# A Gentle Introduction to Panel Data Modeling Using R

M. S. Legrand

# Contents

```
## Error:  object of type 'closure' is not subsettable
```

# Chapter 1

# Panel Data Basics

## 1.1 What is Panel Data?

Panel Data[1], 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.$I$ is 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 obversation 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 $t = 2010$, $i = A$, $X_{i,t} = (155.73, 78.61, 25.24)$ 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 representated as a *data.frame* object. A *data.frame* object is similar to a matrix having named colums 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 longitudal data or repeated measures

| | year | country | y | x1 | x2 |
|---|---|---|---|---|---|
| 1 | 2010 | A | 155.73 | 78.61 | 11.64 |
| 2 | 2011 | A | -52.95 | 25.24 | 123.72 |
| 3 | 2012 | A | 135.09 | 69.93 | 34.98 |
| 4 | 2010 | A | 69.64 | 18.45 | 7.54 |
| 5 | 2011 | B | 97.32 | 95.96 | 105.06 |
| 6 | 2012 | B | 147.75 | 91.87 | 56.44 |
| 7 | 2010 | B | -48.93 | 10.18 | 99.81 |
| 8 | 2011 | B | -51.92 | 17.22 | 126.77 |
| 9 | 2012 | C | 205.59 | 98.60 | 2.28 |
| 10 | 2010 | C | 69.00 | 84.94 | 121.57 |
| 11 | 2011 | C | 9.09 | 66.75 | 155.12 |
| 12 | 2012 | C | 42.76 | 93.52 | 184.79 |

Table 1.1: Panel Data

```
head(panel)
```

```
##   year country      y    x1      x2
## 1 2010       A 155.73 78.61  11.637
## 2 2011       A -52.95 25.24 123.723
## 3 2012       A 135.09 69.93  34.984
## 4 2010       A  69.64 18.45   7.535
## 5 2011       B  97.32 95.96 105.063
## 6 2012       B 147.75 91.87  56.437
```

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

```
print(panel)
```

```
##    year country       y    x1      x2
## 1  2010       A 155.728 78.61  11.637
## 2  2011       A -52.953 25.24 123.723
## 3  2012       A 135.086 69.93  34.984
## 4  2010       A  69.635 18.45   7.535
## 5  2011       B  97.320 95.96 105.063
## 6  2012       B 147.747 91.87  56.437
## 7  2010       B -48.926 10.18  99.809
## 8  2011       B -51.925 17.22 126.765
## 9  2012       C 205.589 98.60   2.280
## 10 2010       C  69.003 84.94 121.571
## 11 2011       C   9.092 66.75 155.120
## 12 2012       C  42.762 93.52 184.794
```

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 represend a single observation $X_{i,t}$.

# Chapter 2

# The General Form of the Linear Model

Linear modeling assumes a model of the form

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

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 specifice effects: $\delta_i$, period specific effects $\gamma_t$ and residuals $\epsilon_{i,t}$. The $\delta_i$ accomodates for hetrogenity across individuals, and the $\gamma_t$ accomodates for hetrogenity across time.

The general form of linear model for panel data is:

- Explanatory Variable
- Dependent Variable

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

- global constant
- explanatory coeffcients
- cross sectional effect
- period effect
- residual

With the assumptions:

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

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

**Definition 2: Exogenous**

The error term $X_{i,t}$ is said to be *exogenous* provided it is uncorreltated with $\epsilon_{i,t}$ , that is $\rho(X_{i,t}, \epsilon_{i,t}) = 0$.

Our study will begin by assuming exongenity of our linear models. We will relax this condition in chapter **??** 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.

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

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 respecitvely.

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 simples 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 seperate fields, with each data entry on a seperate 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("./Data/sample1.dta")
head(panelData)

##   country year           y       x1
## 1       F 1990  1.343e+09 -0.56757
## 2       B 1990 -5.935e+09 -0.08185
## 3       E 1990  1.343e+09  0.45287
## 4       D 1990  1.883e+09 -0.31391
## 5       G 1990  1.343e+09  0.94488
## 6       C 1990 -1.292e+09  1.31256
```

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 ...
## 1  30     1      1        0 723545   12  7.667  58.78 ...
## 2  31     1      1        0 723910   12  8.583  73.67 ...
## 3  33     1      1        0 724641   12 10.179 103.62 ...
## 4  35     1      1        0 725371   12 12.179 148.34 ...
## 5  37     1      1        0 725736   12 13.622 185.55 ...
## 6  36     0      0        1 723545   17  7.577  57.41 ...
##   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 muliple 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 revelant 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 Simultions

Sometimes it is useful to generate some artificial panel data, called synthetic 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  7.646  13.04
## 2    beta 2008 18.723  36.03
## 3   gamma 2008 31.179  60.90
## 4   alpha 2009 40.110  76.55
## 5    beta 2009 46.904  92.99
## 6   gamma 2009 56.205 112.46
```

and can be saved using

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

The approach we take is to use synthetic data for our examples for modeling. This provides two benefits:

- • Constructing the data for a specific model can give a better understanding of the model
- • By specifying the parameters within the construction, we can see how well our estimators perform.

For greater detail of how each model may be constructed see Appendix refappendix::Syn

## 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 | 15.54 | 44.38 | 69.85 |
| 2 | dc | y | 31.49 | 89.52 | 143.26 |
| 3 | virginia | x | 20.24 | 50.17 | 79.36 |
| 4 | virginia | y | 41.52 | 97.77 | 158.29 |
| 5 | maryland | x | 34.26 | 55.45 | 96.29 |
| 6 | maryland | y | 71.87 | 109.73 | 191.87 |

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 15.54  44.38  69.85
## 2        dc   y 31.49  89.52 143.26
## 3  virginia   x 20.24  50.17  79.36
## 4  virginia   y 41.52  97.77 158.29
## 5  maryland   x 34.26  55.45  96.29
## 6  maryland   y 71.87 109.73 191.87
```

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 occuring, 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 15.54  44.38  69.85
## 2        dc   y 31.49  89.52 143.26
## 3  virginia   x 20.24  50.17  79.36
## 4  virginia   y 41.52  97.77 158.29
## 5  maryland   x 34.26  55.45  96.29
## 6  maryland   y 71.87 109.73 191.87
```

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 15.54
## 2        dc   y     2008 31.49
## 3  virginia   x     2008 20.24
## 4  virginia   y     2008 41.52
## 5  maryland   x     2008 34.26
## 6  maryland   y     2008 71.87
```

We rename the column for readability

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

```
##      state var year value
## 1       dc   x 2008 15.54
## 2       dc   y 2008 31.49
## 3 virginia   x 2008 20.24
## 4 virginia   y 2008 41.52
## 5 maryland   x 2008 34.26
## 6 maryland   y 2008 71.87
```

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 15.54  31.49
## 2       dc 2009 44.38  89.52
## 3       dc 2011 69.85 143.26
## 4 maryland 2008 34.26  71.87
## 5 maryland 2009 55.45 109.73
## 6 maryland 2011 96.29 191.87
## 7 virginia 2008 20.24  41.52
## 8 virginia 2009 50.17  97.77
## 9 virginia 2011 79.36 158.29
```

## 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

To see a listing we would normally type

```
dir("Data/Yearly")
```

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

| | 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

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

```
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
```

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

```
})
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
## 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.

```
panel <- read.csv("./Data/Pool-2way.csv")
panel$y_bin <- NULL
head(panel)
```

```
##   country year    x1    x2    x3     y
## 1       A 2001 65.54 97.07 71.27 128.1
## 2       B 2001 72.10 85.43 74.43 141.1
## 3       C 2001 59.64 77.57 60.18 119.2
## 4       D 2001 84.23 95.56 86.84 171.2
## 5       E 2001 96.22 79.79 67.63 194.5
## 6       F 2001 56.55 95.84 62.23 112.6
```

```
summary(panel)
```

```
##  country      year            x1              x2        ...
##  A:10    Min.   :2001   Min.   :50.1   Min.   :50.8   ...
##  B:10    1st Qu.:2003   1st Qu.:58.8   1st Qu.:66.9   ...
##  C:10    Median :2006   Median :68.7   Median :78.1   ...
##  D:10    Mean   :2006   Mean   :70.6   Mean   :77.9   ...
##  E:10    3rd Qu.:2008   3rd Qu.:80.4   3rd Qu.:92.2   ...
##  F:10    Max.   :2010   Max.   :97.6   Max.   :99.9   ...
##  G:10                                                 ...
##        y
##  Min.   :102
##  1st Qu.:120
##  Median :140
```

```
##  Mean   :144
##  3rd Qu.:167
##  Max.   :196
##
```

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 130.2
## 2       B 129.8
## 3       C 146.3
## 4       D 148.2
## 5       E 156.4
## 6       F 145.9
## 7       G 151.8
```

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

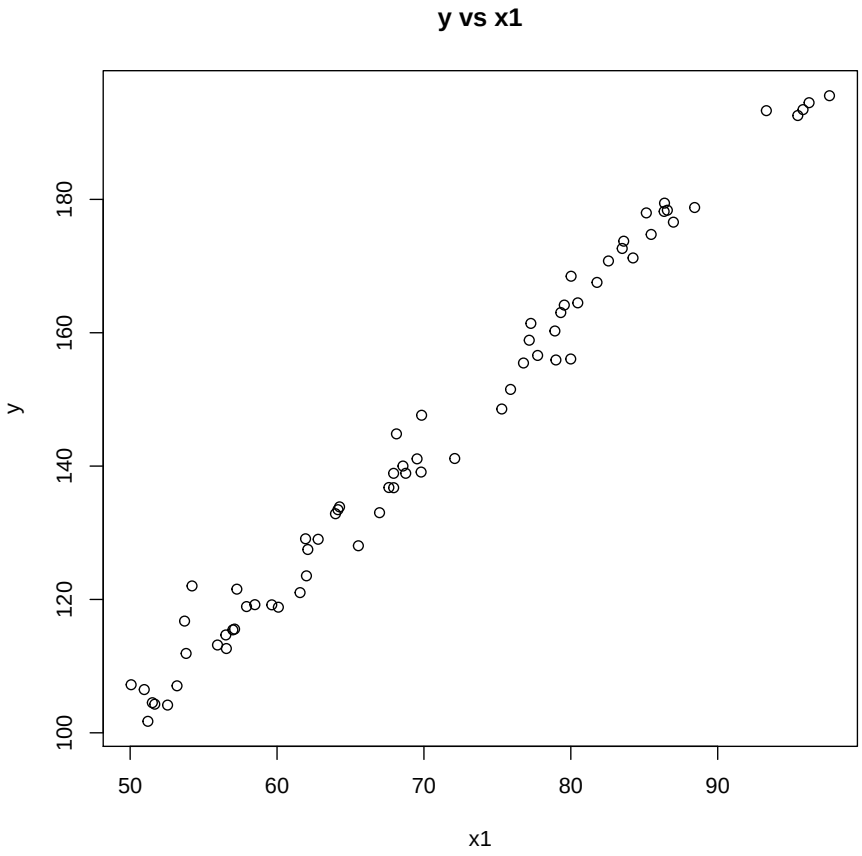
```
library(plyr)
ddply(panel, .(country), function(x) c(meanY = mean(x$y),
    meanX1 = mean(x$x1), meanX2 = mean(x$x2), meanX3 = mean(x$x3),
    varY = var(x$y)))
```

```
##   country meanY meanX1 meanX2 meanX3   varY
## 1       A 130.2  64.80  81.77  77.41  645.0
## 2       B 129.8  63.87  79.34  78.31  231.9
## 3       C 146.3  71.50  76.49  76.99  416.0
## 4       D 148.2  72.32  70.53  85.22 1056.5
## 5       E 156.4  75.46  75.19  73.83  661.5
## 6       F 145.9  71.92  84.56  76.63 1028.2
## 7       G 151.8  74.30  77.47  79.63  841.9
```

## 4.2 Plotting

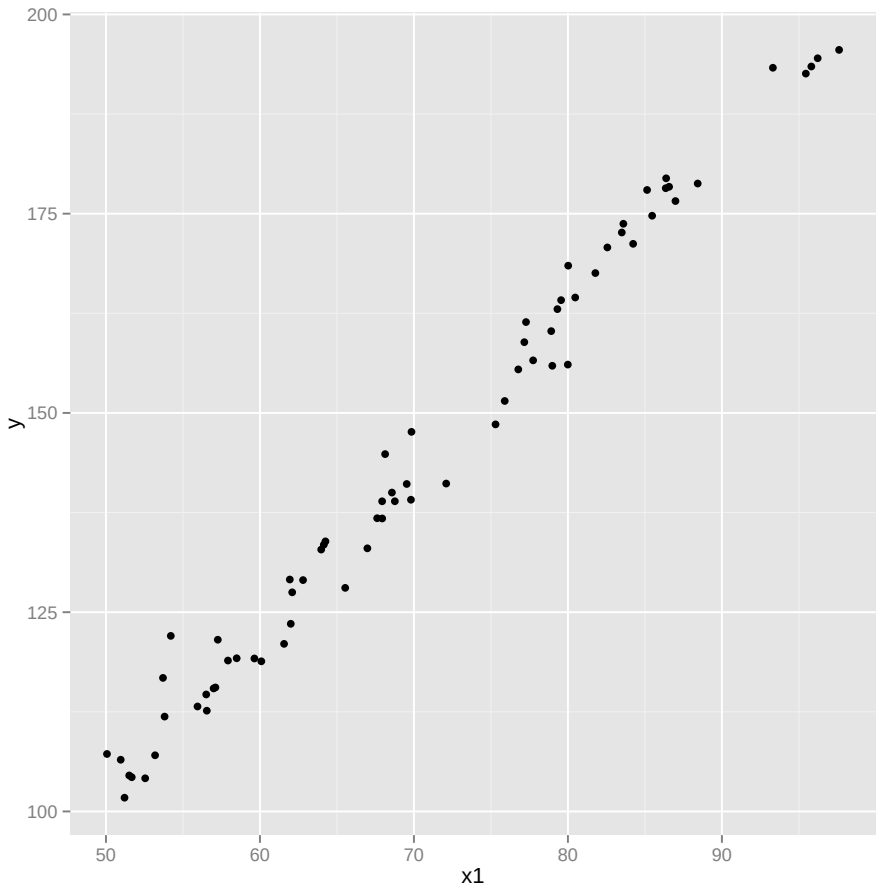
R contains a built in plotting commands *Plot*

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



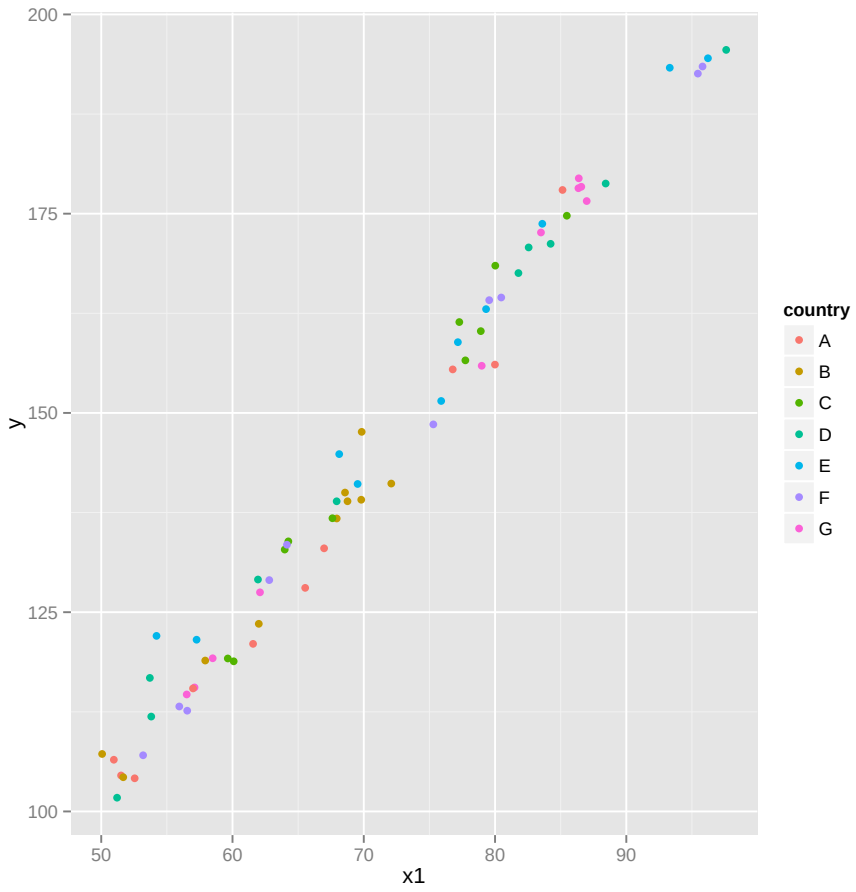

However the package *ggplot* provides additional features and so we will be using ggplot.

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




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

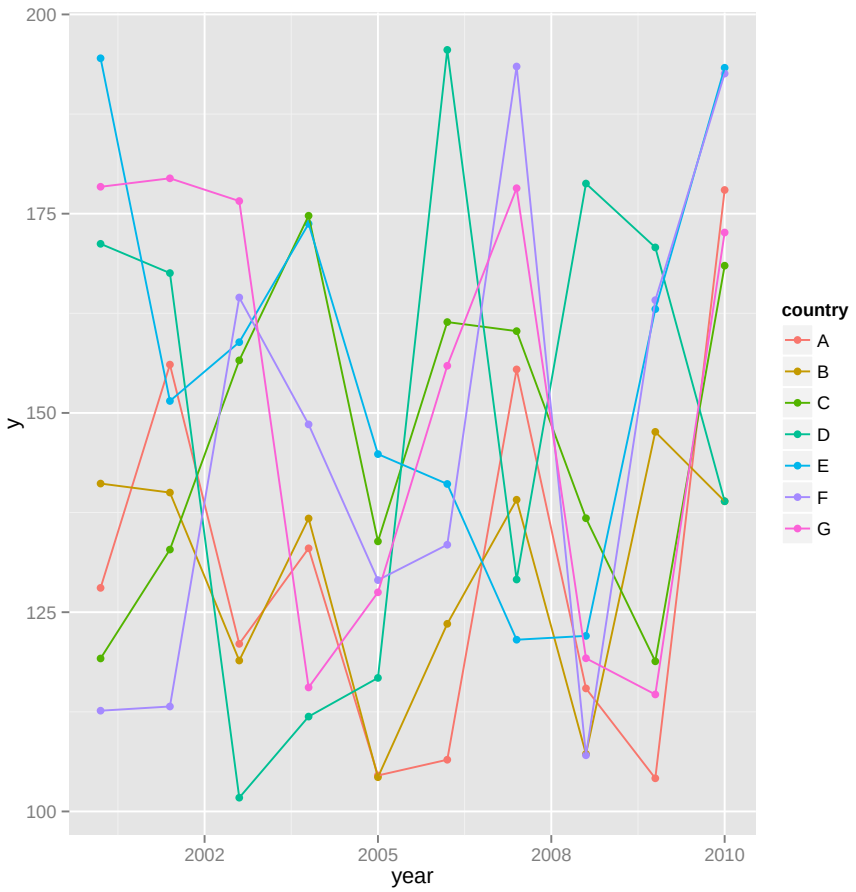
```
library(ggplot2)
ggplot(panel) + geom_point(aes(x = x1, y = y, group = country,
    colour = country))
```




## 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.

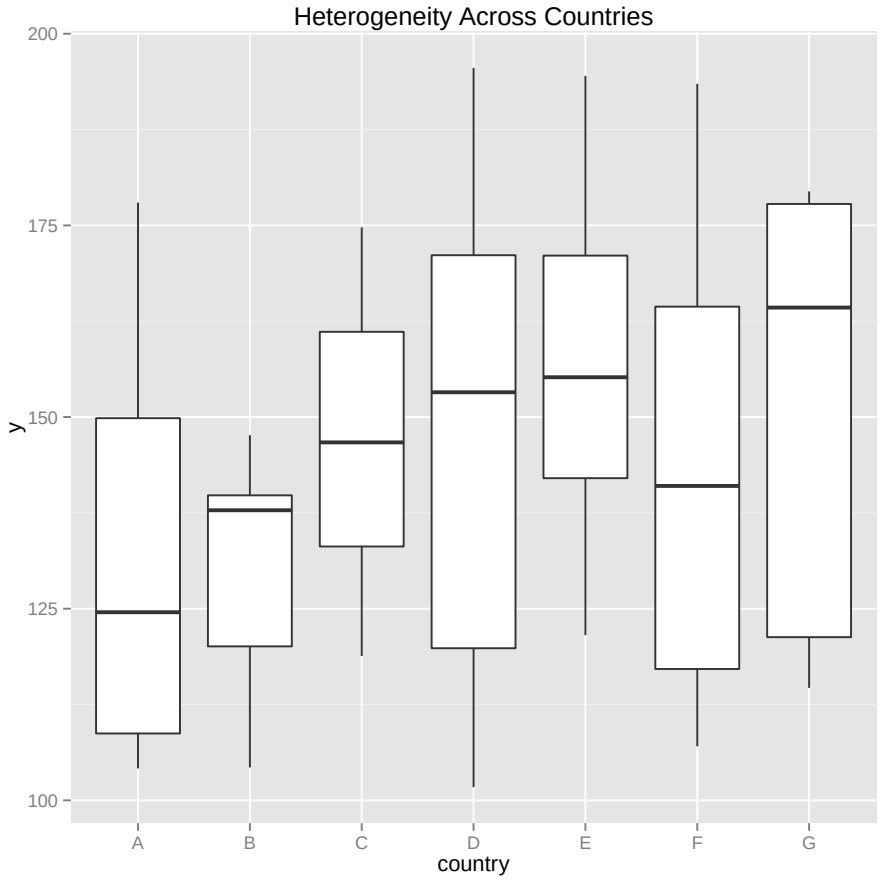
```
library(ggplot2)
ggplot(panel) + geom_point(aes(x = year, y = y, group = country,
    colour = country)) + geom_line(aes(x = year, y = y,
    group = country, colour = country))
```




## 4.4 Heterogeneity Across Countries

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

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



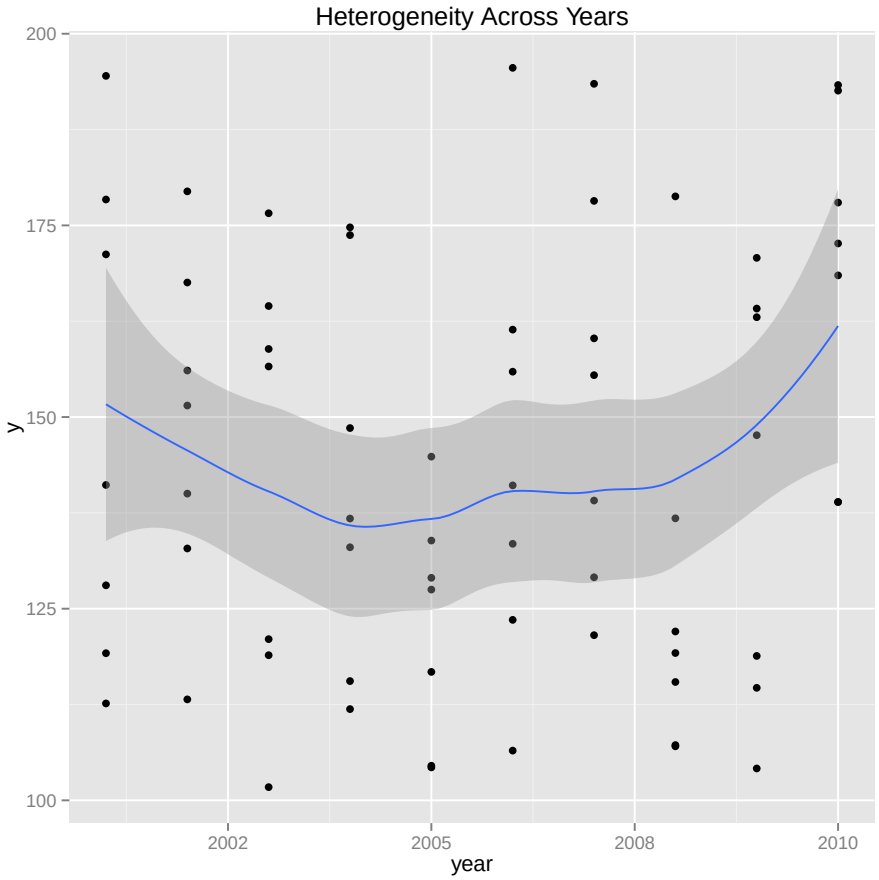

The parameters of each part of a boxplot are determined by various statistics. The middle bar is the 50% percentile, the bottom and top of the box are the 25% and 75% percentiles, etc.

## 4.5 Heterogeneity Across Years

We can also observe the Heterogenity accross 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.
```

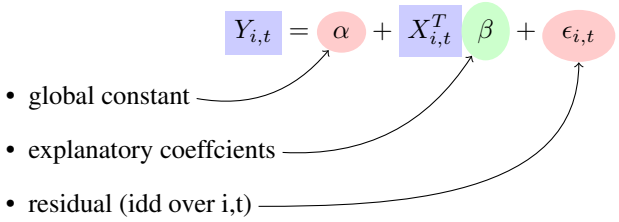



Here the shade area corresponds to a 95% confidence level.

# 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




(5.1)

where, as in eqn 2.2, we assume:

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

The *pooled model* is formed by pooling all the observations together and then performing an ordinary least squares regression (OLS). See the Appendix B for a discussion on Ordiniary Least Squares Regression.

Again, the form for the general equation for pooled panel data becomes

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

This may easily solved first by reading in the panel data frame:

```
df <- read.csv("./Data/Pooled-1.csv")
head(df)

##   country year     x     y
## 1       A 2001 65.54 209.4
## 2       B 2001 72.10 150.7
```

```
## 3       C 2001 59.64 131.4
## 4       D 2001 84.23 227.9
## 5       E 2001 96.22 207.9
## 6       F 2001 56.55 104.8
```

Then, as in Appendix B.1.2, we may solve using the *lm* command of the *stats* package:

```
fit <- lm(y ~ x, df)
summary(fit)

##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -91.99 -27.99   3.55  24.49 107.30
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -49.558     26.951   -1.84     0.07 .
## x              2.806      0.375    7.48    2e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0...
##
## Residual standard error: 41.4 on 68 degrees of freedom
## Multiple R-squared:  0.451,Adjusted R-squared:  0.443
## F-statistic: 55.9 on 1 and 68 DF,  p-value: 1.96e-10
```

In this summary, the $\alpha$ value is given on the line beginning with *(Intercept) -49.5581* . The value of $\alpha$ is -49.5581. This value may also be obtained by

```
fit$coefficients[1]

## (Intercept)
##      -49.56
```

Similarly the value of $\beta$ is given on the line beginning with *x 2.8057* This value may also be obtained by

```
fit$coefficients[2]

##     x
## 2.806
```

An alternative way to solve this is by using the *plm* package as shown below:

```r
library(plm)
```

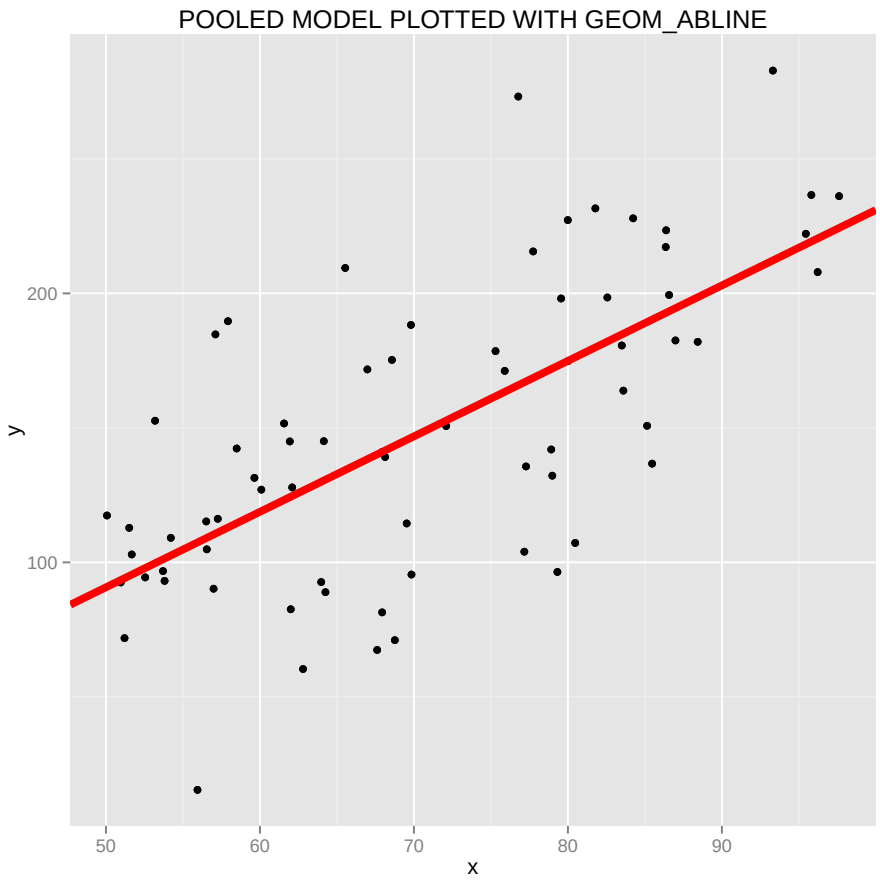
```
## Loading required package: bdsmatrix
##
## Attaching package: 'bdsmatrix'
## The following object is masked from 'package:base':
##
##    backsolve
## Loading required package: nlme
## Loading required package: Formula
## Loading required package: MASS
## Loading required package: sandwich
## Loading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

```r
df <- read.csv("./Data/Pooled-1.csv")
fit <- plm(y ~ x, data = df, index = c("country", "year"), model = "pooling")
summary(fit)
```

```
## Oneway (individual) effect Pooling Model
##
## Call:
## plm(formula = y ~ x, data = df, model = "pooling", ind...
##     "year"))
##
## Balanced Panel: n=7, T=10, N=70
##
## Residuals :
##    Min. 1st Qu.  Median 3rd Qu.    Max.
##  -92.00  -28.00    3.55   24.50  107.00
##
## Coefficients :
##             Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  -49.558     26.951   -1.84     0.07 .
## x              2.806      0.375    7.48    2e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0...
##
## Total Sum of Squares:    212000
## Residual Sum of Squares: 116000
## R-Squared      :  0.451
##       Adj. R-Squared :  0.438
## F-statistic: 55.8911 on 1 and 68 DF, p-value: 1.96e-10
```

This may be plotted by

```
library(ggplot2)
p <- ggplot(data = df) + geom_point(aes(x = x, y = y))
p <- p + geom_point(aes(x = x, y = y))
p <- p + geom_abline(intercept = fit$coefficients[1],
    slope = fit$coefficients[2], size = 2, colour = "red")
p <- p + ggtitle("POOLED MODEL PLOTTED WITH GEOM_ABLINE")
p
```




Alternatively, we may fit and plot in a single step

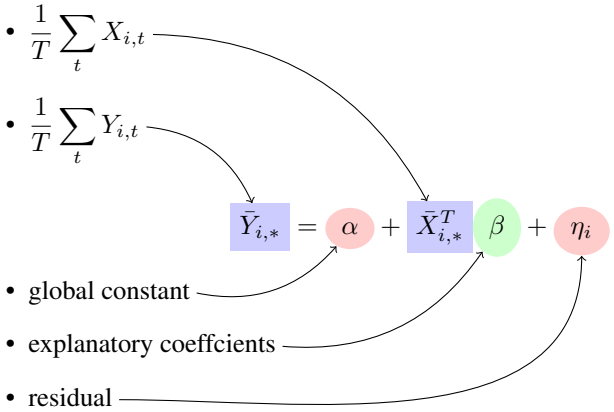
```
panelData <- read.csv("./Data/Pooled-1.csv")
p <- ggplot(data = panelData, aes(x = x,
    y = y))
p <- p + geom_smooth(method = "lm",
    se = FALSE, color = "red",
```

```
    formula = y ~ x)
p <- p + geom_point()
p <- p + ggtitle("POOLED MODEL PLOTTED WITH GEOM_SMOOTH")
p
```

# Chapter 6

# The Between Model

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




(6.1)

**Note:** In this model, t has been "averaged" out. Thus we used $\eta_i$ rather than the usual $\epsilon_{i,t}$. Moreover, if we introduce an individual effect, $\delta_i$ [1], we are then averaging t over

$$Y_{i,t} = \alpha + X^T_{i,t}\beta + \delta_i + \epsilon_{i,t} \tag{6.2}$$

which produces a residual of

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

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

[1] but keep the temporal effect, $\gamma_t$ equal to zero

### 6.0.1 The Between Estimator

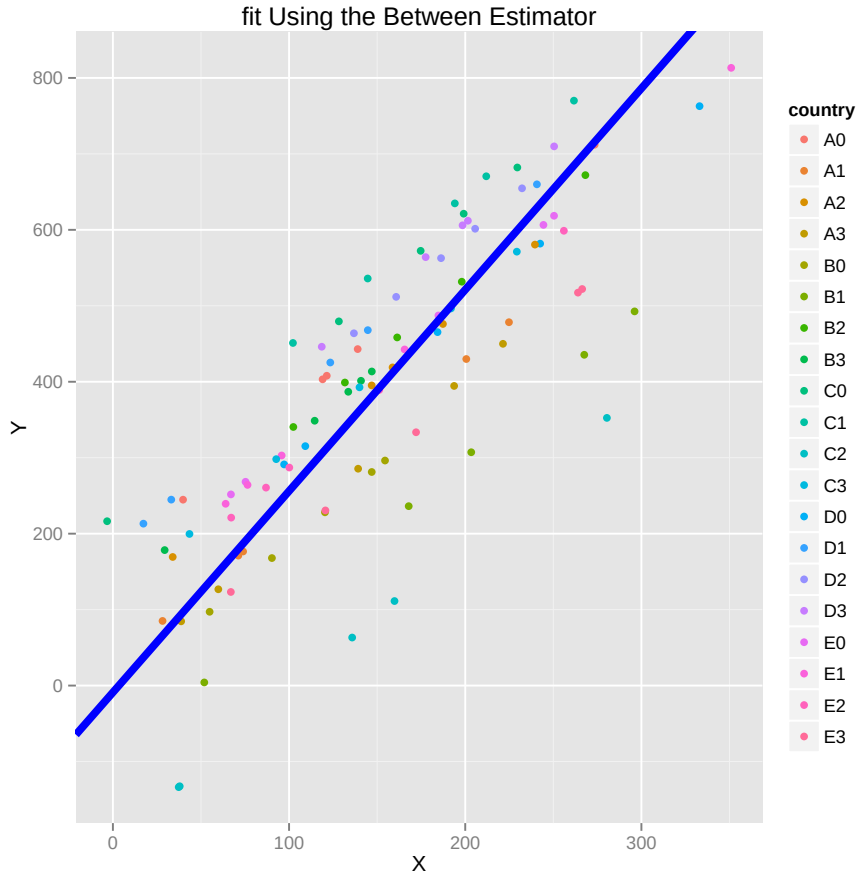
To perform *between* estimation we use the *plm* package, as follows:

```
panelData <- read.csv("./Data/Between-1.csv")
library(plm)
between <- plm(Y ~ X, data = panelData, index = c("country",
    "year"), model = "between")
summary(between)

## Oneway (individual) effect Between Model
##
## Call:
## plm(formula = Y ~ X, data = panelData, model = "betwee...
##     "year"))
##
## Balanced Panel: n=20, T=5, N=100
##
## Residuals :
##    Min. 1st Qu.  Median 3rd Qu.    Max.
##  -284.0   -47.4    21.0    80.4   137.0
##
## Coefficients :
##             Estimate Std. Error t-value Pr(>|t|)
## (Intercept)   -8.967    140.533   -0.06   0.9498
## X              2.651      0.909    2.92   0.0092 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0...
##
## Total Sum of Squares:    345000
## Residual Sum of Squares: 234000
## R-Squared      :  0.321
##       Adj. R-Squared :  0.289
## F-statistic: 8.50259 on 1 and 18 DF, p-value: 0.00922
```

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

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




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

```
alpha.between <- between$coefficients["(Intercept)"]
beta.between <- between$coefficients["X"]
pooled <- plm(Y ~ X, data = panelData, index = c("country",
    "year"), model = "pooling")
alpha.pooled <- pooled$coefficients["(Intercept)"]
beta.pooled <- pooled$coefficients["X"]
library(ggplot2)
p <- ggplot(panelData)
p <- p + geom_point(aes(x = X, y = Y, group = country,
    colour = country))
p <- p + geom_abline(intercept = alpha.between, slope = beta.between,
    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")
```

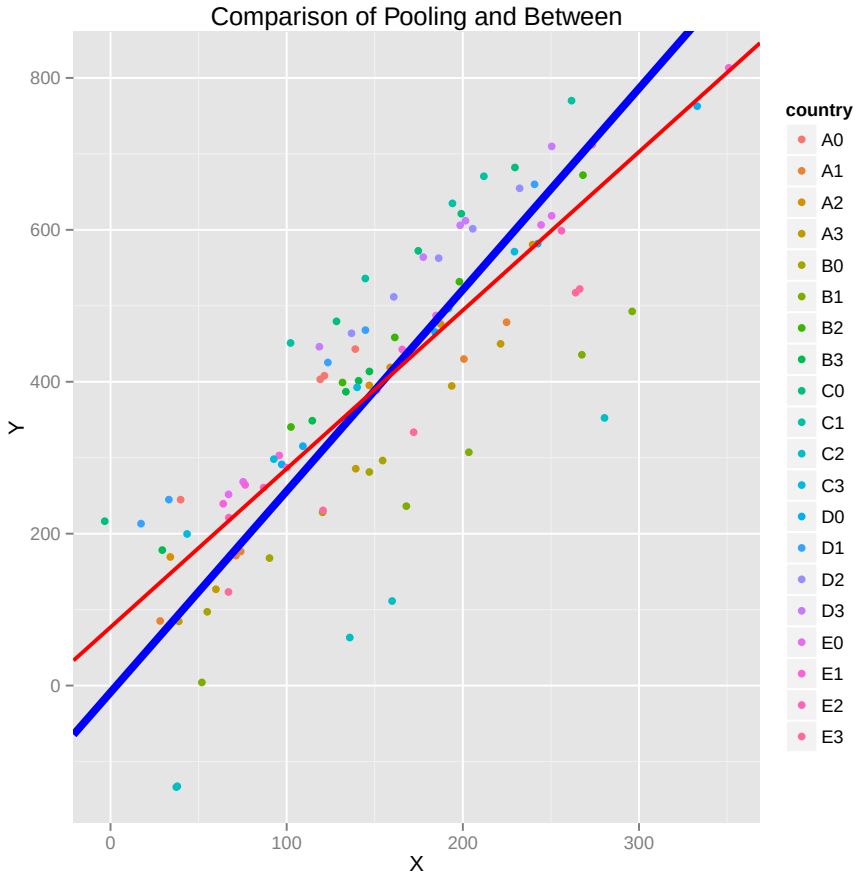



Figure 6.1: Graph with both Between and Pooled

```
p
```

# Chapter 7

# Fixed Effects Model

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

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

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

We may interpret each $\alpha_i$ as the intercept for that specific individual.

$$y_{i,t} = \alpha_i + x_{i,t}^T \beta + \epsilon_{i,t} \tag{7.2}$$

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 that 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 addin dummy variables $d1$, $d2$ to get table 7.2

Next we find the solution to

$$\hat{\theta} = argmin_\theta \|\vec{Y} - M\vec{\theta}\| \tag{7.3}$$

| | 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

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

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)
```

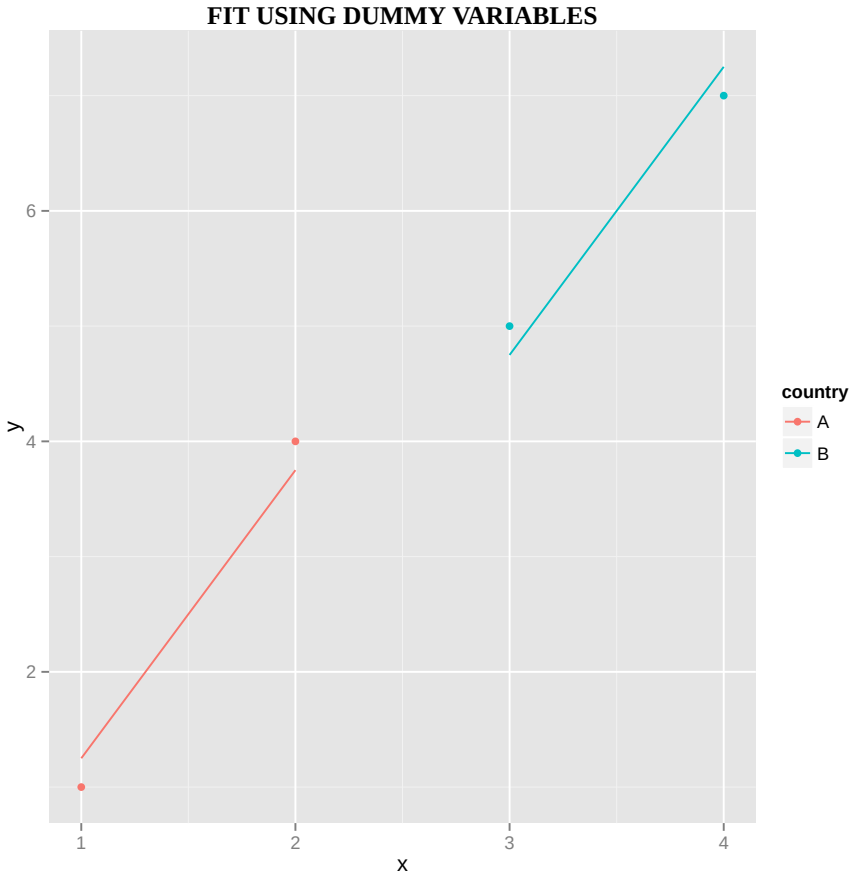
**Note:** The $-1$ in "$fixed.dum \leftarrow lm(y \sim x + d1 + d2 - 1, data = df)$" is to tell the *lm* not to add a column of one's (which is the default when doing ordinary regression with a single intercept.)

```
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 dummary variables need not be insert, since we can use the country names as factors. We illustrate this in the following example:

```
panel <- read.csv("./Data/FE-1.csv")
fixed.dum <- lm(y ~ x + factor(country) - 1, data = panel)
summary(fixed.dum)

##
## Call:
## lm(formula = y ~ x + factor(country) - 1, data = panel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.837 -2.146 -0.188  2.321 11.849
```

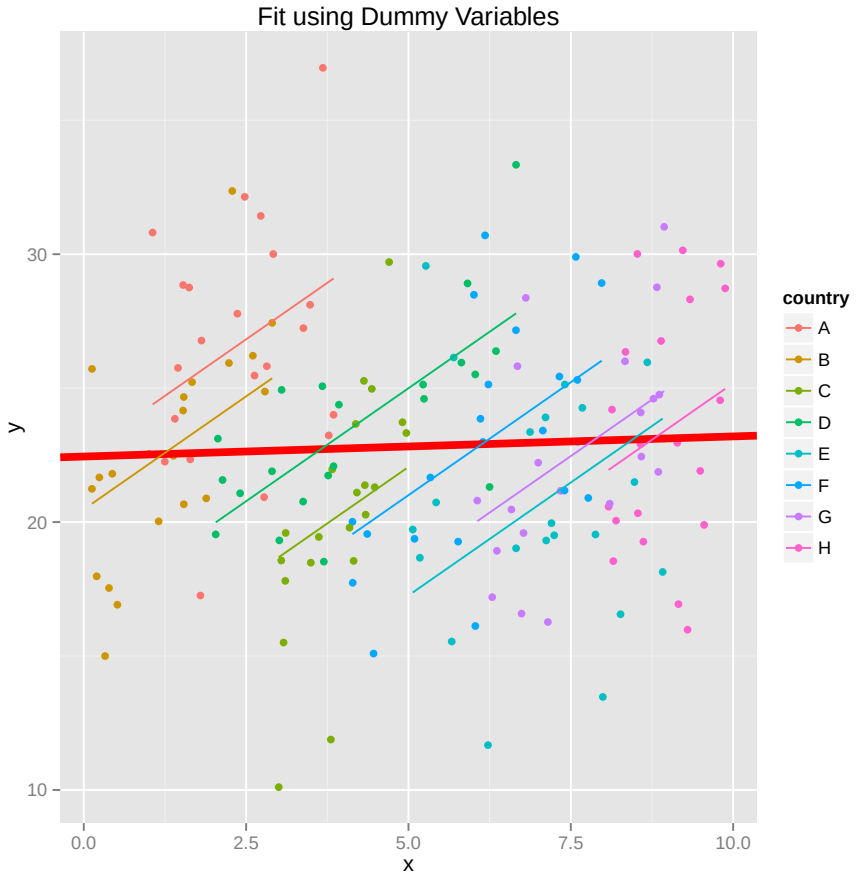
```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## x                    1.687      0.287    5.88  2.4e-08 ***
## factor(country)A    22.604      1.082   20.89  < 2e-16 ***
## factor(country)B    20.475      0.911   22.47  < 2e-16 ***
## factor(country)C    13.623      1.410    9.66  < 2e-16 ***
## factor(country)D    16.562      1.456   11.37  < 2e-16 ***
## factor(country)E     8.826      2.163    4.08  7.1e-05 ***
## factor(country)F    12.570      1.955    6.43  1.4e-09 ***
## factor(country)G     9.812      2.330    4.21  4.2e-05 ***
## factor(country)H     8.314      2.697    3.08   0.0024 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0...
##
## Residual standard error: 3.82 on 159 degrees of freedom
## Multiple R-squared:  0.974,Adjusted R-squared:  0.973
## F-statistic:  674 on 9 and 159 DF,  p-value: <2e-16
```

To compare to the pooled ols we issue the command

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

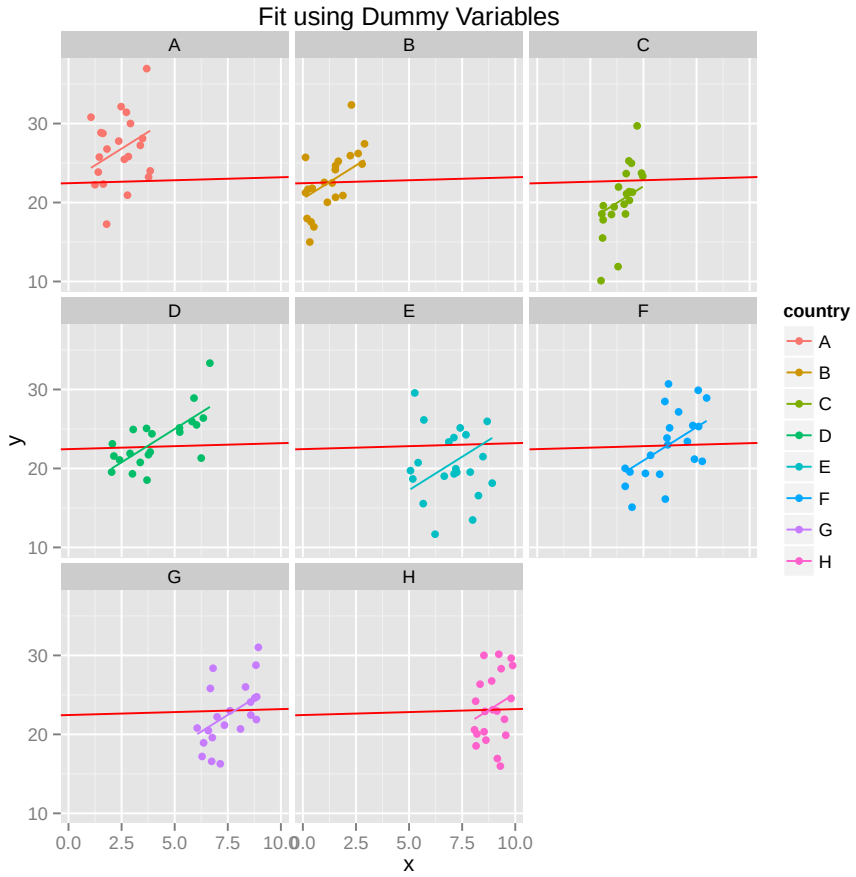
and then plot using

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




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

```
panel$yhat <- yhat
p <- ggplot(panel)
p <- p + geom_point(aes(x = x, y = y, group = country,
    colour = country))
p <- p + geom_abline(intercept = ols$coefficients[1],
    slope = ols$coefficients[2], colour = "red")
p <- p + geom_line(aes(x = x, 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 perfered 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

$$\sum_t yi, t = T\alpha + \sum_t X_{i,t}\beta + T\delta_i + \sum_t \epsilon_{i,t} \tag{7.5}$$

Dividing by T and subtracting from equations we have

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

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

ordinarly least squares estimation applied to equation 7.6

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

```
panelData <- read.csv("./Data/FE-1.csv")
library(plm)
fixed.within <- plm(y ~ x, data = panelData, index = c("country",
    "year"), model = "within")
summary(fixed.within)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = y ~ x, data = panelData, model = "within...
##     "year"))
##
## Balanced Panel: n=8, T=21, N=168
##
## Residuals :
##    Min. 1st Qu.  Median 3rd Qu.    Max.
##  -8.840  -2.150  -0.188   2.320  11.800
##
## Coefficients :
##   Estimate Std. Error t-value Pr(>|t|)
## x    1.687      0.287    5.88  2.4e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0...
##
## Total Sum of Squares:    2830
## Residual Sum of Squares: 2320
## R-Squared      :  0.179
##       Adj. R-Squared :  0.169
## F-statistic: 34.5517 on 1 and 159 DF, p-value: 2.36e-08
```

We can also get the $\alpha_i$ by

```
fixef(fixed.within)
```

```
##      A      B      C      D      E      F      G      H
## 22.604 20.475 13.623 16.562  8.826 12.570  9.812  8.314
```

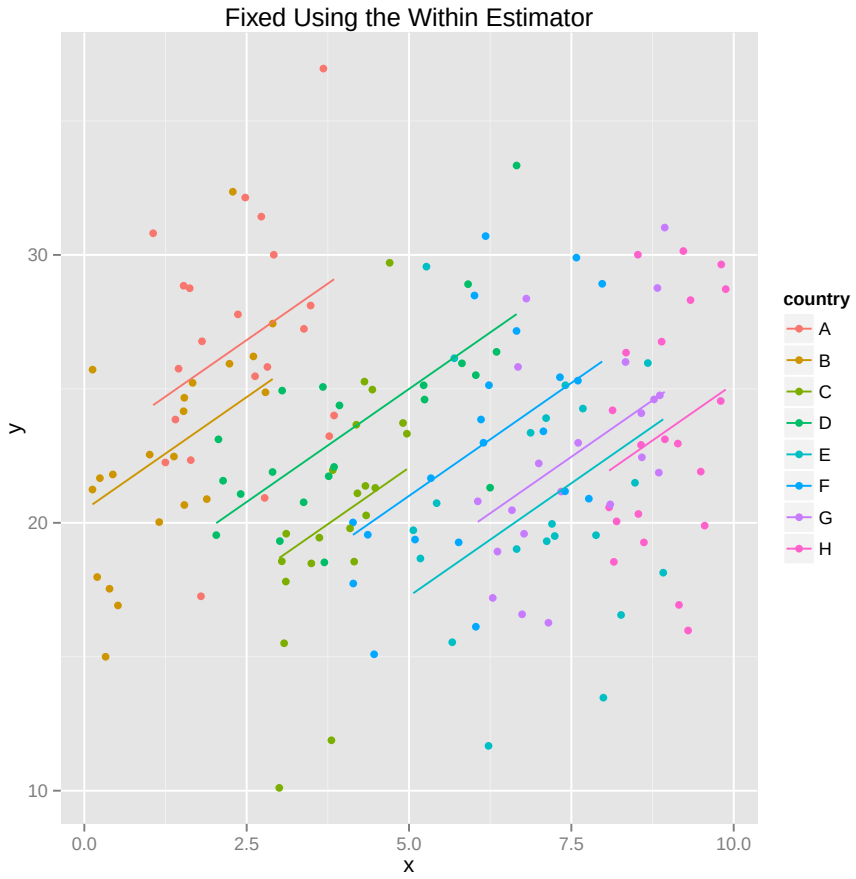
To graph in the same as before we need to do a little work to created an analogous "fitted" function, which is called hat below.

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

```
    b[i] + m * x
}
```

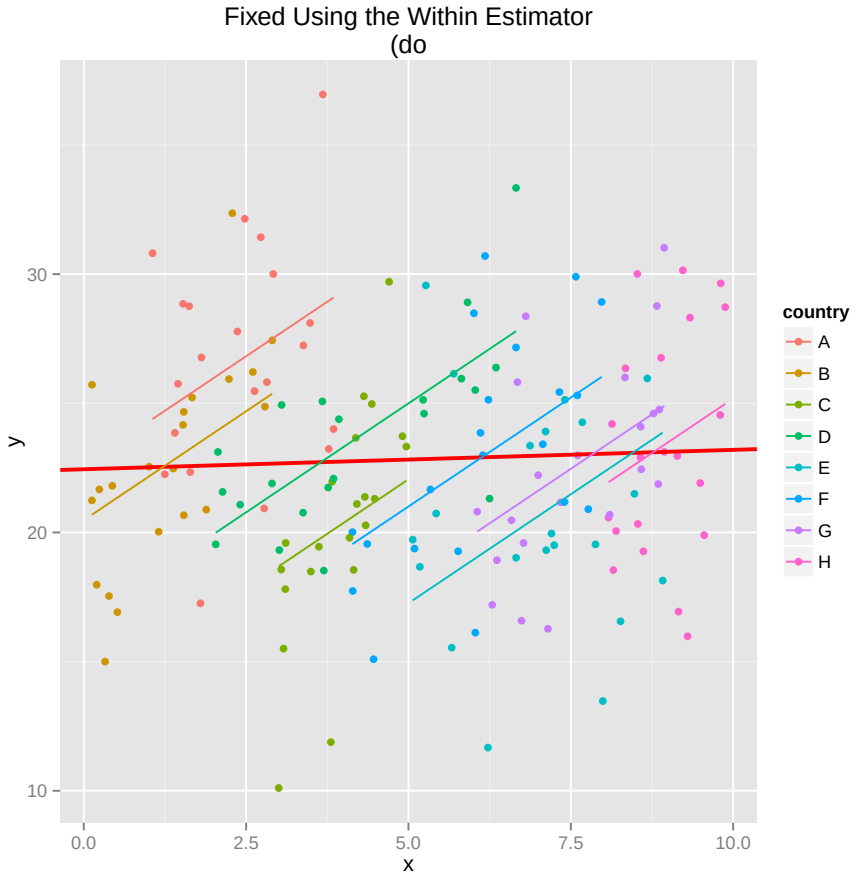
Then we apply hat to get yhat and proceed as before.

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



## 7.3 Fixed vs Pooling Models

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

```
panelData <- read.csv("./Data/FE-1.csv")
library(plm)
model.fixed <- plm(y ~ x, data = panelData, index = c("country",
    "year"), model = "within")
model.pooled <- plm(y ~ x, data = panelData, index = c("country",
    "year"), model = "pooling")
pFtest(model.fixed, model.pooled)

##
##  F test for individual effects
##
## data:  y ~ x
## F = 10.45, df1 = 7, df2 = 159, p-value = 9.233e-11
## 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.

# 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 $\delta_i$ that is uncorrelated with $X_{i,t}$, then since $\delta_i$ 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:

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

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

Here, our assumptions are:

- The period effect $\gamma_t$ statisfies $\gamma_t = 0$ for each t.
- The residual $\epsilon_{i,t}$ satisfies $\epsilon_{i,t} \sim IDD(0, \sigma_\epsilon^2)$
- $\delta_i$ is uncorrelated with $\epsilon_{i,t}$, that is $\rho(\delta_i, \epsilon_{i,t}) = 0$
- $\delta_i$ is uncorreltated 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 $E\left[\zeta_{i,t}\zeta_{j,s}\right]$ becomes

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

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

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

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

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

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

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

In practice, $\Omega$ is not known, but must be estimated. Since $\Omega$ is expressed in terms of $\sigma_\delta^2$ and $\sigma_\zeta^2$, this may be accomplished by estimation of $\sigma_\delta^2$ and $\sigma_\zeta^2$. 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 reading in data.[1]

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 ...
##     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
```

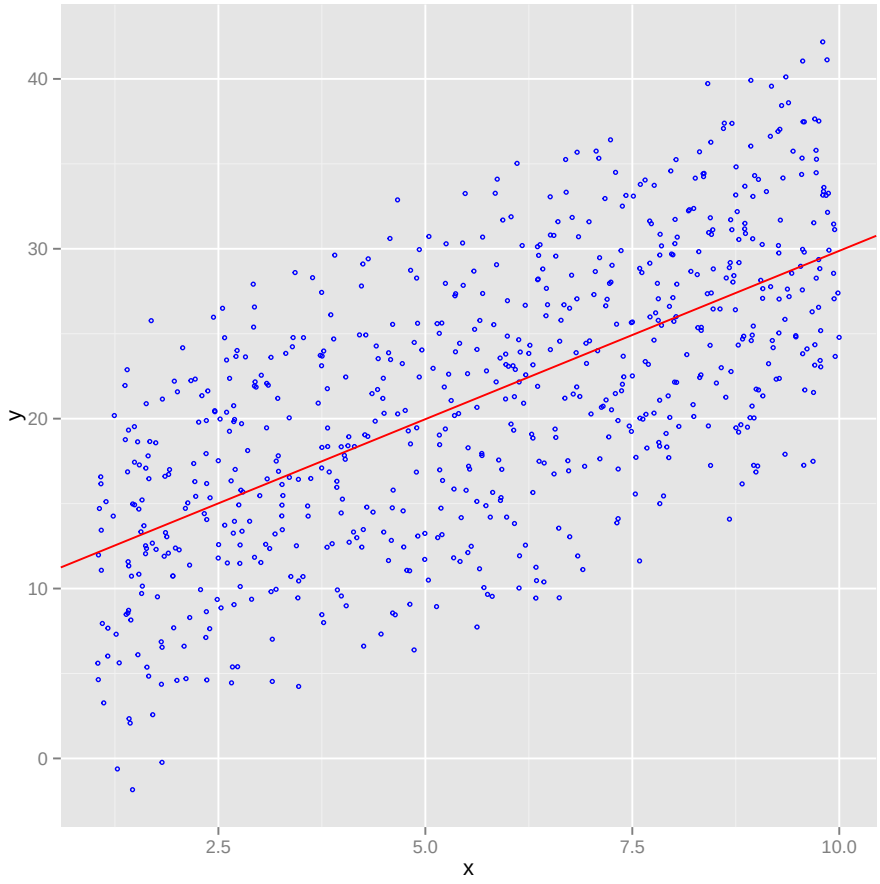
[1] To see how this data was generated see the Appendix A.4

```
##
## 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...
##
## 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 that .05 means that the 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 Multipler Test

From the looks of the graph, one might wonder if pooling might be the 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 performan 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 *phtest* 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", )
phtest(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 that $0.05$ then the *Fixed Model* is perfered. In this case we see that the *Random Effects* model is the perferred model

However, the *phtest* 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")
phtest(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 14.416
## 2      Ba 2008 1.652 17.81 17.588
## 3      Ca 2008 8.784 19.21 16.893
## 4      Da 2008 8.261 34.16  1.132
## 5      Ea 2008 8.947 20.74  9.820
## 6      Fa 2008 6.209 12.55 13.591
```

and the 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, e...
##     model = "random", random.method = "swar")
##
## Balanced Panel: n=234, T=3, N=702
##
## Effects:
##                 var std.dev share
## idiosyncratic  7.71    2.78   0.2
## individual    31.69    5.63   0.8
## theta:  0.726
##
## Residuals :
##    Min. 1st Qu.  Median 3rd Qu.    Max.
```

```
## -7.7100 -2.1000 -0.0234  1.9700  6.6700
##
## Coefficients :
##             Estimate Std. Error t-value Pr(>|t|)
## (Intercept)  10.2088     0.5087   20.07   <2e-16 ***
## x             1.9831     0.0468   42.34   <2e-16 ***
## z            -0.0155     0.0216   -0.72     0.47
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0...
##
## Total Sum of Squares:    19200
## Residual Sum of Squares: 5380
## R-Squared      :  0.72
##       Adj. R-Squared :  0.716
## F-statistic: 896.54 on 2 and 699 DF, p-value: <2e-16
```

**Note** For z, the value of $Pr(>|t|)$ is $0.67$ This exceeds $0.05$ which by conventional wisdom, indicates that z should be eliminated as an explanatory variable.

# Chapter 9

# Instrument Variables

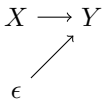
In the previous chapters we assumed that our linear model was exogenous. We now consider what happens when that assumption is relaxed. But first we give a definition:

**Definition 3: Endogenous**

The explanatory term $Xn_{i,t}$ is said to be *endogenous* provided it is correltated with $\sigma_{i,t}$ , that is $\rho(X_{i,t}, \epsilon_{i,t}) \neq 0$.

## 9.1 Bias Arrising From Endogenity 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

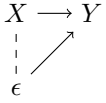



Here the arrows illustrate Y dependence on $X$ and $\epsilon$.

Now the assumption that $X$ and $\epsilon$ are uncorrelate may not in practice be true, i.e. we may find ourselves in the situation where

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

We can illustrate the correlation between $X$ and $\epsilon$ by connection them with a line as follows:



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

$$Y_i = \beta X_i + \epsilon \tag{9.2}$$

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

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

Now recall

$$\sigma_X^2 = E(X^2) - \mu_X^2 \tag{9.4}$$

and

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

So

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

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

$$\hat{\beta} \to \beta + \frac{\sigma_X \sigma_\epsilon \rho_{x\epsilon}}{\sigma_X^2 + \sigma_\epsilon^2} \tag{9.7}$$

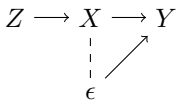
Now since $\sigma_X \neq 0$ [1] and $\sigma_\epsilon \neq 0$ we see that the necessary and sufficient condition for $\hat{\beta} \to \beta$ is that $\rho_{x\epsilon} \neq 0$.

## 9.2 Two Stage Least Squares

Thus, when $X$ and $\epsilon$ 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 $\epsilon$. 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

$$X = Z\alpha + \eta \tag{9.8}$$

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

$$\tilde{X} = Z\hat{\alpha} \tag{9.9}$$

[1] there is more that one X value

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

$$Y = \tilde{X}\beta + \epsilon \tag{9.10}$$

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

## 9.3 Combining the 2 Stage Regression Calculations

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

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

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

so $\tilde{X}$ is given by

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

For notational convenience, define P by

$$P = Z[Z^T Z]^{-1} Z^T \tag{9.13}$$

Then

$$\tilde{X} = PX \tag{9.14}$$

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

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

So

$$\hat{\beta} = [X^T P^T PX]^{-1} X^T P^T Y \tag{9.16}$$

This can be simplifed further by noting that P is symmetric, [2] to get

$$\hat{\beta} = [X^T P^2 X]^{-1} X^T PY \tag{9.17}$$

Furthermore, we note P is a projection, [3] thus

$$\hat{\beta} = [X^T PX]^{-1} X^T PY \tag{9.18}$$

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 $XZ^T (XZ^T)^{-1}$ in front of the Y of eq 9.18 to get

$$\hat{\beta} = [X^T PX]^{-1} X^T PXZ^T (XZ^T)^{-1} Y \tag{9.19}$$

---

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

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

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

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

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

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

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

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

## 9.4 Using Instrument Variables in R

Using the *plm package* we may incorporate instrument variables using the "|" symbol followed by a list of instruments. As an example, consider the data generated in Appendix ??? .

```
df = read.csv("./Data/InstrumentVbl.csv")
res <- plm(Y ~ X | Z, data = df, index = "Date", model = "pooling")
summary(res)

## Oneway (individual) effect Pooling Model
## Instrumental variable estimation
##    (Balestra-Varadharajan-Krishnakumar's transformation)
##
## Call:
## plm(formula = Y ~ X | Z, data = df, model = "pooling",...
##
## Balanced Panel: n=121, T=1, N=121
##
## Residuals :
##    Min. 1st Qu.  Median 3rd Qu.    Max.
## -38.300 -13.200  -0.637  12.800  62.200
##
## Coefficients :
##             Estimate Std. Error t-value Pr(>|t|)
## (Intercept)   0.8855     3.9079    0.23     0.82
## X             3.0487     0.0638   47.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0...
##
## Total Sum of Squares:    3880000
## Residual Sum of Squares: 49200
## R-Squared      :  0.996
##       Adj. R-Squared :  0.979
## F-statistic: 9275.59 on 1 and 119 DF, p-value: <2e-16
```

We may may compare this to the model without the instrument variable $Z$:

```
df = read.csv("./Data/InstrumentVbl.csv")
res2 <- plm(Y ~ X, data = df, index = "Date", model = "pooling")
summary(res2)

## Oneway (individual) effect Pooling Model
##
## Call:
## plm(formula = Y ~ X, data = df, model = "pooling", ind...
##
## Balanced Panel: n=121, T=1, N=121
##
## Residuals :
##    Min. 1st Qu.  Median 3rd Qu.    Max.
## -27.600  -8.980  -0.891   6.860  34.500
##
## Coefficients :
##             Estimate Std. Error t-value Pr(>|t|)
## (Intercept) -15.6800     1.5420   -10.2   <2e-16 ***
## X             3.3554     0.0203   165.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0...
##
## Total Sum of Squares:    3880000
## Residual Sum of Squares: 16900
## R-Squared      :  0.996
##       Adj. R-Squared :  0.979
## F-statistic: 27254.4 on 1 and 119 DF, p-value: <2e-16
```

Now the data in the panel "InstrumentVbl.csv" was generated using a value of $\beta = 3$ [4], we can ascertain the goodness of our estimates.

[4] See Appendix **??**

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

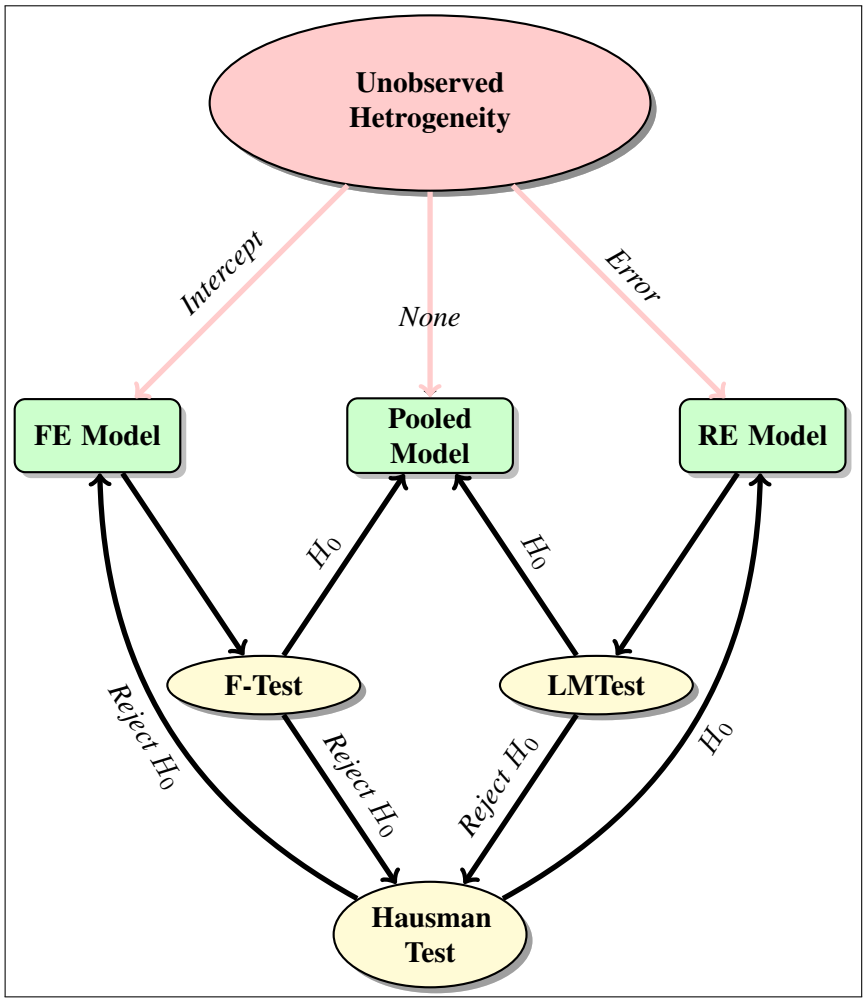



Figure 10.1: Pooled, Fixed Effects, Random Effects at a Glance

# Appendices

# Appendix A

# Synthetic Data Generation

## A.1 Pooled Models

### A.1.1 One-Way Pooled Model

In this section our goal is to create synthetic data for a typical pooled model:

$$y_{i,t} = \alpha + X_{i,t}^T \vec{\beta} + \epsilon_{i,t} \tag{A.1}$$

Data Specifications:

- Seven countries, denoted by the first seven capital letters.
- Assume 10 annual observations between the years 2001 and 2010.
- One explanatory, $X$ and one dependent variable $Y$.
- Set $\alpha = 60$
- Set $\beta = 2$
- Set the standard deviation of $\epsilon$ equal to 40 ($\sigma_\epsilon = 40$).
- randomly from the interval between $[50, 100]$ using a uniform random distribution.
- Save the result as a csv file in the subdirectory *"Data"* with filename of *"Pooled-1.csv"*.

```
set.seed(273)
country <- LETTERS[1:7]
years <- 2001:2010
panel <- expand.grid(country = country, year = years)
n <- nrow(panel)
```

```
panel$x <- runif(n, 50, 100)
panel$y <- panel$x * 2 + rnorm(n, mean = 0, sd = 50)
write.csv(panel, "./Data/Pooled-1.csv", row.names = FALSE)
```

### A.1.2 Two-way Pooled Model

```
set.seed(273)
country <- LETTERS[1:7]
years <- 2001:2010
panel <- expand.grid(country = country, year = years)
n <- nrow(panel)
panel$x1 <- runif(n, 50, 100)
panel$x2 <- runif(n, 50, 100)
panel$x3 <- runif(n, 50, 100)
panel$y <- panel$x1 * 2 + (panel$year - 2000) * 0.5 +
    rnorm(n, mean = 0, sd = 3)
write.csv(panel, "./Data/Pool-2way.csv", row.names = FALSE)
```

## A.2 Between Models

In this section our goal is to create synthetic data for a typical between model:

$$\bar{Y}_{i,*} = \alpha + \bar{X}_{i,*}\beta + \eta_i \tag{A.2}$$

where $\bar{X}$ and $\bar{Y}$ are time average values of $X$ and $Y$. That is, $\bar{X}_{i,*} = \frac{1}{T}\sum_i X_{i,t}$ and $\bar{Y}_{i,*} = \frac{1}{T}\sum_i Y_{i,t}$ .

We construct our model with the folowing Data Specifications:

- 260 countries, denoted by the capital letters followed by a digit.
- Assume 5 annual observations between the years 2001 and 2005.
- One explanatory, $X$ and one dependent variable $Y$.
- Set $\alpha = 60$
- Set $\beta = 2$
- For each i, pick $\bar{X}_{i,*}$ randomly from the interval between $[100, 200]$ using a uniform random distribution.
- For each i, Pick $\eta_i$ randomly from the normal distribution with $\sigma = 50$.

- For each i,t Pick $z_{i,t}$ from normal distribution with $\sigma = 20$
- For each i,t, set $x_{i,t} = z_{i,t} + \bar{X}_{i,*}$, thus making $E_t(x_{i,t}) \rightarrow X_{i,*}$
- Save the result as a csv file in the subdirectory *"Data"* with filename of *"Betweeen-1.csv"*.

```
set.seed(512)
alpha = 60
beta = 2
countries <- apply(expand.grid(LETTERS[1:2], 0:2),
    1, paste, collapse = "")
I <- length(countries)
X.bar <- runif(I, 100, 200)
eta <- rnorm(I, 50)
names(eta) <- countries
years <- 2001:2005
# form matrix z,whose indices are country, time,
# with mean 0
z <- t(sapply(countries, function(x) {
    a <- runif(4, -50, 50)
    c(a, -sum(a))
}))
x <- z + X.bar  #add X.bar to each column of z
colnames(x) <- years  #to make indexing easier
panel <- expand.grid(country = countries, year = years)
# pc<-as.character(panel$country)
# yr<-as.character(panel$year)
panel$X <- apply(panel, 1, function(r) {
    x[r["country"], r["year"]]
})
panel$Y <- 2 * panel$X + eta[as.character(panel$country)]
write.csv(panel, "./Data/Between-1.csv", row.names = FALSE)
```

## A.3 Fixed Effects Model

In this section our goal is to create synthetic data for a Fixed effects model:

$$y_{i,t} = \alpha_i + x_{i,t}^T \beta + \epsilon_{i,t} \tag{A.3}$$

We break $\alpha_i$ into two components:$\alpha$ and $\delta_i$. Since $delta_i$ is indexed by country, in the code below we form a array $LDelta$ indexed by countries. Although not necessary, we additionaly create a range of valid X values on a per country basis. This example produces a $\beta = 2$

```
country <- LETTERS[1:8]
T <- 1990:2010
panel <- expand.grid(country = country, year = T)
LRange <- list(c(1, 4), c(0, 3), c(3, 5), c(2, 7),
    c(5, 9), c(4, 8), c(6, 9), c(8, 10))
names(LRange) <- country
LDelta <- c(16, 15, 8, 11, 2, 5, 3, -1)
# LDelta<-c(0,0,0,0,0,0,0,0)
names(LDelta) <- country
panel$x <- sapply(panel$country, function(cn) {
    runif(1, LRange[[cn]][1], LRange[[cn]][2])
})
delta <- sapply(panel$country, function(cn) LDelta[cn])
alpha <- 5
beta <- 2
epsilon <- rnorm(nrow(panel), sd = 4)
# epsilon<-0
panel$y <- alpha + delta + beta * panel$x + epsilon
# panel$y<-panel$x + epsilon
write.csv(panel, "./Data/FE-1.csv", row.names = FALSE)
# panel<-NULL
```

```
head(panel)
```

```
##   country year     x     y
## 1       A 1990 2.917 30.01
## 2       B 1990 1.007 22.55
## 3       C 1990 3.002 10.11
## 4       D 1990 5.815 25.95
## 5       E 1990 8.266 16.56
## 6       F 1990 4.137 20.01
```

## A.4 Random Effects Model

```
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
```

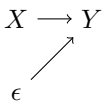
```
df$y <- alpha[df$country] + df$x * 2 + runif(nrow(df),
    -5, 5)
write.csv(df, "./Data/Random-1.csv", row.names = FALSE)
df <- NULL
```

The motivation here is to generate data with many countries having randomly distributed $\delta_i$ 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 date to satisfy

$$y_{i,t} = 10 + 2 * x_{i,t} + (\delta_i + \epsilon_{i,t}) \tag{A.4}$$

## A.5 Instrument Variables Model

To generate appropriate synthetic data for an example of analysis using *instrument variables* we require $X$ to be correlated with $\epsilon$ and an *instrumnet variable*, $Z$ which is correlated with $X$.




One way of accomplishing this is to set $X$ to be a linear combination of $\epsilon$ and $Z$ with some additional noise $\eta$ added. That is:

```
set.seed(314)
dates <- seq(as.Date("2000/1/1"), as.Date("2010/1/1"),
    "months")
N <- length(dates)
Z <- runif(N, 0, 100)
epsilon <- rnorm(N, sd = 20)
eta <- rnorm(N, sd = 10)
X <- Z + 2 * epsilon + eta
Y = 3 * X + epsilon
df <- data.frame(Date = dates, Y = Y, X = X, Z = Z)
write.csv(df, "./Data/InstrumentVbl.csv", row.names = FALSE)
```

In this case,

# Appendix B

# Review of Ordinary Least Square Regression (OLS)

## B.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 ordinarly linear regression

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

$$y_i = \alpha + \sum_{j=1}^{P} \vec{x}_{i,j}\beta_j + \epsilon_i \tag{B.1}$$

$$y_i = \alpha + \vec{x}_i^T\vec{\beta} + \epsilon_i \tag{B.2}$$

Or in matrix form

$$\vec{y} = M\vec{\theta} + \vec{\epsilon} \tag{B.3}$$

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

### B.1.1 A Simple Example

Suppose we have observed the following data

and we want to represent the relationship between x and y using a single line, say

| | 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 B.1: Simple Example

$y = mx + b$. This corresponds to the systems of equations

$$\begin{array}{rcl} 1 & = & 1m + 1b \\ 4 & = & 2m + 1b \\ 5 & = & 3m + 1b \\ 7 & = & 4m + 1b \\ 8 & = & 5m + 1b \end{array} \tag{B.4}$$

or in matrix notation

$$\vec{Y} = M\vec{\theta} \tag{B.5}$$

where

$$\vec{Y} = \begin{vmatrix} 1 \\ 4 \\ 5 \\ 7 \\ 8 \end{vmatrix}, \; M = \begin{vmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{vmatrix}, \; \vec{\theta} = \begin{vmatrix} m \\ b \end{vmatrix} \tag{B.6}$$

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

$$\vec{Y} = M\vec{\theta} + \vec{\epsilon} \tag{B.7}$$

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

$$\widehat{\theta} = argmin_{\theta} \|\vec{Y} - M\vec{\theta}\| \tag{B.8}$$

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

$$M^T \left[\vec{Y} - M\widehat{\theta}\right] = \vec{0} \tag{B.9}$$

and so we have

$$M^T\vec{Y} = M^T M\widehat{\theta} \tag{B.10}$$

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

$$\widehat{\theta} = \left[M^T M\right]^{-1} M^T\vec{Y} \tag{B.11}$$

.

**Our Analysis**

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

$$\vec{Y_i} = \alpha + \sum_{j=1}^{P} X_{i,j}\beta_j + \epsilon_i \tag{B.12}$$

. or coding it into matrix notation

$$Y = M\theta + \vec{\epsilon} \tag{B.13}$$

, and we produced an algorithm to estimate $\theta$. Moreover, we implicitly assumed that the $\epsilon_i$ had

And

### B.1.2 Using lm to perform OLS

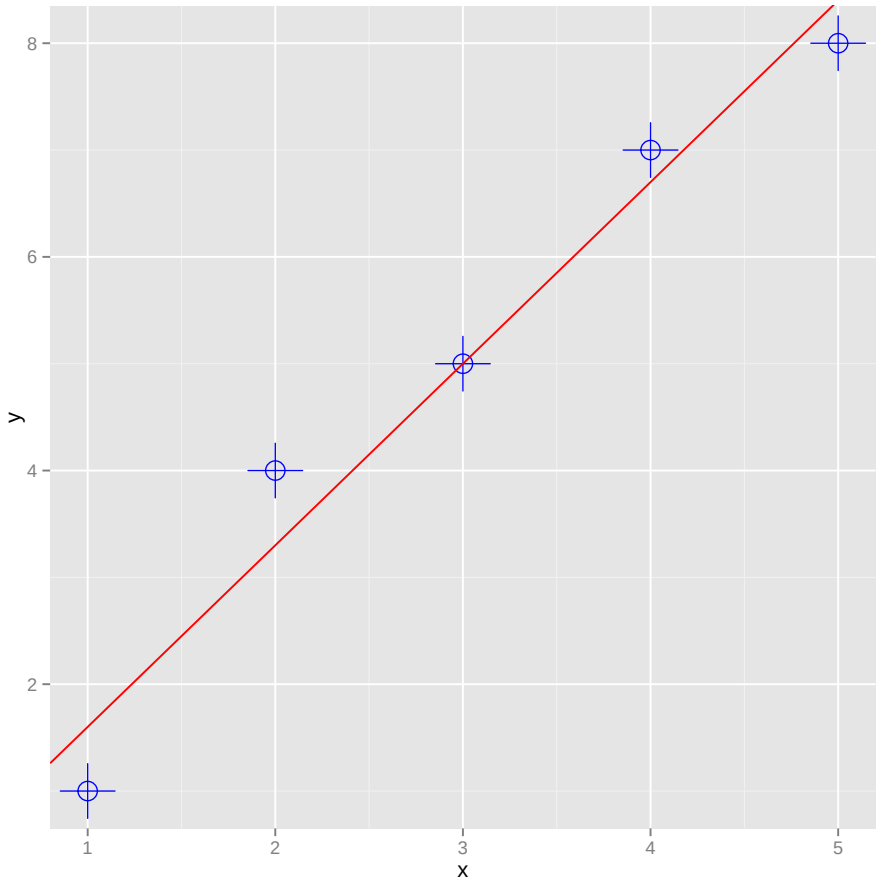
Using the *Stats* package we solve by issuing the *lm* (linear model) command.

```
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...
##
## 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")
```




## B.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 4: Linear Estimator**

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

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

**Definition 5: Best Linear Unbiased Estimator**

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

- *Linear in Y* The estimator is of the form $\widehat{\theta} = B^T \vec{Y}$.
- Unbiased $E\left[\widehat{\theta}\right] = \theta$
- 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 $Var(\epsilon) = \sigma^2 \vec{\iota}$. The the least square estimate $\widehat{\theta} = \left(M^T X\right)^{-1} \vec{Y}$ is the *Best Linear Unbiased Estimate* of $\theta$.

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

Second note that $E[\widehat{\theta} - \theta] = E\left[\left[M^T M\right]^{-1} M^T \vec{Y} - \theta\right] = \left[M^T M\right]^{-1} M^T E\left[\vec{Y}\right] - \theta$ and since $E[Y] = E[M\theta + \epsilon] = M\theta + E[\epsilon] = M\theta$. Thus $E[\widehat{\theta} - \theta] = \left[M^T M\right]^{-1} M^T M\theta - \theta = 0$. 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 $\widetilde{\theta}$ be another linear unbiased estimator. To show $var(\widetilde{\theta}) \geq \widehat{\theta}$. Since $\widetilde{\theta}$ is a linear estimator, it is of the form $\widetilde{\theta} = \sum c_i Y_i = CY$. Now $\widetilde{\theta} - \widehat{\theta} = \left[CY - (M^T M)^{-1} M^T Y\right] = \left[C - (M^T M)^{-1} M^T\right] Y$. Set $D = C - (M^T M)^{-1} M^T$, so $\widetilde{\theta} = CY = \left[(M^T M)^{-1} M^T + D\right] Y$ That is, $\widetilde{\theta} = \left[(M^T M)^{-1} M^T + D\right](M\theta + \epsilon)$. Taking the expectation and noting that $E[\epsilon] = 0$ we have $E[\widetilde{\theta}] = \left[(M^T M)^{-1} M^T + D\right] M\theta$. But $\widetilde{\theta}$ is unbiased, so $\left[(M^T M)^{-1} M^T + D\right] M\theta = \theta$, i.e. $(I + DM)\theta = \theta$. Thus $DM\theta = 0$ and since $D$ does not depend on $\theta$, $DM = 0$. Now computing the $var(\widetilde{\theta})$ we get $var(\widetilde{\theta}) = var(CY(CY)^T) = var(CYY^T C^T) = \sigma^2 var(CC^T)$. Now $CC^T = \left[(M^T M)^{-1} M^T + D\right]\left[(M^T M)^{-1} M^T + D\right]$. Multiplying out and dropping term containing $DM$ we have $CC^T = (M^T M)^{-1} + DD^T$, so $var(\widetilde{\theta}) = \sigma^2 (M^T M)^{-1} + \sigma^2 DD^T$. But $\sigma^2 (M^T M)^{-1} = var(\widehat{\beta})$ and $DD^T$ is positive semidefinite, so $var(\widehat{\theta}) \leq var(\widetilde{\theta})$ ■