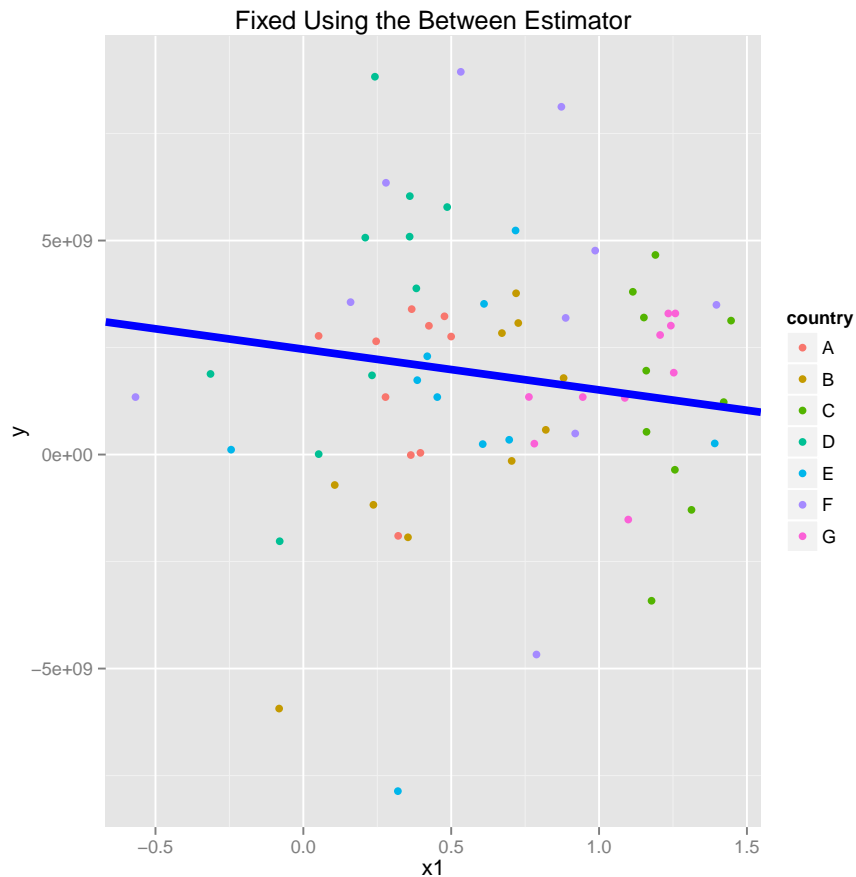
fixed.between <- plm(y ~ x1, data = panelData, index = c("country",
  "year"), model = "between")
summary(fixed.between)

## Oneway (individual) effect Between Model
##
## Call:
## plm(formula = y ~ x1, data = panelData, model = "between", index = c("cou
##      "year"))
##
## Balanced Panel: n=7, T=10, N=70
##
## Residuals :
##      Min.    1st Qu.    Median      Mean    3rd Qu.      Max.
## -1.76e+09 -8.18e+08  6.21e+07  0.00e+00  8.20e+08  1.69e+09
##
## Coefficients :
##              Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  2.46e+09   1.09e+09   2.25    0.074 .
## x1          -9.52e+08   1.48e+09  -0.64    0.549
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Total Sum of Squares:    1.04e+19
## Residual Sum of Squares: 9.57e+18
## R-Squared      :    0.0763
##      Adj. R-Squared :    0.0545
## F-statistic: 0.413259 on 1 and 5 DF, p-value: 0.549
```

Note there this model has but a single intercept. Plotting it is simple:

```
alpha <- fixed.between$coefficients["(Intercept)"]
beta <- fixed.between$coefficients["x1"]
library(ggplot2)
p <- ggplot(panelData)
p <- p + geom_point(aes(x = x1, y = y, group = country,
  colour = country))
p <- p + geom_abline(intercept = alpha, slope = beta,
  size = 2, colour = "blue")
p <- p + ggtitle("Fixed Using the Between Estimator")
p
```



To place the ols on this graph we can also use *plm*, but with the "pooling" model:

```
alpha.fixed <- fixed.between$coefficients["(Intercept)"]
beta.fixed <- fixed.between$coefficients["x1"]
pooled <- plm(y ~ x1, data = panelData, index = c("country",
  "year"), model = "pooling")
alpha.pooled <- pooled$coefficients["(Intercept)"]
beta.pooled <- pooled$coefficients["x1"]
library(ggplot2)
p <- ggplot(panelData)
p <- p + geom_point(aes(x = x1, y = y, group = country,
  colour = country))
p <- p + geom_abline(intercept = alpha.fixed, slope = beta.fixed,
  size = 2, colour = "blue")
p <- p + geom_abline(intercept = alpha.pooled, slope = beta.pooled,
  size = 1, colour = "red")
p <- p + ggtitle("Comparison of Pooling and Between")
```

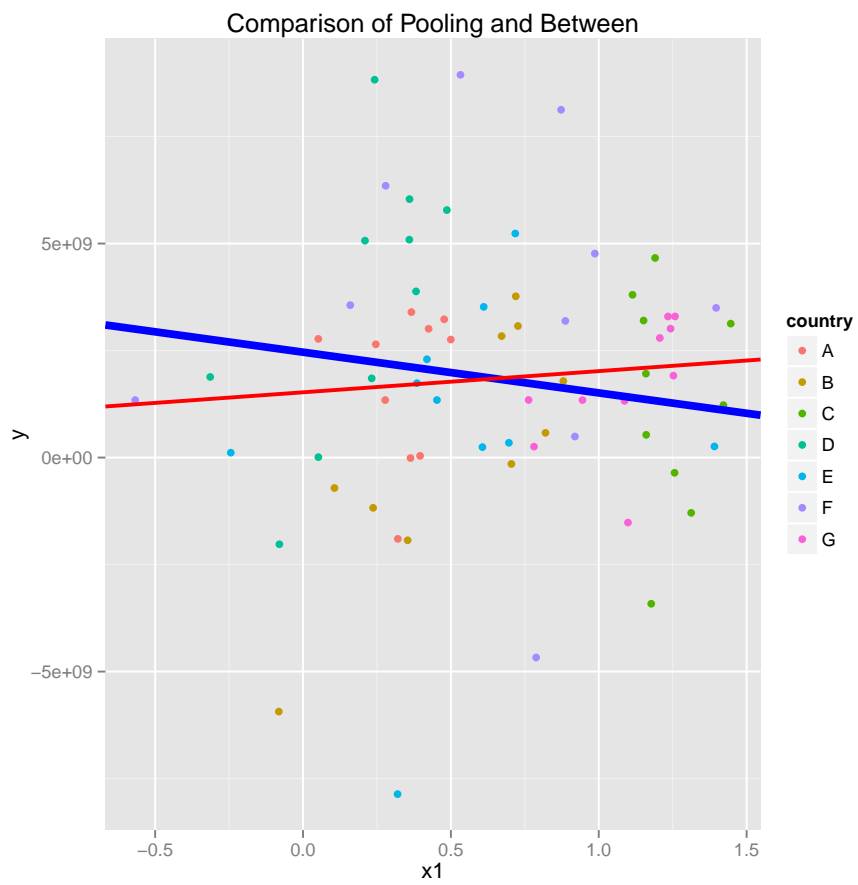


Figure 6.1: Graph with both Between and Pooled

Chapter 7

Fixed Effects Model

In the Fixed Effects Model may be obtained from the general model under the assumptions that γ_t is zero and δ_i is correlated with $X_{i,t}$. Setting $\alpha_i = \alpha + \delta_i$, we have

$$Y_{i,t} = \alpha_i + X_{i,t}^T \beta + \epsilon_{i,t} \quad (7.1)$$

- Correlated: $\rho[\alpha_i, x_{i,t}] \neq 0$
- explanatory coefficients
- residual

We may interpret each α_i as the intercept for that specific individual.

$$y_{i,t} = \alpha + x_{i,t}^T \beta + \epsilon_{i,t} \quad (7.2)$$

In this chapter, we discuss two approaches to produce estimates of the model coefficients: *Dummy Variable Estimator* § 7.1, and the *Within Estimator* § 7.2.

7.1 Dummy Variable Estimator (LSDV)

The Dummy Variable Estimator approach is to extend X by introducing dummy variables for each individual i , indicating whether that individual belongs to that group. This approach works well only when the number of time observations per individual is much larger than the number of individuals in the panel.

To illustrate this consider the following simple example: Suppose that our panel data is given by table 7.1

To find our solution, we begin by adding dummy variables d_1, d_2 to get table 7.2

Next we find the solution to

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\vec{Y} - M\vec{\theta}\| \quad (7.3)$$

	country	year	y	x
1	A	2009	1.00	1
2	A	2010	4.00	2
3	B	2009	5.00	3
4	B	2010	7.00	4

Table 7.1: panel data

	country	year	y	x	d1	d2
1	A	2009	1.00	1	1.00	0.00
2	A	2010	4.00	2	1.00	0.00
3	B	2009	5.00	3	0.00	1.00
4	B	2010	7.00	4	0.00	1.00

Table 7.2: panel data with dummy variables

where

$$\vec{Y} = \begin{bmatrix} 1 \\ 4 \\ 5 \\ 7 \end{bmatrix}, M = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \\ 4 & 0 & 1 \end{bmatrix}, \vec{\theta} = \begin{bmatrix} m \\ b_1 \\ b_2 \end{bmatrix} \quad (7.4)$$

To illustrate this example, consider the following R code:

```
df <- data.frame(country = c("A", "A", "B", "B"), year = 2009:2010,
  y = c(1, 4, 5, 7), x = 1:4)
head(df)

##   country year y x
## 1      A 2009 1 1
## 2      A 2010 4 2
## 3      B 2009 5 3
## 4      B 2010 7 4
```

Creating dummy variables we have

```
df$d1 <- c(1, 1, 0, 0)
df$d2 <- c(0, 0, 1, 1)
head(df)

##   country year y x d1 d2
## 1      A 2009 1 1  1  0
## 2      A 2010 4 2  1  0
## 3      B 2009 5 3  0  1
## 4      B 2010 7 4  0  1
```

Finally solving we have


```
fixed.dum <- lm(y ~ x + d1 + d2 - 1, data = df)
```

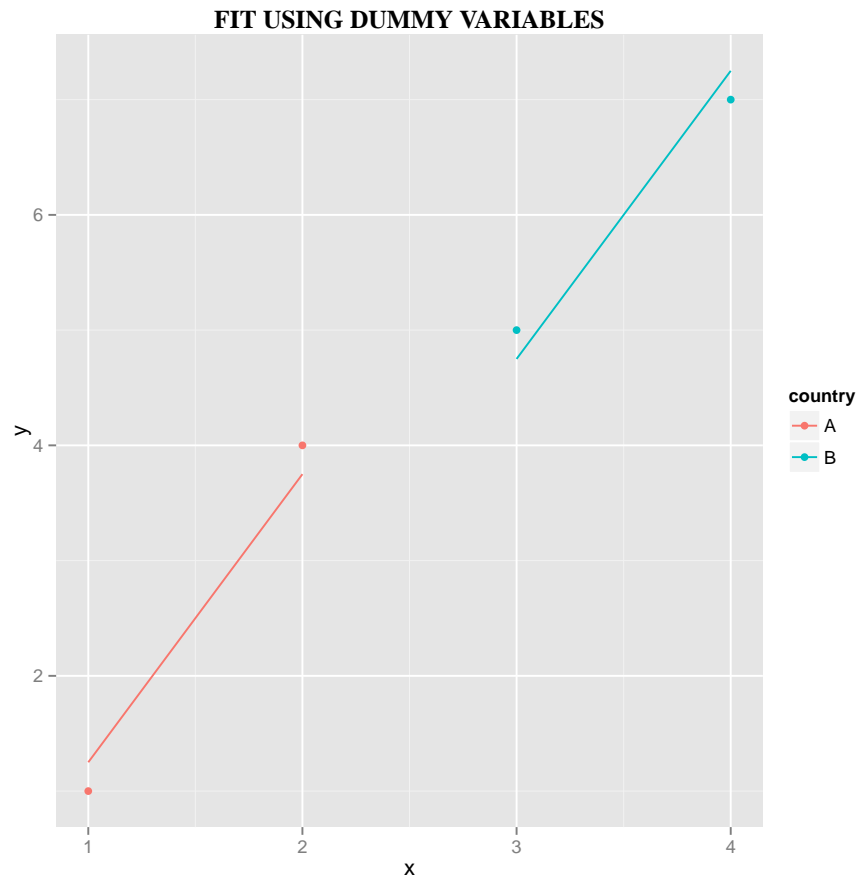
Note: The -1 in " $fixed.dum \leftarrow lm(y \sim x + d1 + d2 - 1, data = df)$ " is to tell the *lm* not to add a column of one's (which is the default when doing ordinary regression with a single intercept.)

```
summary(fixed.dum)

##
## Call:
## lm(formula = y ~ x + d1 + d2 - 1, data = df)
##
## Residuals:
##      1      2      3      4
## -0.25  0.25  0.25 -0.25
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## x          2.500      0.500   5.00   0.13
## d1         -1.250      0.829  -1.51   0.37
## d2         -2.750      1.785  -1.54   0.37
##
## Residual standard error: 0.5 on 1 degrees of freedom
## Multiple R-squared:  0.997, Adjusted R-squared:  0.989
## F-statistic: 121 on 3 and 1 DF, p-value: 0.0667
```

This is easily plotted as follows

```
yhat <- fixed.dum$fitted
library(ggplot2)
# plot fitted dummy
p <- ggplot(df)
p <- p + geom_point(aes(x = x, y = y, group = country,
  colour = country))
p <- p + geom_line(aes(x = x, y = yhat, group = country,
  colour = country))
p <- p + ggtitle("FIT USING DUMMY VARIABLES") + theme(plot.title = element_text(lineh
  face = "bold", family = "Times"))
p
```



7.1.1 Using Factors as Dummy variables

In practice, the dummy variables need not be insert, since we can use the country names as factors. We illustrate this in the following

we use the previous panel data, *panel*. This process for the panel data *panel* is implemented as follows:

```
fixed.dum <- lm(y ~ x1 + factor(country) - 1, data = panel)
summary(fixed.dum)

##
## Call:
## lm(formula = y ~ x1 + factor(country) - 1, data = panel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

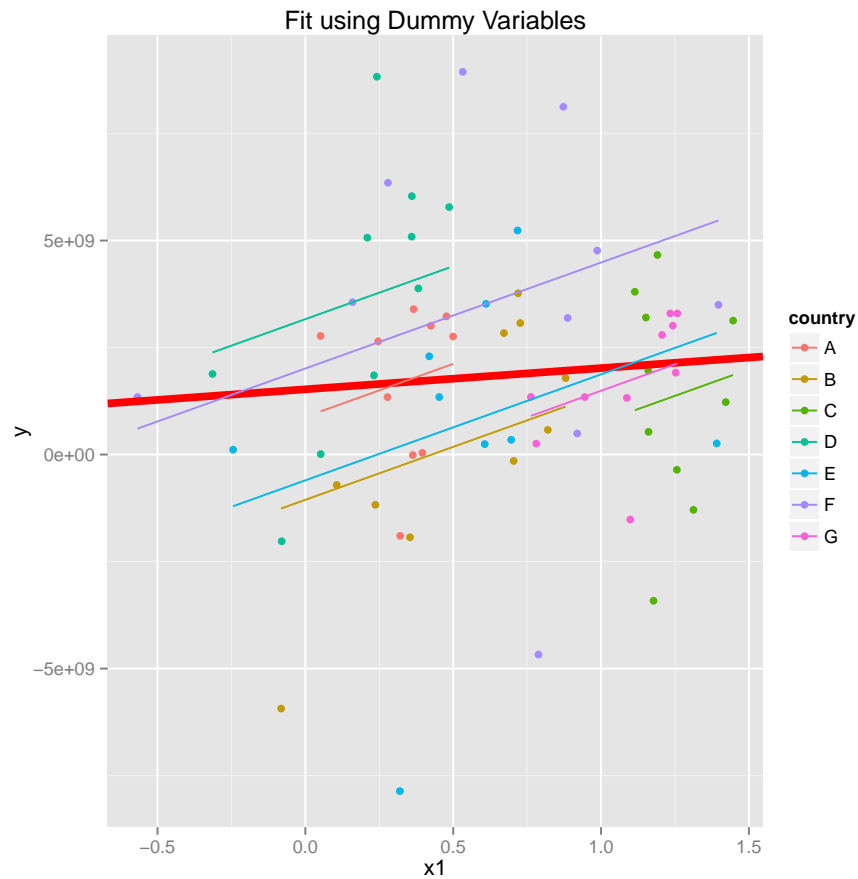
```
## -8.63e+09 -9.70e+08 5.40e+08 1.39e+09 5.61e+09
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## x1              2.48e+09   1.11e+09    2.24  0.02889 *
## factor(country)A  8.81e+08   9.62e+08    0.92  0.36347
## factor(country)B -1.06e+09   1.05e+09   -1.01  0.31811
## factor(country)C -1.72e+09   1.63e+09   -1.06  0.29508
## factor(country)D  3.16e+09   9.09e+08    3.48  0.00093 ***
## factor(country)E -6.03e+08   1.06e+09   -0.57  0.57329
## factor(country)F  2.01e+09   1.12e+09    1.79  0.07821 .
## factor(country)G -9.85e+08   1.49e+09   -0.66  0.51190
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.8e+09 on 62 degrees of freedom
## Multiple R-squared:  0.44, Adjusted R-squared:  0.368
## F-statistic: 6.09 on 8 and 62 DF, p-value: 8.89e-06
```

To compare to the pooled ols we issue the command

```
ols <- lm(y ~ x1, data = panel)
```

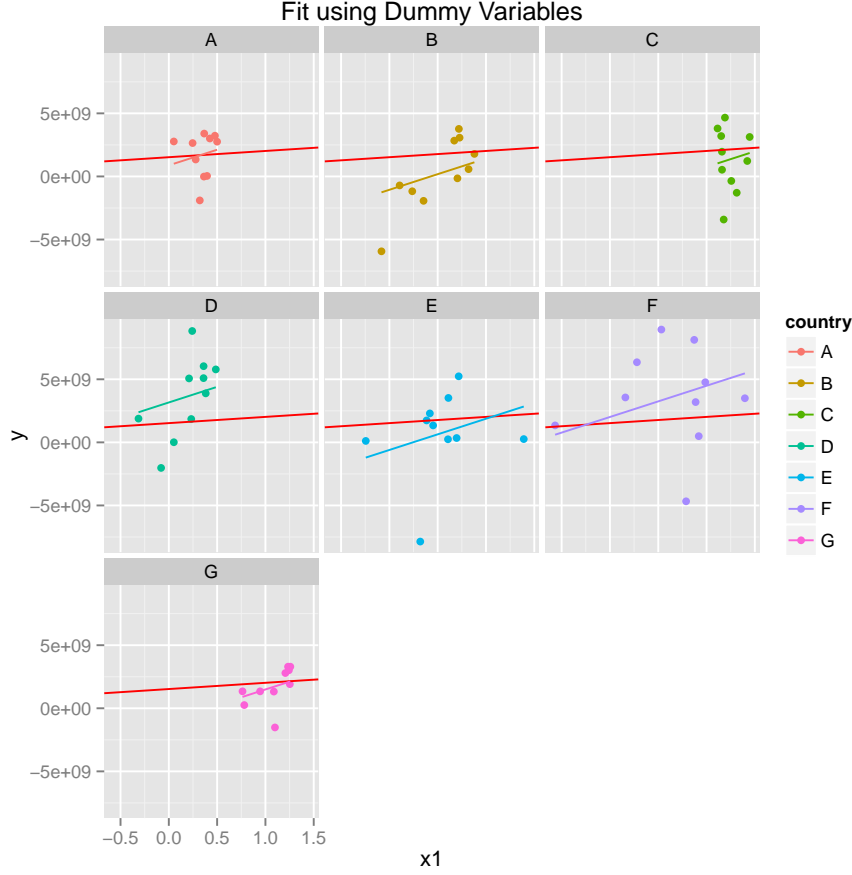
and then plot using

```
yhat <- fixed.dum$fitted
library(ggplot2)
# plot fitted dummy
p <- ggplot(panel)
p <- p + geom_point(aes(x = x1, y = y, group = country,
  colour = country))
p <- p + geom_abline(intercept = ols$coefficients[1],
  slope = ols$coefficients[2], colour = "red", size = 2)
p <- p + geom_line(aes(x = x1, y = yhat, group = country,
  colour = country))
p <- p + ggtitle("Fit using Dummy Variables")
p
```



Alternatively, we can show each in a separate graph as follows:

```
panel$yhat <- yhat
p <- ggplot(panel)
p <- p + geom_point(aes(x = x1, y = y, group = country,
  colour = country))
p <- p + geom_abline(intercept = ols$coefficients[1],
  slope = ols$coefficients[2], colour = "red")
p <- p + geom_line(aes(x = x1, y = yhat, group = country,
  colour = country))
p <- p + ggtitle("Fit using Dummy Variables")
p <- p + facet_wrap(~country)
p
```



7.2 The Within Estimator (FE)

The *Within Estimator* is also known as the *Fixed Effects Estimator*, and is mathematically equivalent to using the *Dummy Estimator*, but is computationally more efficient, thus is the preferred method.

The *Within Estimator* estimates the fixed effects by first *demeaning* and then using ordinary least squares on the result. That is, summing over t for each i we have

$$\sum_t y_{i,t} = T\alpha + \sum_t X_{i,t}\beta + T\delta_i + \sum_t \epsilon_{i,t} \quad (7.5)$$

Dividing by T and subtracting from equations we have

$$y_{i,t} - \bar{y}_i = (X_{i,t} - \bar{X}_i)^T + \epsilon_{i,t} - \bar{\epsilon}_i \quad (7.6)$$

where $\bar{y}_i = \frac{1}{T} \sum_t y_{i,t}$, $\bar{x}_i = \frac{1}{T} \sum_t x_{i,t}$ and $\bar{\epsilon}_i = \frac{1}{T} \sum_t \epsilon_{i,t}$. The within estimator is

ordinarily least squares estimation applied to equation 7.6

To perform within estimation we use the *plm* package, as follows:

```
library(foreign)
panelData <- read.dta("http://dss.princeton.edu/training/Panel101.dta")
library(plm)
fixed.within <- plm(y ~ x1, data = panelData, index = c("country",
  "year"), model = "within")
summary(fixed.within)

## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = y ~ x1, data = panelData, model = "within", index = c("country",
##   "year"))
##
## Balanced Panel: n=7, T=10, N=70
##
## Residuals :
##      Min.    1st Qu.    Median      Mean     3rd Qu.      Max.
## -8.63e+09 -9.70e+08  5.40e+08  0.00e+00  1.39e+09  5.61e+09
##
## Coefficients :
##      Estimate Std. Error t-value Pr(>|t|)
## x1 2.48e+09    1.11e+09    2.24   0.029 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    5.24e+20
## Residual Sum of Squares: 4.85e+20
## R-Squared      : 0.0747
##      Adj. R-Squared : 0.0661
## F-statistic: 5.00411 on 1 and 62 DF, p-value: 0.0289
```

We can also get the α_i by

```
fixef(fixed.within)

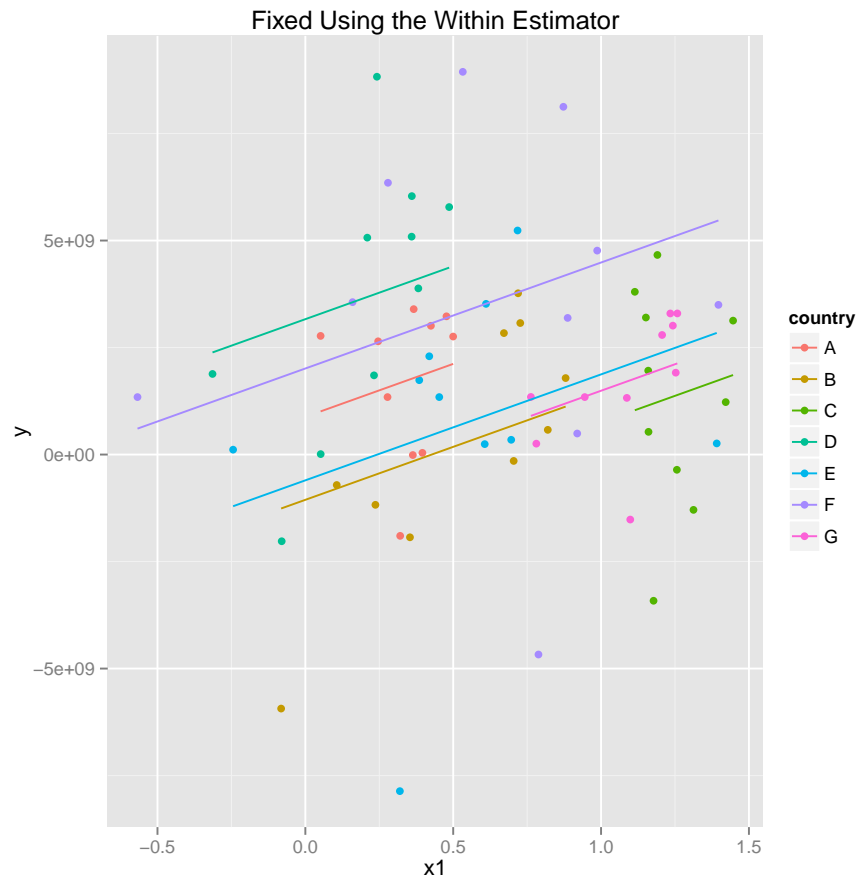
##           A           B           C           D           E           F
## 8.805e+08 -1.058e+09 -1.723e+09  3.163e+09 -6.026e+08  2.011e+09
##           G
## -9.847e+08
```

To graph in the same as before we need to do a little work to created an analogous "fitted" function, which is called `hat` below.

```
hat <- function(x, i) {  
  b <- fixef(fixed.within)  
  m <- fixed.within[[1]]  
  b[i] + m * x  
}
```

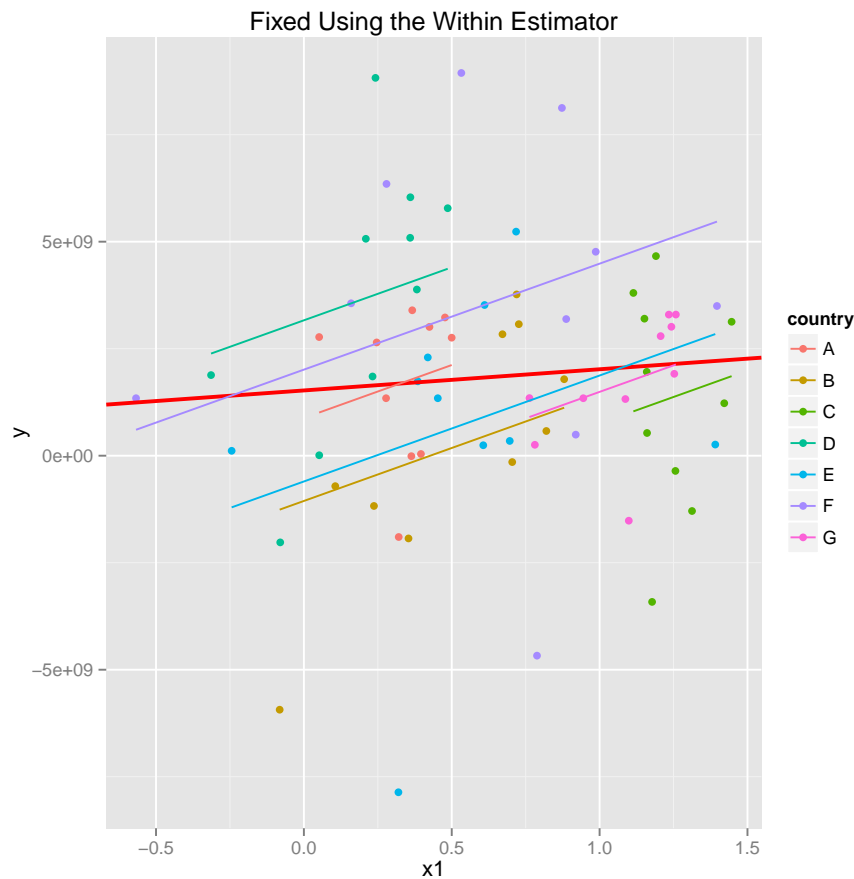
Then we apply `hat` to get `yhat` and proceed as before.

```
yhat <- hat(panelData$x1, panelData$country)  
library(ggplot2)  
panelData$yhat <- yhat  
p <- ggplot(panelData)  
p <- p + geom_point(aes(x = x1, y = y, group = country,  
  colour = country))  
p <- p + geom_line(aes(x = x1, y = yhat, group = country,  
  colour = country))  
p <- p + ggtitle("Fixed Using the Within Estimator")  
p
```



To place the ols on this graph we can also use *plm*, but with the "pooling" model:

```
pooled <- plm(y ~ x1, data = panelData, index = c("country",
  "year"), model = "pooling")
yhat <- hat(panelData$x1, panelData$country)
library(ggplot2)
panelData$yhat <- yhat
p <- ggplot(panelData)
p <- p + geom_point(aes(x = x1, y = y, group = country,
  colour = country))
p <- p + geom_abline(intercept = pooled$coefficients[1],
  slope = pooled$coefficients[2], colour = "red",
  size = 1)
p <- p + geom_line(aes(x = x1, y = yhat, group = country,
  colour = country))
p <- p + ggtitle("Fixed Using the Within Estimator")
p
```

7.3 Fixed vs Pooling Models

Given panel data, how can ascertain which model is more appropriated: *Fixed* or *Pooling*? One way is to use the *F-test*, as follows:

```
library(foreign)
panelData <- read.dta("http://dss.princeton.edu/training/Panel101.dta")
library(plm)
model.fixed <- plm(y ~ x1, data = panelData, index = c("country",
  "year"), model = "within")
model.pooled <- plm(y ~ x1, data = panelData, index = c("country",
  "year"), model = "pooling")
pFtest(model.fixed, model.pooled)
```

```
##
## F test for individual effects
##
## data: y ~ x1
## F = 2.966, df1 = 6, df2 = 62, p-value = 0.01307
## alternative hypothesis: significant effects
```

If the p-value is less than 0.05 then the NULL hypothesis is rejected and the fixed.model is considered more appropriate.

7.4 FE Panel Data Generation

```
country <- c("alpha", "beta", "gamma", "delta")
T <- c(2001, 2002, 2003, 2004, 2005)
panel <- expand.grid(country = country, year = T)
panel$x <- runif(nrow(panel), 10, 20)
library(plyr)
delta <- dapply(panel, .(country), function(df) {
  sum(df$x)
})
delta <- delta/sqrt(delta %*% delta)
alpha <- 5
epsilon <- runif(nrow(panel), -1, 1)
panel$y <- alpha + delta + 2 * df$x + epsilon
```

```
head(panel)
```

```
## country year      x      y
## 1  alpha 2001 19.90  7.122
## 2  beta 2001 13.51 10.337
## 3 gamma 2001 17.90 12.019
## 4 delta 2001 16.79 14.358
## 5  alpha 2002 14.93  7.501
## 6  beta 2002 12.24 10.409
```

Chapter 8

The Random Effects Model

8.1 The RE Model

If we alter the conditions on the *Fixed Effects* model to allow for a random δ_i that is uncorrelated with $X_{i,t}$, then since δ_i can be combined with the $\epsilon_{i,t}$ term to producing random effects with differing variances: That is setting $\zeta_{i,t} = \delta_i + \epsilon_{i,t}$, we have the *Random Effects Model* who has the form of:

$$Y_{i,t} = \alpha + X_{i,t}^T \beta + \zeta_{i,t} \quad (8.1)$$

- global constant
- explanatory coefficients
- random effects component ($\delta_i + \epsilon_{i,t}$)

Here, our assumptions are:

- The period effect γ_t satisfies $\gamma_t = 0$ for each t .
- The residual $\epsilon_{i,t}$ satisfies $\epsilon_{i,t} \sim IDD(0, \sigma_\epsilon^2)$
- δ_i is uncorrelated with $\epsilon_{i,t}$, that is $\rho(\delta_i, \epsilon_{i,t}) = 0$
- δ_i is uncorrelated with $X_{i,t}$, that is $\rho(\delta_i, X_{i,t}) = 0$
- $X_{i,t}$ is uncorrelated with $\epsilon_{i,t}$, that is $\rho(X_{i,t}, \epsilon_{i,t}) = 0$

Now since $\rho(\delta_i, X_{i,t}) = 0$ we have $var(\zeta_{i,t}) = var(\delta_i) + var(\epsilon_{i,t})$. Moreover, assuming

- $\delta_i \sim IID(0, \sigma_\delta^2)$
- $\epsilon_{i,t} \sim IID(0, \sigma_\epsilon^2)$

the covariance structure for the composite errors $E[\zeta_{i,t}\zeta_{j,s}]$ becomes

$$E[\zeta_{i,t}\zeta_{j,s}] = \begin{pmatrix} \sigma_\delta^2 + \sigma_\epsilon^2 & \sigma_\delta^2 & \cdots & \sigma_\delta^2 \\ \sigma_\delta^2 & \sigma_\delta^2 + \sigma_\epsilon^2 & \cdots & \sigma_\delta^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_\delta^2 & \sigma_\delta^2 & \cdots & \sigma_\delta^2 + \sigma_\epsilon^2 \end{pmatrix} \quad (8.2)$$

and the variance-covariance matrix for the entire disturbances is given by

$$\Omega = I_n \otimes \Sigma = \begin{pmatrix} \Sigma & 0 & \cdots & 0 \\ 0 & \Sigma & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma \end{pmatrix} \quad (8.3)$$

The *Random Effects* model corresponds to a GLS regression model

$$\begin{aligned} \vec{Y} &= \alpha \iota_{NT} + X^T \beta + \zeta \\ \text{var}(\zeta) &= \sigma_\delta^2 (I_N \otimes J_T) + \sigma_\epsilon^2 I_{NT} \end{aligned} \quad (8.4)$$

where ι_{NT} is a vector of length NT composed entirely of 1's. Let $W = c(\iota_{NT}, X^T)$, that is result of concatenating ι_{NT} with X^T . Then the problem of finding $\hat{\beta}$ takes the form of

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \left(W^T \hat{\Omega} W \right)^{-1} W^T \hat{\Omega}^{-1} Y \quad (8.5)$$

In practice, Ω is not known, but must be estimated. Since Ω is expressed in terms of σ_δ^2 and σ_ϵ^2 , this may be accomplished by estimation of σ_δ^2 and σ_ϵ^2 . There are many ways to estimate these parameters, the estimation methods supported in R are:

- **SWAR** method (the default method)
- **AMEMIYA** method
- **WALHUS** method
- **NERLOVE** method
- **KINLA** method

8.2 Random Effects Estimation

In this section we use R to estimate the parameters for a Random Effects Model. We begin by creating some data:

```
country <- outer(LETTERS, letters[1:9], function(x,
  y) {
    paste(x, y, sep = " ")
  })
df <- expand.grid(country = country, Year = 2008:2010)
df$x <- runif(nrow(df), 1, 10)
alpha <- runif(length(country), 0, 20)
names(alpha) <- country
df$y <- alpha[df$country] + df$x * 2 + runif(nrow(df),
  -5, 5)
write.csv(df, "./Data/random1.csv", row.names = FALSE)
df <- NULL
```

The motivation here is to generate data with many countries having randomly distributed δ_i that are independent of $X_{i,t}$. Note, we used the *outer* function as a cheap device to generate, $26 \times 10 = 260$, country names, and we use the function *expand.grid* to generate all country-year combinations. We constructed this data to satisfy

$$y_{i,t} = 10 + 2 * x_{i,t} + (\delta_i + \epsilon_{i,t}) \quad (8.6)$$

Next we read in the desired panel data and examine the head

```
random.panel.data <- read.csv("./Data/random1.csv")
head(random.panel.data)

##   country Year      x      y
## 1      Aa 2008 5.969 21.16
## 2      Ba 2008 1.652 17.81
## 3      Ca 2008 8.784 19.21
## 4      Da 2008 8.261 34.16
## 5      Ea 2008 8.947 20.74
## 6      Fa 2008 6.209 12.55
```

And next, we create the *Random Model*

```
library(plm)
random.model <- plm(y ~ x, data = random.panel.data,
  effect = "individual", model = "random", random.method = "swar")
```

Finally we inspect the results

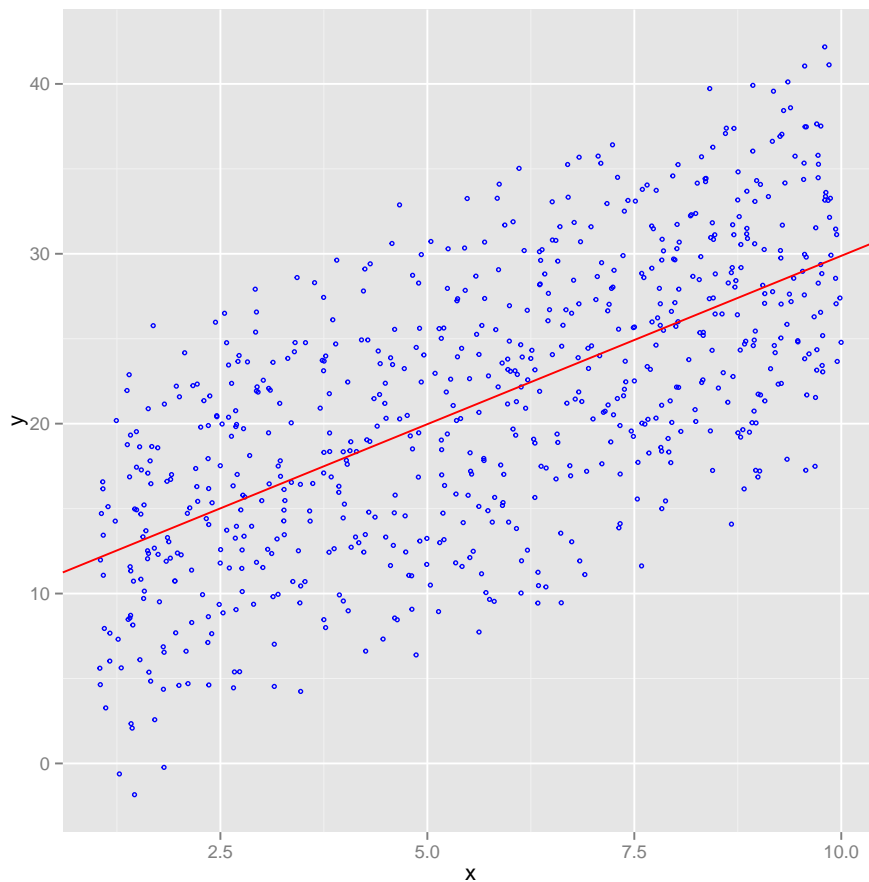
```
summary(random.model)

## Oneway (individual) effect Random Effect Model
##      (Swamy-Arora's transformation)
##
## Call:
## plm(formula = y ~ x, data = random.panel.data, effect = "individual",
##      model = "random", random.method = "swar")
##
## Balanced Panel: n=234, T=3, N=702
##
## Effects:
##              var std.dev share
## idiosyncratic  7.70    2.78  0.2
## individual    31.55    5.62  0.8
## theta:  0.726
##
## Residuals :
##      Min. 1st Qu.  Median 3rd Qu.    Max.
## -7.7100 -2.0800 -0.0209  1.9400   6.6400
##
## Coefficients :
##              Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  10.0607    0.4645    21.7   <2e-16 ***
## x              1.9818    0.0468    42.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    19200
## Residual Sum of Squares: 5390
## R-Squared      : 0.719
##      Adj. R-Squared : 0.717
## F-statistic: 1792.24 on 1 and 700 DF, p-value: <2e-16
```

Note: The p-value less than .05 means that all coefficients in the model are non-zero.

As usual we plot the output, shown in

```
library(ggplot2)
m <- random.model$coefficients["x"]
b <- random.model$coefficients["(Intercept)"]
ggplot(random.panel.data) + geom_point(aes(x = x, y = y),
  shape = 1, size = 1, color = "blue") + geom_abline(intercept = b,
  slope = m, color = "red")
```



8.3 Lagrange Multiplier Test

From the looks of the graph, one might wonder if pooling might be the appropriate model for *random.panel.data*

We can test this hypothesis using a *Lagrange Multiplier Test*. First we create a Pooling Estimation.

```
library(plm)
pooling.model <- plm(y ~ x, data = random.panel.data,
  effect = "individual", model = "pooling", )
```

Next we perform the *Lagrange Multiplier Test*

```
plmtest(pooling.model)
```

```
##
##  Lagrange Multiplier Test - (Honda)
##
## data:  y ~ x
## normal = 21.25, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

And we see that the hypothesis NULL hypothesis is rejected!

8.4 Hausman Test

At this point, having rejected the *Pool Model* one might consider the consider using the *Fixed Effects Model*. To distinguish these two choices, we apply the *Hausman Test* to the pair of models. In R, this can be accomplished as using the *phptest* function in *plm* package.

```
library(plm)
random.panel.data <- read.csv("./Data/random1.csv")
random.model <- plm(y ~ x, data = random.panel.data,
  effect = "individual", model = "random", )
fixed.model <- plm(y ~ x, data = random.panel.data,
  effect = "individual", model = "within", )
phptest(fixed.model, random.model)

##
##  Hausman Test
##
## data:  y ~ x
## chisq = 1.004, df = 1, p-value = 0.3163
## alternative hypothesis: one model is inconsistent
```

If the resulting *p-value* is less than 0.05 then the *Fixed Model* is preferred. In this case we see that the *Random Effects* model is the preferred model

However, the *phptest* function can generate the models necessary models, providing a simpler way to perform this same test:

```
library(plm)
random.panel.data <- read.csv("./Data/random1.csv")
phptest(y ~ x, data = random.panel.data)

##
##  Hausman Test
##
## data:  y ~ x
## chisq = 1.004, df = 1, p-value = 0.3163
## alternative hypothesis: one model is inconsistent
```


8.5 Eliminating Additional Explanatory Variables

In order to examine the effect of additional variables, we modify the panel data of the previous example, by adding an independent variable, called z .

```
random.panel.data <- read.csv("./Data/random1.csv")
modified.panel.data <- data.frame(random.panel.data,
  z = runif(nrow(random.panel.data), 0, 20))
head(modified.panel.data)

##   country Year      x      y      z
## 1      Aa 2008 5.969 21.16  4.571
## 2      Ba 2008 1.652 17.81 19.127
## 3      Ca 2008 8.784 19.21  4.367
## 4      Da 2008 8.261 34.16  3.325
## 5      Ea 2008 8.947 20.74  8.848
## 6      Fa 2008 6.209 12.55 10.804
```

and then perform a *random model estimation* on *modified.panel.data* with two explanatory variables: x and z

```
library(plm)
modified.random.model <- plm(y ~ x + z, data = modified.panel.data,
  effect = "individual", model = "random", random.method = "swar")
```

Finally we inspect the results

```
summary(modified.random.model)

## Oneway (individual) effect Random Effect Model
##   (Swamy-Arora's transformation)
##
## Call:
## plm(formula = y ~ x + z, data = modified.panel.data, effect = "individual",
##     model = "random", random.method = "swar")
##
## Balanced Panel: n=234, T=3, N=702
##
## Effects:
##               var std.dev share
## idiosyncratic  7.68    2.77  0.2
## individual    31.52    5.61  0.8
## theta:  0.726
##
## Residuals :
##   Min. 1st Qu.  Median 3rd Qu.    Max.
```

```
## -7.6900 -2.0700 0.0433 1.9600 6.5300
##
## Coefficients :
##             Estimate Std. Error t-value Pr(>|t|)
## (Intercept)   9.7660     0.5082   19.22  <2e-16 ***
## x             1.9815     0.0468   42.37  <2e-16 ***
## z             0.0297     0.0208    1.43    0.15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    19200
## Residual Sum of Squares: 5370
## R-Squared          : 0.72
##      Adj. R-Squared : 0.717
## F-statistic: 899.112 on 2 and 699 DF, p-value: <2e-16
```

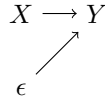
Note For z , the value of $Pr(> |t|)$ exceeding 0.05 means that z should be eliminated as an explanatory variable.

Chapter 9

Instrument Variables

9.1 Bias Arrises From Endogeneity of X

In the the previous sections we made the the assumption that the error term was uncorrelated with the explanatory variables. One way to visualize this is

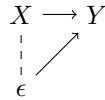


Here the arrows illustrate Y dependence on X and ϵ .

Now the assumption that X and ϵ are uncorrelated may not in practice be true, i.e. we may find ourselves in the situation where

$$\rho(X_{i,t}, \epsilon_{i,t}) \neq 0 \quad (9.1)$$

We can illustrate the correlation between X and ϵ by connecting them with a line as follows:



In this case, when X and ϵ are correlated the OLS estimator becomes *biased*. This is easily illustrated by considering the following simple example: Let

$$Y_i = \beta X_i + \epsilon \quad (9.2)$$

Now since, $\hat{\beta} = [X^T X]^{-1} x^T Y$ we see

$$\hat{\beta} \rightarrow \frac{E(XY)}{E(XX)} = \frac{E(X(\beta X + \epsilon))}{E(X^2)} = \beta + \frac{E(X\epsilon)}{E(X^2)} \quad (9.3)$$

Now recall

$$\sigma_X^2 = E(X^2) - \mu_X^2 \quad (9.4)$$

and

$$\rho_{X\epsilon} = \frac{E(X\epsilon) - \mu_X\mu_\epsilon}{\sigma_X\sigma_\epsilon} \quad (9.5)$$

So

$$\frac{E(X\epsilon)}{E(X^2)} = \frac{\sigma_X\sigma_\epsilon\rho_{X\epsilon} + \mu_X\mu_\epsilon}{\sigma_X^2 + \sigma_\epsilon^2} \quad (9.6)$$

Now since $\mu_\epsilon = 0$ the last term drops out so we get

$$\hat{\beta} \rightarrow \beta + \frac{\sigma_X\sigma_\epsilon\rho_{X\epsilon}}{\sigma_X^2 + \sigma_\epsilon^2} \quad (9.7)$$

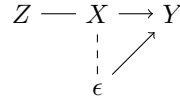
Now since $\sigma_X \neq 0$ ¹ and $\sigma_\epsilon \neq 0$ we see that the necessary and sufficient condition for $\hat{\beta} \rightarrow \beta$ is that $\rho_{X\epsilon} \neq 0$.

9.2 Two Stage Least Squares

Thus, when X and ϵ are correlated, we are tempted find a work around in order to avoid bias.

One such work around is to have another variable, say Z , which is correlated with X but not correlated with ϵ . Such a variable Z is called an *Instrument Variable*.

We may visualize this as



An instrument variable can allow us to avoid bias by replacing the role of X in $Y = X\beta + \epsilon$ by a proxy \tilde{X} . More precisely, first regress on

$$X = Z\alpha + \eta \quad (9.8)$$

to obtain $\hat{\alpha}$ Next compute the predicted values

$$\tilde{X} = Z\hat{\alpha} \quad (9.9)$$

Finally compute $\hat{\beta}$ by regressing on

$$Y = \tilde{X}\beta + \epsilon \quad (9.10)$$

Since Z and ϵ are uncorrelated, \tilde{X} and ϵ are also uncorrelated.

¹there is more than one X value

9.3 Combining the 2 Stage Regression Calculations

In the previous section we used instrument variables to perform a two stage regression. Here we will examine the details and combine to provide a single equivalent calculation.

Now the OLS solution of $X = Z\alpha + \eta$ is given by

$$\hat{\alpha} = [Z^T Z]^{-1} Z^T X \quad (9.11)$$

so \tilde{X} is given by

$$\tilde{X} = Z\hat{\alpha} = Z[Z^T Z]^{-1} Z^T X \quad (9.12)$$

For notational convenience, define P by

$$P = Z[Z^T Z]^{-1} Z^T \quad (9.13)$$

Then

$$\tilde{X} = PX \quad (9.14)$$

and $\hat{\beta}$ is given by

$$\hat{\beta} = [(PX)^T (PX)]^{-1} (PX)^T Y \quad (9.15)$$

So

$$\hat{\beta} = [X^T P^T P X]^{-1} X^T P^T Y \quad (9.16)$$

This can be simplified further by noting that P is symmetric,² to get

$$\hat{\beta} = [X^T P^2 X]^{-1} X^T P Y \quad (9.17)$$

Furthermore, we note P is a projection,³ thus

$$\hat{\beta} = [X^T P X]^{-1} X^T P Y \quad (9.18)$$

Now eq 9.18 is the general form for the solution using instrument variables.

In the event that the number of instruments is equal to the number of explanatory variables, the term $Z^T X$ becomes a square matrix. In this case, it makes sense to speak of $[Z^T X]^{-1}$ and to recast eq 9.18 as follows: Insert $X Z^T (X Z^T)^{-1}$ in front of the Y of eq 9.18 to get

$$\hat{\beta} = [X^T P X]^{-1} X^T P X Z^T (X Z^T)^{-1} Y \quad (9.19)$$

Since the product of $[X^T P X]^{-1}$ and $[X^T P X Z^T]$ is the identity,

$$\hat{\beta} = Z^T (X Z^T)^{-1} Y \quad (9.20)$$

Now premultiply $Z^T (X Z^T)^{-1} Y$ by the identity $(Z^T X)^{-1} Z^T X$ to get

$$\hat{\beta} = (Z^T X)^{-1} Z^T X Z^T (X Z^T)^{-1} Y \quad (9.21)$$

Reducing the $X Z^T (X Z^T)^{-1}$ term we finally get

$$\hat{\beta} = (Z^T X)^{-1} Z^T Y \quad (9.22)$$

²This follows since $P^T = [Z[Z^T Z]^{-1} Z^T]^T = [Z^T]^T [[Z^T Z]^{-1}]^T [Z]^T = Z[[Z^T Z]^T]^{-1} Z^T = Z[Z^T Z]^{-1} Z^T$. Thus $P^T = P$, so P is symmetric

³To see P is a projection, note $P^2 = [Z[Z^T Z]^{-1} Z^T] [Z[Z^T Z]^{-1} Z^T] = Z[Z^T Z]^{-1} Z^T Z [Z^T Z]^{-1} Z^T = Z[Z^T Z]^{-1} Z^T = P$

Chapter 10

Guidelines for Model Selection

To better understand the interplay between the model and our tests consider fig 10
To better understand the interplay between the model and our tests consider fig 10
To better understand the interplay between the model and our tests consider fig 10
To better understand the interplay between the model and our tests consider fig 10
To better understand the interplay between the model and our tests consider fig 10
To better understand the interplay between the model and our tests consider fig 10
To better understand the interplay between the model and our tests consider fig 10
To better understand the interplay between the model and our tests consider fig 10
To better understand the interplay between the model and our tests consider fig 10
To better understand the interplay between the model and our tests consider fig 10 To
better understand the interplay between the model and our tests consider fig 10

Review of Ordinary Least Square Regression (OLS)

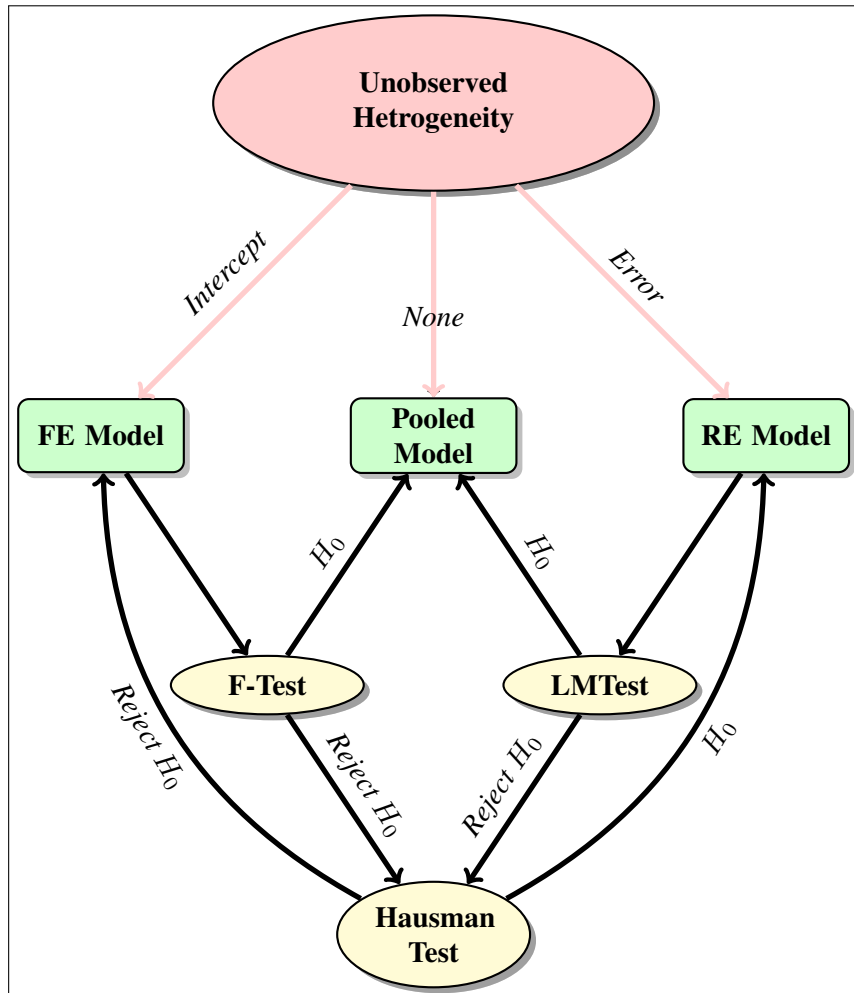


Figure 10.1: Pooled, Fixed Effects, Random Effects at a Glance

Appendices

.1 Ordinary Least Square Regression (OLS)

R has an extensive *Stats* package, containing numerous methods and is well worth exploring. However, our modeling efforts will concentrate on linear models. To better understand these, we begin with quick review of ordinary linear regression

The goal in OLS is to find a linear relationship between a dependent variable, y and one or more explanatory variables x based upon a collection of observations $\{y_i, \vec{x}_i\}_1^N$, where in general $\vec{x} = \langle x_1, \dots, x_P \rangle$. The general form of ordinary least squares regression is given by

$$y_i = \alpha + \sum_{j=1}^P \vec{x}_{i,j} \beta_j + \epsilon_i \quad (1)$$

$$y_i = \alpha + \vec{x}_i^T \vec{\beta} + \epsilon_i \quad (2)$$

Or in matrix form

$$\vec{y} = M\vec{\theta} + \vec{\epsilon} \quad (3)$$

where the i^{th} row of M is given by $\{x_{i,1}, \dots, x_{i,P}, 1\}$ and the vector $\vec{\theta}$ is given by $\{1, \dots, 1\}$ and is of length $P + 1$. Here we assume ϵ_i 's are uncorrelated and have mean 0.

.1.1 A Simple Example

Suppose we have observed the following data

	y	x
1	1.00	1.00
2	4.00	2.00
3	5.00	3.00
4	7.00	4.00
5	8.00	5.00

Table 1: Simple Example

and we want to represent the relationship between x and y using a single line, say $y = mx + b$. This corresponds to the systems of equations

$$\begin{aligned} 1 &= 1m + 1b \\ 4 &= 2m + 1b \\ 5 &= 3m + 1b \\ 7 &= 4m + 1b \\ 8 &= 5m + 1b \end{aligned} \quad (4)$$

or in matrix notation

$$\vec{Y} = M\vec{\theta} \quad (5)$$

where

$$\vec{Y} = \begin{bmatrix} 1 \\ 4 \\ 5 \\ 7 \\ 8 \end{bmatrix}, M = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{bmatrix}, \vec{\theta} = \begin{bmatrix} m \\ b \end{bmatrix} \quad (6)$$

Now obviously this system of equations is inconsistent, so the best we can do is to solve for

$$\vec{Y} = M\vec{\theta} + \vec{\epsilon} \quad (7)$$

with the minimum absolute value of the residual term $\vec{\epsilon} = \vec{Y} - M\vec{\theta}$, ie

$$\hat{\theta} = \operatorname{argmin}_{\theta} \|\vec{Y} - M\vec{\theta}\| \quad (8)$$

Noting that the $M\hat{\theta}$ is the projection of \vec{Y} onto the column space of M , it is clear that $\vec{Y} - M\hat{\theta}$ is orthogonal to the column space of M , and hence in the null space of M^T , i.e.

$$M^T [\vec{Y} - M\hat{\theta}] = \vec{0} \quad (9)$$

and so we have

$$M^T \vec{Y} = M^T M \hat{\theta} \quad (10)$$

Equation 10 is called the normal equation. Assuming $\operatorname{rank}(M^T M)$ is non-zero, this may be solved as

$$\hat{\theta} = [M^T M]^{-1} M^T \vec{Y} \quad (11)$$

.

Our Analysis

The salient point of our analysis is, we assumed that a linear relationship existed of the form

$$\vec{Y}_i = \alpha + \sum_{j=1}^P X_{i,j} \beta_j + \epsilon_i \quad (12)$$

. or coding it into matrix notation

$$Y = M\theta + \vec{\epsilon} \quad (13)$$

, and we produced an algorithm to estimate θ . Moreover, we implicitly assumed that the ϵ_i had

And

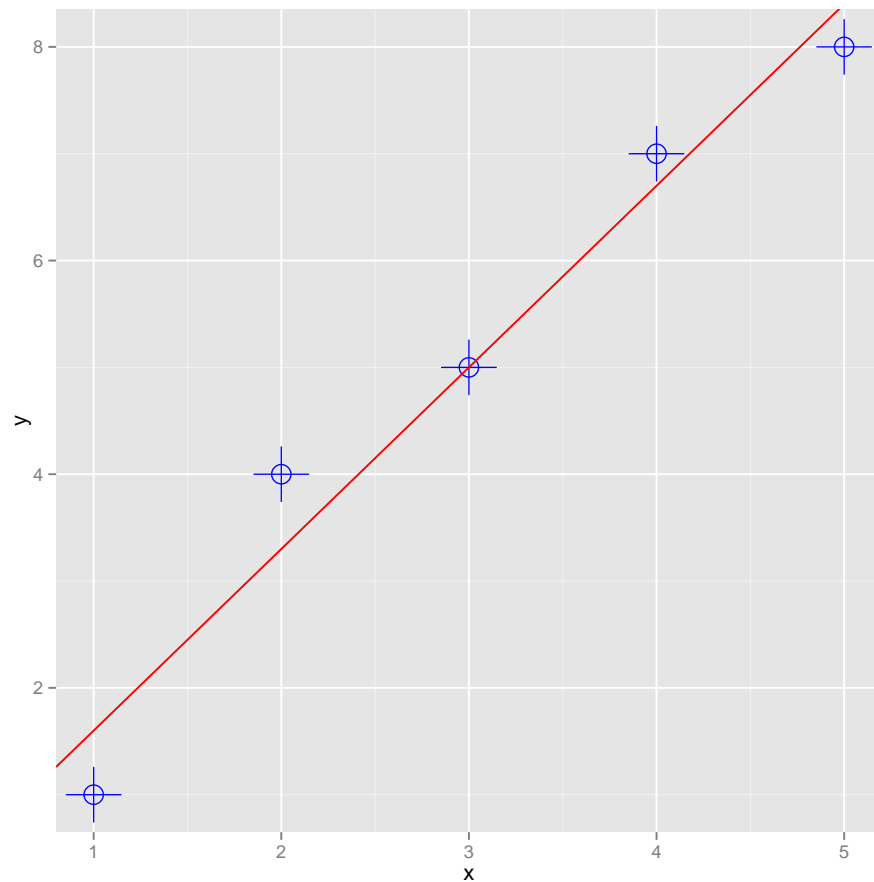
Using the *Stats* package we solve by issuing the *lm* (linear model) command.

```
df <- data.frame(y = c(1, 4, 5, 7, 8), x = c(1, 2,
      3, 4, 5))
fit <- lm(y ~ x, df)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##      1      2      3      4      5
## -6.00e-01  7.00e-01  2.08e-17  3.00e-01 -4.00e-01
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.100      0.635   -0.16   0.885
## x              1.700      0.191    8.88   0.003 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.606 on 3 degrees of freedom
## Multiple R-squared:  0.963, Adjusted R-squared:  0.951
## F-statistic: 78.8 on 1 and 3 DF,  p-value: 0.00301
```

We may plot our *fit* as follows

```
m <- fit$coefficients["x"]
b <- fit$coefficients["(Intercept)"]
ggplot(df) + geom_point(aes(x = x, y = y), shape = 3,
  size = 10, color = "blue") + geom_point(aes(x = x,
  y = y), shape = 1, size = 5, color = "blue") +
  geom_abline(intercept = b, slope = m, color = "red")
```



.2 Gauss Markov Theorem

Our least-squares approach to regression analysis is an optimal estimation. To explain this we require a couple of definitions

First we define *Linear Estimator*

Definition 2: Linear Estimator

A *Linear Estimator* of β_j is a linear combination of the observation Y_i , depending on $X_{i,j}$ but not on the unobserved β_j . that is, $\hat{\beta}_j = \sum c_{i,j} Y_i$. Thus $\hat{\beta} = CY$ for some matrix C where C depends on X .

Next we define what we mean by a *Best Linear Unbiased Estimator*

Definition 3: Best Linear Unbiased Estimator

The *Best Linear Unbiased Estimate* (BLUE) of a parameter θ based on data Y is

- *Linear in Y* The estimator is of the form $\hat{\theta} = B^T \vec{Y}$.
- *Unbiased* $E[\hat{\theta}] = \theta$
- *minimal variance* Has the least variance among all unbiased linear estimators

Theorem: (Gauss-Markov) Suppose $\vec{Y} = M\vec{\theta} + \epsilon$, where $E(\epsilon) = \vec{0}$ and $Var(\epsilon) = \sigma^2 \vec{I}$. The the least square estimate $\hat{\theta} = (M^T M)^{-1} M^T \vec{Y}$ is the *Best Linear Unbiased Estimate* of θ .

Proof: First note that $\hat{\theta}$ is a linear combination of \vec{Y} by eqn 11.

Second note that $E[\hat{\theta} - \theta] = E\left[\left[M^T M\right]^{-1} M^T \vec{Y} - \theta\right] = \left[M^T M\right]^{-1} M^T E[\vec{Y}] - \theta$ and since $E[Y] = E[M\theta + \epsilon] = M\theta + E[\epsilon] = M\theta$. Thus $E[\hat{\theta} - \theta] = \left[M^T M\right]^{-1} M^T M\theta - \theta = 0$. Thus OLS is unbiased.

It remains to show that OLS is optimal in the sense of having the least variance among all linear unbiased estimators. To this end, let $\tilde{\theta}$ be another linear unbiased estimator. To show $var(\tilde{\theta}) \geq var(\hat{\theta})$. Since $\tilde{\theta}$ is a linear estimator, it is of the form $\tilde{\theta} = \sum c_i Y_i = CY$. Now $\tilde{\theta} - \hat{\theta} = [CY - (M^T M)^{-1} M^T Y] = [C - (M^T M)^{-1} M^T] Y$. Set $D = C - (M^T M)^{-1} M^T$, so $\tilde{\theta} - \hat{\theta} = DY$. That is, $\tilde{\theta} = [(M^T M)^{-1} M^T + D] Y$. Taking the expectation and noting that $E[\epsilon] = 0$ we have $E[\tilde{\theta}] = [(M^T M)^{-1} M^T + D] M\theta$. But $\tilde{\theta}$ is unbiased, so $[(M^T M)^{-1} M^T + D] M\theta = M\theta$, i.e. $(I + DM)\theta = \theta$. Thus $DM\theta = 0$ and since D does not depend on θ , $DM = 0$. Now computing the $var(\tilde{\theta})$ we get $var(\tilde{\theta}) = var(CY(CY)^T) = var(CYY^T C^T) = \sigma^2 var(CC^T)$. Now $CC^T = [(M^T M)^{-1} M^T + D] [(M^T M)^{-1} M^T + D]$. Multiplying out and dropping term containing DM we have $CC^T = (M^T M)^{-1} + DD^T$, so $var(\tilde{\theta}) = \sigma^2 (M^T M)^{-1} + \sigma^2 DD^T$. But $\sigma^2 (M^T M)^{-1} = var(\hat{\theta})$ and DD^T is positive semidefinite, so $var(\tilde{\theta}) \geq var(\hat{\theta})$ ■