Module 3: Multiple Regression

R Practical

Camille Szmaragd and George Leckie[1]
Centre for Multilevel Modelling

Pre-requisites box

- Modules 1-2

Contents

[1] This R practical is adapted from the corresponding MLwiN practical: Steele, F. (2008) Module 3: Multiple Regression MLwiN Practical. LEMMA VLE, Centre for Multilevel Modelling. Accessed at http://www.cmm.bris.ac.uk/lemma/course/view.php?id=13.

Some of the sections within this module have online quizzes for you to test your understanding. To find the quizzes:

EXAMPLE

From within the LEMMA learning environment

- Go down to the section for **Module 3: Multilevel Modelling**
- Click "3.1 Regression with a Single Continuous Explanatory Variable" to open Lesson 3.1
- Click Q 1 to open the first question

Pre-requisites

- Understanding of types of variables (continuous vs. categorical variables, dependent and explanatory); covered in Module 1.
- Correlation between variables
- Confidence intervals
- Hypothesis testing, p-values
- Independent samples t-test for comparing the means of two groups

Online resources:

http://www.sportsci.org/resource/stats/
http://www.socialresearchmethods.net/
http://www.animatedsoftware.com/statglos/statglos.htm
http://davidmlane.com/hyperstat/index.html

The aim of these exercises is to gain practical experience of the application and interpretation of multiple regression.

Introduction to the Scottish Youth Cohort Trends Dataset

You will be analysing data from the Scottish School Leavers Survey (SSLS), a nationally representative survey of young people. We use data from seven cohorts of young people collected in the first sweep of the study, carried out at the end of the final year of compulsory schooling (aged 16-17) when most sample members had taken Standard grades.[2] These are subject-based examinations, typically taken in up to eight subjects. Each subject is graded on a scale from 1 (highest) to 7 (lowest). The dependent variable is a total attainment score calculated by assigning 7 points for a '1', 6 for a '2' and so on.

The analysis dataset contains the following five variables:

Variable name	Description and codes
Caseid	Anonymised student identifier
Score	Point score calculated from awards in Standard grades. Scores range from 0 to 75, with a higher score indicating a higher attainment
cohort90	The sample includes the following cohorts: 1984, 1986, 1988, 1990, 1996 and 1998. The **cohort90** variable is calculated by subtracting 1990 from each value. Thus values range from -6 (corresponding to 1984) to 8 (1998), with 1990 coded as zero
Female	Sex of student (1 = female, 0 = male)
Sclass	Social class, defined as the higher class of the mother or father (1 = managerial and professional, 2 = intermediate, 3 = working, 4 = unclassified)

There are 33,988 students in the dataset.

[2] We are grateful to Linda Croxford (Centre for Educational Sociology, University of Edinburgh) for providing us with these data. The dataset was constructed as part of an ESRC-funded project on Education and Youth Transitions in England, Wales and Scotland 1984-2002. Further analyses of the data can be found in Croxford and Raffe (2006).

P3.1 Regression with a Single Continuous Explanatory Variable

We will begin by looking at the relationship between attainment (**score**) and cohort (**cohort90**). Has attainment changed over time and, if so, is the trend linear?

P3.1.1 Examining the data

Download the R dataset for this lesson:

From within the LEMMA Learning Environment
- Go to **Module 3: Multiple Regression**, and scroll down to **R Datasets and R files**
- Right click "3.1.txt" and select **Save Link As…** to save the dataset to your computer

Read the dataset into R using the `read.table` function and create a dataframe object named **mydata:**[3]

```
> mydata <- read.table(file = "3.1.txt", sep = ",", header = TRUE)
```

We specify the `sep = ","` argument as the file is a comma separated variable file while we specify the `header = TRUE` argument as the first line of this file contains the names of the variables. If we did not specify this second argument, the variable names would be incorrectly treated as the first row of data in the dataframe.

Use the `dim` function to display the number of rows of data and the number of columns of variables in the dataframe:

```
> dim(mydata)
[1] 33988     5
```

and use the `str` function to produce a basic description of the dataframe, which includes some general information on the number of variables, along with a list of the variables and their first few values:

```
> str(mydata)
'data.frame':   33988 obs. of  5 variables:
 $ caseid  : int  339 340 345 346 352 353 354 361 362 363 ...
```

[3] At the beginning of your R session, you will need to set R's working directory to the file location where you saved the dataset. This can be done using the command line and the `setwd` command:

```
> setwd("C:\userdirectory\")
```

Or through selecting Change Dir… on the File menu.

```
$ score   : int  49 18 46 43 17 29 15 19 45 12 ...
$ cohort90: int  -6 -6 -6 -6 -6 -6 -6 -6 -6 -6 ...
$ female  : int  0 0 0 0 0 0 0 0 0 0 ...
$ sclass  : int  2 3 4 3 3 2 3 2 3 1 ...
```

The dataframe contains 33,988 observations on 5 variables.

Dataframes may be displayed in matrix form. You can view subsets of the dataframe using standard matrix indexing conventions. Here we specify rows `1:20` to display only the first 20 rows of observations in the data. Note, we have not specified which columns we wish to display and so the values of all the variables are displayed:

```
> mydata[1:20, ]
   caseid score cohort90 female sclass
1     339    49       -6      0      2
2     340    18       -6      0      3
3     345    46       -6      0      4
4     346    43       -6      0      3
5     352    17       -6      0      3
6     353    29       -6      0      2
7     354    15       -6      0      3
8     361    19       -6      0      2
9     362    45       -6      0      3
10    363    12       -6      0      1
11   6824     0       -4      0      1
12   6826     0       -4      0      3
13   6827    20       -4      0      2
14   6828    32       -4      0      1
15   6829     0       -4      0      2
16   6834    24       -4      0      3
17   6836    23       -4      0      2
18  13206     7       -2      0      3
19  13209    38       -2      0      3
20  13215    46       -2      0      1
```

For example, the 10th student in the data belongs to the 1984 cohort and scored 12 out of 75. This student is a boy from a managerial social class background.

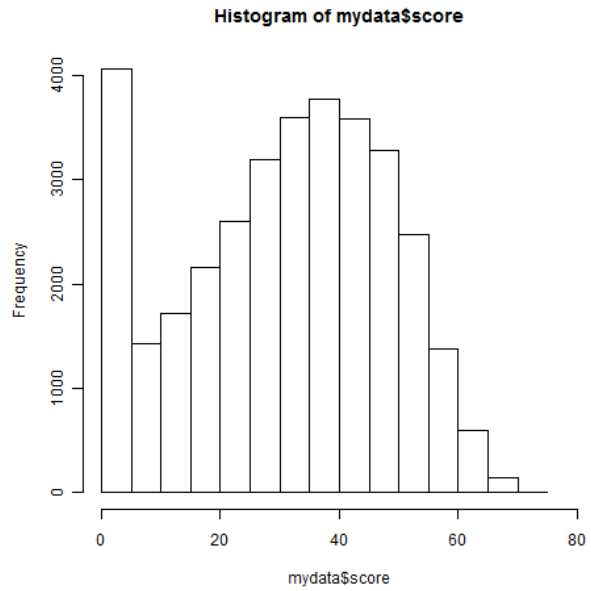
Having viewed the data we will examine **score** and **cohort90**, the variables to be considered in our first regression analysis.

Distribution of score

We will begin by obtaining a histogram and descriptive statistics for the dependent variable, **score**. We obtain a histogram with the `hist` command:

```
> hist(mydata$score, xlim = c(0,80))
```

where we have referred to the **score** variable within the **mydata** dataframe as `mydata$score`



The histogram should look like the above figure. The `xlim` argument is used to scale the x-axis from zero to 80. Apart from a peak at around zero, the distribution looks approximately normal. Remember that in a linear regression model it is the residuals that are assumed to be normal; we will check this assumption at the end of the exercise.

The `summary` command can be used to calculate and display a variety of univariate summary statistics for the variables in the dataset. To obtain summary statistics only for **score**:

```
> summary(mydata$score)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   19.00   33.00   31.09   45.00   75.00
```

The standard deviation of **score** can also be obtained with the `sd` command

```
> sd(mydata$score)
[1] 17.31437
```

We see that **score** has a mean of 31.09, a standard deviation of 17.31 and can range between a minimum and maximum value of 0 and 75.

Distribution of cohort90

Because **cohort90** contains only six distinct values, we will look at its distribution in a frequency table rather than graphically. The `table` command produces one-way (and two-way) tables of frequency counts. We use this command to tabulate **cohort90** and we store the table as a new object **mytable**

```
> mytable <- table(mydata$cohort90)

> mytable

  -6   -4   -2    0    6    8
6478 6325 5245 4371 4244 7325
```

The number of observations in each category from -6 (year 1984) to 8 (year 1998) are shown. To obtain the row percentages we use the `prop.table` command

```
> prop.table(mytable)

        -6        -4        -2         0         6         8
0.1905967 0.1860951 0.1543192 0.1286042 0.1248676 0.2155172
```

For example, 12.86% of students in our dataset belong to the 1990 cohort (coded zero). The largest proportion of students is from the 1998 cohort, with somewhat smaller proportions from 1990 and 1996.

Finally we can also calculate the cumulative percentages by combining the previous `prop.table` command with the `cumsum` command:

```
> cumsum(prop.table(mytable))
        -6        -4        -2         0         6         8
0.1905967 0.3766918 0.5310109 0.6596152 0.7844828 1.0000000
```


These three pieces of output can easily be combined into a new matrix using the `cbind` command which can combine the columns of vectors, matrices and dataframes. We do this below, calling the new matrix **mytablecomb**

```
> mytablecomb <- cbind(mytable, prop.table(mytable),
cumsum(prop.table(mytable)))

> mytablecomb
   mytable
-6    6478 0.1905967 0.1905967
-4    6325 0.1860951 0.3766918
-2    5245 0.1543192 0.5310109
0     4371 0.1286042 0.6596152
6     4244 0.1248676 0.7844828
8     7325 0.2155172 1.0000000
```

Finally we add to the matrix the column headings “Freq”, “Perc” and “Cum”

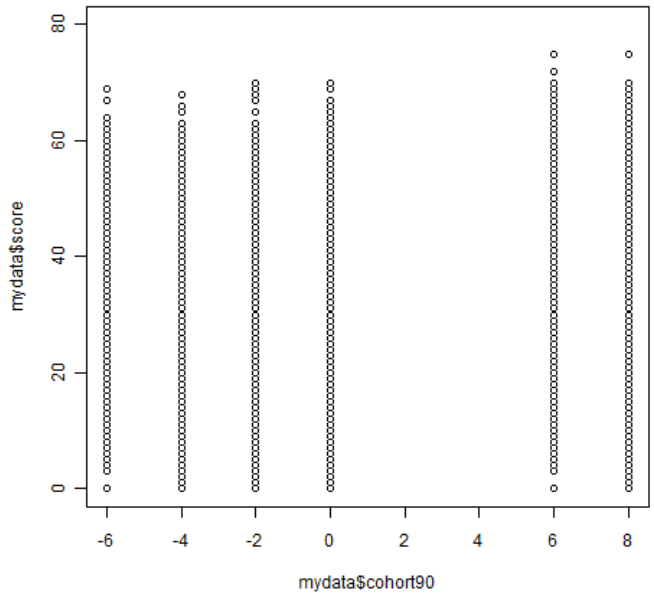
```
> colnames(mytablecomb) <- c("Freq", "Perc", "Cum")

> mytablecomb
   Freq      Perc       Cum
-6 6478 0.1905967 0.1905967
-4 6325 0.1860951 0.3766918
-2 5245 0.1543192 0.5310109
0  4371 0.1286042 0.6596152
6  4244 0.1248676 0.7844828
8  7325 0.2155172 1.0000000
```

Relationship between score and cohort90

Before fitting a linear regression model with attainment and cohort, we use the `plot` command to examine the nature of their relationship using a scatterplot. The `ylim` argument is used to scale the y-axis from zero to 80.

```
> plot(mydata$cohort90, mydata$score, ylim = c(0,80))
```



Although there is some suggestion of a positive linear trend, it is difficult to see the relationship because of the small number of distinct values of **cohort90**. We will therefore supplement the scatterplot with a table of the mean attainment score for each value of **cohort90**.

To tabulate the mean of **score** for each value of **cohort90**, we first need to use a series of commands to construct the table and then to store the table in a new object called **tablescore**.

```
> l <- tapply(mydata$score, factor(mydata$cohort90), length)

> m <- tapply(mydata$score, factor(mydata$cohort90), mean)

> s <- tapply(mydata$score, factor(mydata$cohort90), sd)

> tableScore <- cbind("Freq" = l, "mean(score)" = m, "sd(score)" = s)
```


The `tapply` command applies a function to each group of values given by the levels of the specified factor. Thus, the first use of the `tapply` command calculates the length of **score** for each value of **cohort90**, providing the number of observations for each level of **cohort90**. The second use of the `tapply` command calculates the mean value of **score** separately for each value of **cohort90** while the third use of the `tapply` command calculates the standard deviation of **score** for each cohort.

We can view **tableScore** by simply typing its name into the R console

```
> tableScore
   Freq mean(score) sd(score)
-6 6478    23.65545  18.07995
-4 6325    24.77265  17.37533
-2 5245    28.52450  15.93629
0  4371    29.10043  15.76355
6  4244    39.43473  13.55147
8  7325    41.33065  13.00926
```

We can see that the mean attainment has increased over time. (Note also that the variability in attainment has decreased; we shall return to this in Module 5.) We will fit a linear trend in our regression analyses. However, in P3.2, you will see how to fit a nonlinear trend, following Croxford and Raffe's (2006) approach.

The Pearson correlation coefficient for the linear relationship between **score** and **cohort90** can be obtained with the `cor` command:

```
> cor(mydata$score, mydata$cohort90)
[1] 0.4088625
```

The correlation is 0.409.

P3.1.2 A simple linear regression analysis

Fitting a linear regression model in R

If we assume that the trend in attainment is linear, we can represent the relationship between attainment and cohort by a linear regression model of the form:

$$\textbf{score}_i = \beta_0 + \beta_1 \textbf{cohort90}_i + e_i$$

where $\textbf{score}_i$ is the attainment score for student i and $\textbf{cohort90}_i$ indicates their school year (centred at 1990). The difference between a student's actual score and that predicted for their cohort is the residual e_i.

β_0 and β_1 are the intercept and slope of the population regression line. Because **cohort90** is centred around 1990, β_0 is the attainment score expected for a

This document is only the first few pages of the full version.
To see the complete document please go to learning materials and register:
http://www.cmm.bris.ac.uk/lemma
The course is completely free. We ask for a few details about yourself for our research purposes only. We will not give any details to any other organisation unless it is with your express permission.