1. Introduction and Motivation

Many of the problems associated with teaching econometrics or statistics arise from the acquisition of data that are known to exhibit specific characteristics and are sufficiently different from other examples so as to ensure that students can *individually* demonstrate the skills being assessed. If an instructor's goal is to assess an individual student's work then using an identical data set is open to opportunities of plagiarism, though hopefully this will be a rare event. Clearly with essay style questions this problem has been addressed to some extent by the likes of Turnitin (iParadigms LLC (2006)). Such software is difficult to use in numerical disciplines where good students will be flagged as plagiarising when they get the numerical aspects of their work correct. Identifying plagiarism becomes more difficult in classes of significant size where the sheer numbers of students often necessitate the use of teaching assistants such that there is the possibility of plagiarism across groups which becomes almost impossible to detect.

The purpose of this note is to present a simple method for generating individual data sets for students and the relevant answers based upon an identical structure. Though the underlying data generating process (DGP) is identical for each student, the data itself are not. This ensures that though students can discuss results, they are very unlikely to have any identical outcomes, thus the focus becomes one of interpretation of the results of the analysis rather than using a group approach to individual coursework or project work. There is of course the problem of 'commissioning' the writing of coursework. This will be a perpetual problem with no easy solution as discussed in Hunt (2007).

The context of the modelling task is without doubt of great importance[1]. This is an area where the assessment writer might consider utilising the tools presented here in order to ensure that the analytical skills of the students are tested. In essence this note gives the assessor a skeleton on which to hang the meat of the assessment's background. A natural framework would be an underlying theory, such as the example used here of the Capital Asset Pricing Model (Sharpe (1964)), supplemented by other factors to allow the testing and analysis of other multi-factor models considered in finance (for example Fama and French (1996) or Chen, Roll, and Ross (1986)). This allows the students to demonstrate understanding of modelling and focus on the meaning of the statistical output as suggested by Hubbard (1997) and Petocz and Reid (2007). This further gives the students' empirical work a purpose that is fundamental to the non-specialists' engagement with the topic.

The example given in this paper is a simple multi-factor model for a stock with a named factor with the intent that students should write a report to compare this to the classic Capital Asset Pricing Model. In this scenario it is clear that the student must demonstrate a number of skill sets; statistical, such as model selection and interpretative such as the implications of the model's estimated βs for assets allocation and pricing. The onus becomes one in which the student is able to apply theoretical insights from their readings and other work in order to demonstrate an understanding of the models that they consider. Despite the fact that the data are fictional, the interpretation of their relationships can be practical and linked to theoretical models due to the underlying structure upon which the data are based. This

[1] Thanks to the anonymous referee and editor for commenting on this aspect

allows the student to consider approaches that can be used to model the problem in hand and to deal with the issues that the models bring to light. The data provided using the techniques here could be supplemented by extraneous data to further test the student's abilities to select models and to present the information in the data set. Further the appropriateness of the models can be considered by the students in their analyses. An additional permutation in order to require the students to perform their own analyses is the addition the names of the variables are taken from a sample list. Although this list must be long to ensure that replication of the names is not too common, the data will still differ between the variables of the same name.

2. Method

An approach to solving the underlying problem of identical data sets and answers for (large) groups has been considered by a number of authors (for example Hunt (2007) and Davies and Payne (2001)) and in the case of examination setting by Grün and Zeileis (2009). Generally these approaches involve some form of sampling from a large database to generate data sets for the individual students. Though these approaches have their attractions they are not followed here. Rather using an unique numerical identifier (SID - Student ID) as the seed (to ensure that the random numbers are always identical each time the generator is used), a specific functional form is given with explanatory variables and a random number generator. Each additional explanatory variable is given a new seed based upon the identifier to ensure the appearance of purely random data sets. A natural addition would be to take 'real' numbers, such as stock prices or returns and introduce an innovation in an identical manner to allow students to demonstrate relevant techniques and learning.

Using a set of simple R (2009)[2] functions it is possible to set a specific data generating process and to generate both answers for marking and the data sets for the students. It is further possible to generate spurious data elements to append to the data set in order for students to be able to demonstrate the principles involved in dropping irrelevant variables. At a more advanced level it is possible to use the plethora of R packages to model more complex relationships such as GARCH, ARIMA or the like. The data generating function is applied across the number of students, thus replicating the data generation and analysis for the whole cohort of students. The data are generated with the answers being calculated using a similar function. In the current form the dependent variable is located at the end of the data set rather than the beginning; changes to this would need to be cascaded through the code as appropriate. The DGP can be modified for specific courses or pieces of work. Each variable is given a name based on a number of possibilities. This list can be added to by the assessor to give each student a more varied data set for analysis and to test their ability to link their statistical analysis to the underlying meaning of the data in 'reality'.

```
## Sets up the possible names for the LHS and RHS variables.
```

[2]For readers unfamiliar with the R software project, this is available from http://www.r- project.org/) for Linux, Macintosh and Windows under the General Public License. Precompiled binaries are available for these operating systems. The functions here have been tested with versions 2.8.1 and 2.9.x.

```
## Here only one possible LHS is considered. Further variable names
## can be added to this list increasing the number of possibilities
RHS<-c('RetFish','RetMkt','RetCompetition','FishStockG')
LHS<-c('RetOnStock')


 CWquestions<-function(SID)
     {
 ## Isolate the Student identifier from SID data frame
      SID<-as.numeric(SID[,1])
 ## Set up initial data
     set.seed(SID)
     u<-rnorm(100,0,1)
     x1<-3*u
     SID2<-2*SID
     set.seed(SID2)
     x2<-rnorm(100,0,3)

 ## The number of names may have to be increased as the RHS
 ## increases
     RHSname<-sample(RHS,2,replace=FALSE)
     LHSname<-sample(LHS,1,replace=FALSE)

 ## Data Generating Process
     y<-3+3*x1-0.5*x2+rnorm(100,sd=4)
     dataset<-as.data.frame(cbind(x1,x2,y))
     colnames(dataset)[1]<-RHSname[1]
     colnames(dataset)[2]<-RHSname[2]
     colnames(dataset)[3]<-LHSname[1]
    dataset
  }



CWanswers<-function(SID)
    {
 testset<-CWquestions(SID)
 attach(testset)
## Generate the basic linear regressions
## & names them with the names used in CWquestions

     fit<-lm(as.formula(paste(names(testset)[length(testset)],'~',
     paste(names(testset)[1:(length(testset)-1)],collapse='+'))))
     fit$call<-paste(attr(fit$terms,'variables')[2],'~',
     paste(attr(fit$terms,'variables')[3:length(attr(fit$terms,'variables'))],
     collapse='+'))
```

```
## Returns a list of the answers for extraction using $XXX
     return=list(
      Name=paste(student='Student: ',SID[,2]),
      Summary=summary(fit),
      Fit=fit,
      BG=bgtest(fit),                          ## Breusch Godfrey
      DW=dwtest(fit,alternative=c('two.sided')),                    ## Durbin Watson
      GQ=gqtest(fit,fraction=1/3,alternative=c('two.sided')), ## Goldfeld Quandt
      BPK=bptest(fit,studentize=TRUE),                                ## Koenker
      data=testset
      )

}
```

In addition to the functions `CWquestions` and `CWanswers`, a function `CWdata` is provided. This function creates a CSV file for the student to analyse, each file named and numbered with an abbreviation of their surname and their unique identifier. These may be deposited in an accessible folder for student download or emailed to the student at the relevant time allowing analysis in the preferred package. Security of individual student's data is a potential issue that must be considered; it is important that the students are only able to download the data generated and not able to write this file.

Given the potentially large number of students on a specific course, the output of results for each of the students' data set can also be provided. This may be written to a text file or further linked to a LaTeX output file for more presentable output. As a further level of functionality, the output for each student is kept within R to ensure that, should a student perform an unexpected operation it is possible for the marker to call the relevant information and to generate these innovative analyses. Likewise various plots of the data are trivial to add to the process, but at present are not included due to the proliferation of files that it might generate[3].

The process involved in running this system is relatively straight-forward. The requirements are a data source with an unique identifier for each student, their name and an abbreviation of this and an encoded data generating process. The student information is held as a data frame within R. Requirements from the students, such as Durbin- Watson tests or other such analyses can be added to the output and so long as they are included in the returned list will be easily accessible with minimal additional coding. The data set can also be reported as an output of the function `CWanswers`. If a student performs an unexpected analysis, this can be replicated rapidly as soon as the student's number is identified. The time required is generally minimal once the data generating process is finalised. A group of 50 students' data and results were generated in a matter of seconds using the `lapply` call with the `CWdata` function. It can be seen that the `CWanswers` output is a list of regression outputs and a number of standard

[3]Sending the output of the students' results through Sweave (2002) to a single document might minimise this. A similar results might also be achieved by use of the multipage PDF for graphs and a text `sink` output for the statistical output. Further this would also ease marking as the statistical output can be searched easily for individual student results.

statistical tests. Each element can be called by the usual call from a list: `ans[[XX]]$Fit` where `XX` is the relevant student number. An example of this is listed beneath the `CWdata` function.

```
CWdata<-function(SID)
{
## Individual Filenames
    filename<-paste(SID[,3],SID[,1],'.csv',sep='')
## Data Generation
    data<-CWquestions(SID)
    colnames(data)<-names(data)
## Creates CSV for analysis
    write.csv(data,filename)
}
```

```
## SIDS is the name of the data set that includes the relevant student information.
## This can be read into the system using
## a csv with Student ID, name and abbrev
#   SIDS<-read.csv(file.choose(),header=F)
#   attach(SIDS)
## To create a vector of the length of the number of students.
#   i<-seq(1:dim(SIDS)[1])
## Prints out all the data sets into a set of CSVs
#   lapply(i,function(x){CWdata(SIDS[x,])})
```

3. Example

The necessary code is available in the script `TestingNote.R` that accompanies this note. It requires the package `lmtest` (Zeileis and Hothorn (2002)) in order to have all the relevant test statistics available. The results all come from the underlying data generating process of:

$$\text{Return on Stock}_t = 3 + 3\text{Fish Stock Growth}_t - \frac{1}{2}\text{Return on Market}_t + \epsilon_t \qquad (1)$$

This can be considered as a simple multi-factor model of stock returns with two indices influencing the return on the stock and as such leads to ample opportunity for the students to discuss their results in light of this approach. The modification of the function `CWquestions` will be all that is required to change the underlying DGP and therefore allow more complex or different style questions. An example of the output is given below with the variables. The Student Identifier is set at 1 for replication purposes and the student is Joe Smith. This information is held in the data frame `SID`. Using calls such as `ans$BPK` would give the studentised Breusch Pagan test as specified in the listing associated with the output. Further the actual fitted model is also returned so that, in the case of an imaginative student, updating of the fit is possible to easily check for unexpected simulations.

```
> ans <- CWanswers(SID)
```

```
> print(ans$Summary)
```

```
Call:
"RetOnStock ~ FishStockG+RetMkt"

Residuals:
    Min      1Q  Median      3Q     Max
-8.3053 -2.5461  0.2634  2.5785 11.9899

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.2911     0.4396   5.211 1.06e-06 ***
FishStockG    2.9696     0.1647  18.028  < 2e-16 ***
RetMkt       -0.5787     0.1275  -4.538 1.63e-05 ***
---
Signif. codes:  0 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô Õ 1

Residual standard error: 4.364 on 97 degrees of freedom
Multiple R-squared: 0.7965,        Adjusted R-squared: 0.7923
F-statistic: 189.9 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
> print(ans$BPK)
```

```
        studentized Breusch-Pagan test

data:  fit
BP = 1.8761, df = 2, p-value = 0.3914
```

The tidied results from the linear regression for the exemplar student are given in the Table 1[4]. This also uses the student name as the table caption to aid in the marking and navigation of the answers. With further beautification it would be possible to generate a set of ideal answers for the students and if so required, plots similar to those in Figure 1 can be included.

```
> xtable(ans$Summary, caption = ans$Name, label = "T1")
```

4. Conclusions

This note has presented a straightforward method for providing students with 'pet' data sets of known behaviour that are dissimilar enough to ensure that the student will be required to

[4] xtable Dahl (2009) is used for Table 1

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.2911	0.4396	5.21	0.0000
FishStockG	2.9696	0.1647	18.03	0.0000
RetMkt	-0.5787	0.1275	-4.54	0.0000

Table 1: Student: Joe Smith

perform their own analyses. It will not be a panacaea, in that it is still possible for students to collaborate when the statistic is of the same order, however even this will require the student to understand the impact of a specific test statistic and its implications and that the statistic is indeed of the same order. Though one can never predict what analyses the students might come up with, it should be possible to acquire the answers to a sufficient number of possibilities such that the results of the expected output are generated and thus minimising the calculation of any individual results on a systematic basis. Indeed any systematic innovative analysis might be seen as signs of an extraordinary (in all ways) student.

5. Acknowledgements

The author would like to thank the Editor and anonymous referees for their comments and suggestions along with Neville Hunt whose comments were invaluable. Any errors or omissions remain the author's.

References

Chen N, Roll R, Ross SA (1986). "Economic Forces and The Stock Market." *Journal of Business*, **59**(3), 383–403.

Dahl DB (2009). *xtable: Export tables to LaTeX or HTML*. R package version 1.5-5, URL http://CRAN.R-project.org/package=xtable.

Davies N, Payne B (2001). "Web-created real data worksheets." *MSOR Connections*, **4**(1), 15–17.

Fama E, French KR (1996). "Multifactor explanations of asset pricing anomalies." *Journal of Finance*, **51**, 55–84.

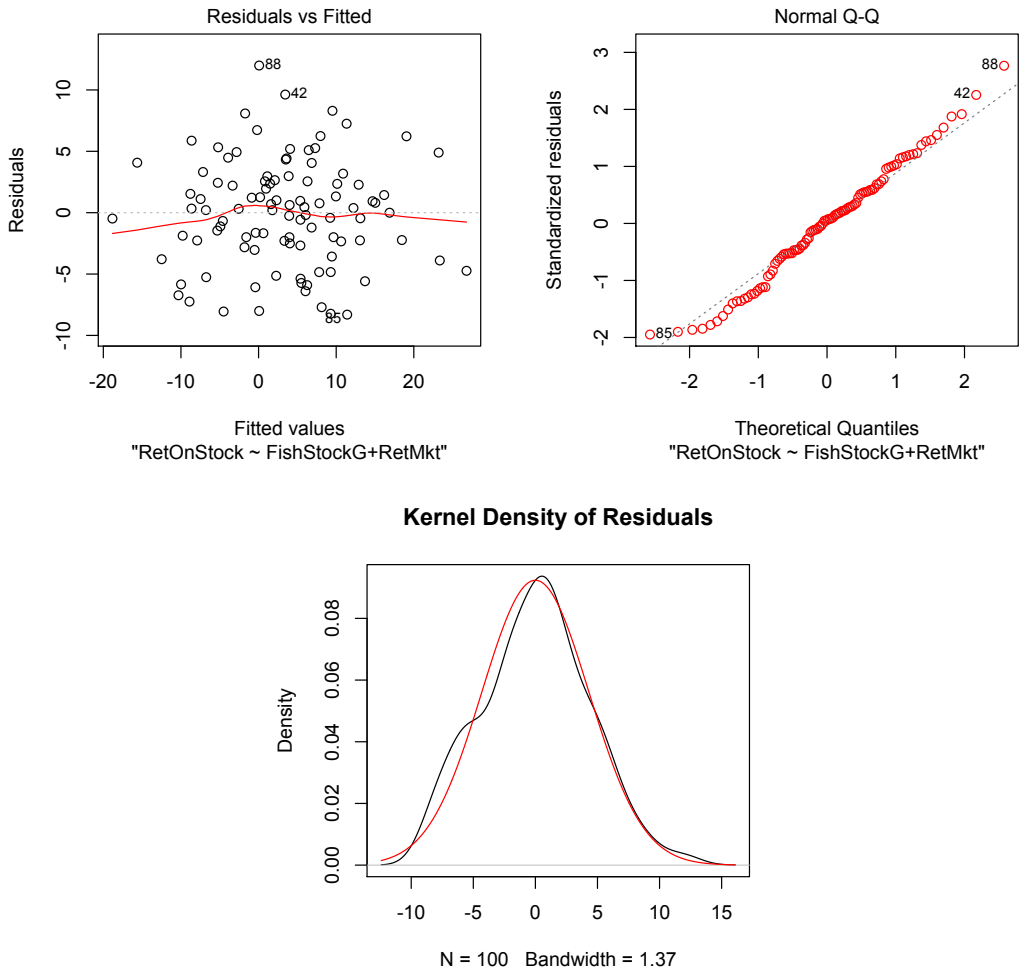
Grün B, Zeileis A (2009). "Automatic Generation of Exams in R." *Journal of Statistical Software*, **29**(10), 1–14. URL http://www.jstatsoft.org/v29/i10/.

Hubbard R (1997). "Assessment & The Process of Learning Statistics." *Journal of Statistics Education*, **5**(1).

Hunt N (2007). "Individualised Statistics Coursework Using Spreadsheets." *Teaching Statistics*, **29**(2), 38–43.

iParadigms LLC (2006). "Turnitin." URL www.turnitin.com.

Figure 1: A Sample of Plots from Student Data



Leisch F (2002). "Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis." In W Härdle, B Rönz (eds.), "Compstat 2002 — Proceedings in Computational Statistics," pp. 575–580. Physica Verlag, Heidelberg. ISBN 3-7908-1517-9, URL http://www.stat.uni-muenchen.de/~leisch/Sweave.

Petocz P, Reid A (2007). "Learning & Assessment in Statistics." *IASE/ ISI Satellite Conference*.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Sharpe WF (1964). "Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk." *Journal of Finance*, **19**(3), 425–442.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." *R News*, **2**(3), 7–10. URL http://CRAN.R-project.org/doc/Rnews/.