Coding Practice in Economics: A Survey and Recommendations

Joe Hirschberg and Jenny Lye

Department of Economics

University of Melbourne

1. Introduction

It is now common procedure for academic journals to require the submission of all the code and data used in the generation of the results published in the journal to ensure the replicability of the conclusions presented. It has become a requirement that data and computer programs be submitted as part of a submission of a research essay or dissertation.

This has become especially important with the advent of applied research that employ extensive estimation procedures applied to large scale datasets.

Although the requirements for the description of data are often quite extensive there is little direction for the nature of the computer code submitted other than the ability to perform the reported analysis.

In this paper we report on an attempt to categorise the software and the code submitted to a highly ranked journal of economics and to make recommendations as to how improve the coding style of these submissions.

2. A survey of coding practice in the *AER* and *AEA P&P* for 2020

Survey of software used

Here, we replicate the approach taken by research conducted by Jon Fiva, Tuva Værøy and Federico Herrera (FVH) the results of which has been reported in tweets posted in 2019 and 2021. In our sample we examine the scripts we are able to download from papers that included them in the replication resources for 171 papers published in the *AER* and *AEA P&P* for 2020.

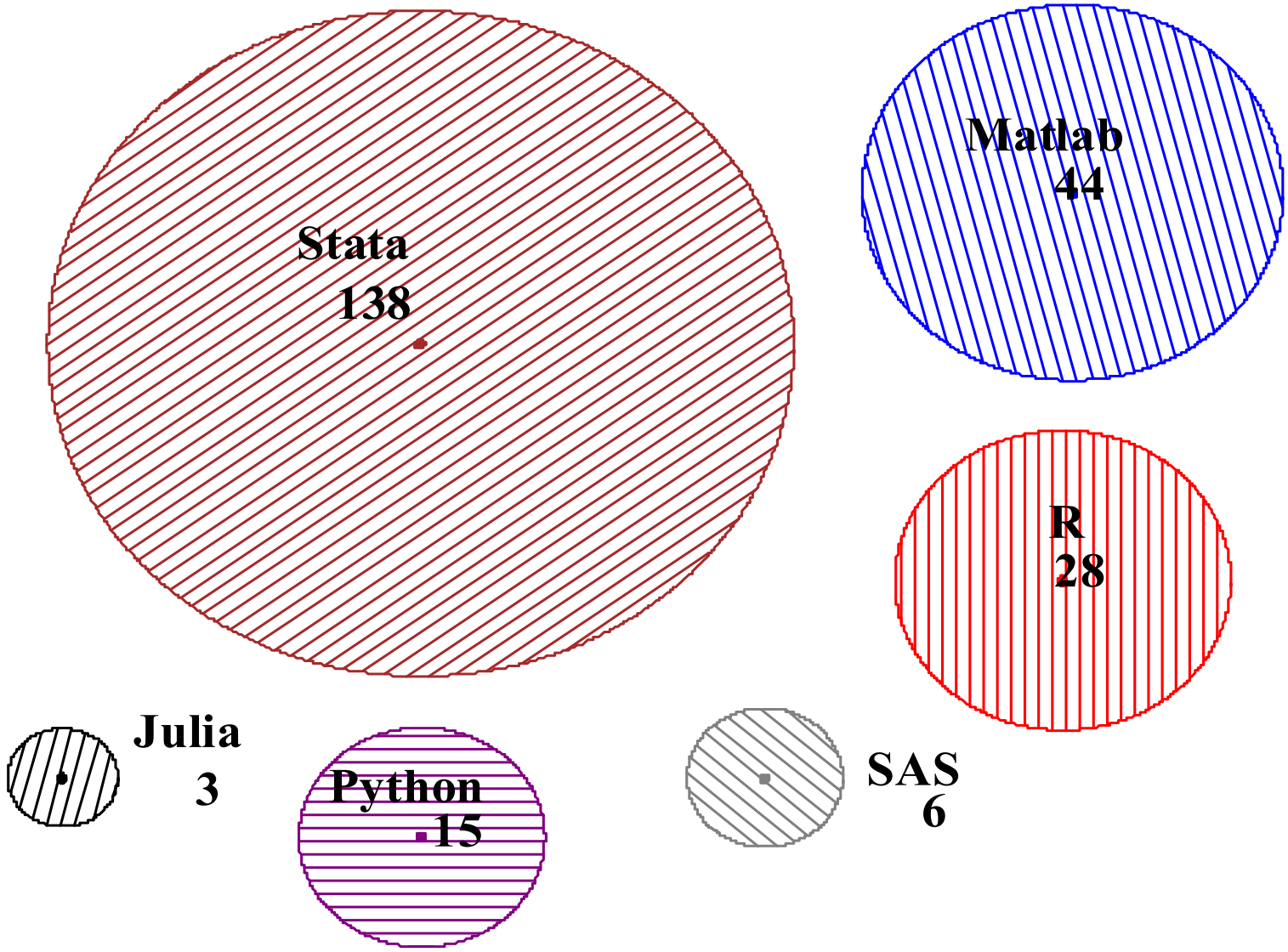
The number of all programs used by 171 papers in 2020 issues.

Issue	Number	Mean	Median	Max	Min
AEA P&P	81	7.25	3.0	57	1
AER	90	54.82	23.5	270	1

The distribution of software in the 3,908 scripts in the 171 papers.

Software	%
Julia	1.05
Matlab	34.01
Python	3.20
R	10.02
SAS	2.28
Stata	49.44
Total	100.00

Software use by the 171 papers



Software combinations

Of the 171 papers 113 exclusively use only one type of software while 58 employ a combination of different software scripts. In all, we find 234 paper and software combinations. We find that 64% of papers that use *Stata* – only use *Stata*.

The software employed for 234 paper/software combinations.

Software	Number of papers (%)		
	Use	Also use other	Exclusive use
Julia	3	2 (67%)	1 (33%)
Matlab	44	31 (70%)	13 (30%)
Python	15	14 (87%)	2 (13%
R	28	19 (68%)	9 (32%)
SAS	6	6 (100%)	0 (0%)
Non-Stata	96	72 (75%)	25 (25%)
Stata	138	47 (34%)	88 (64%)
Total	234	119 (52%)	113 (48%)

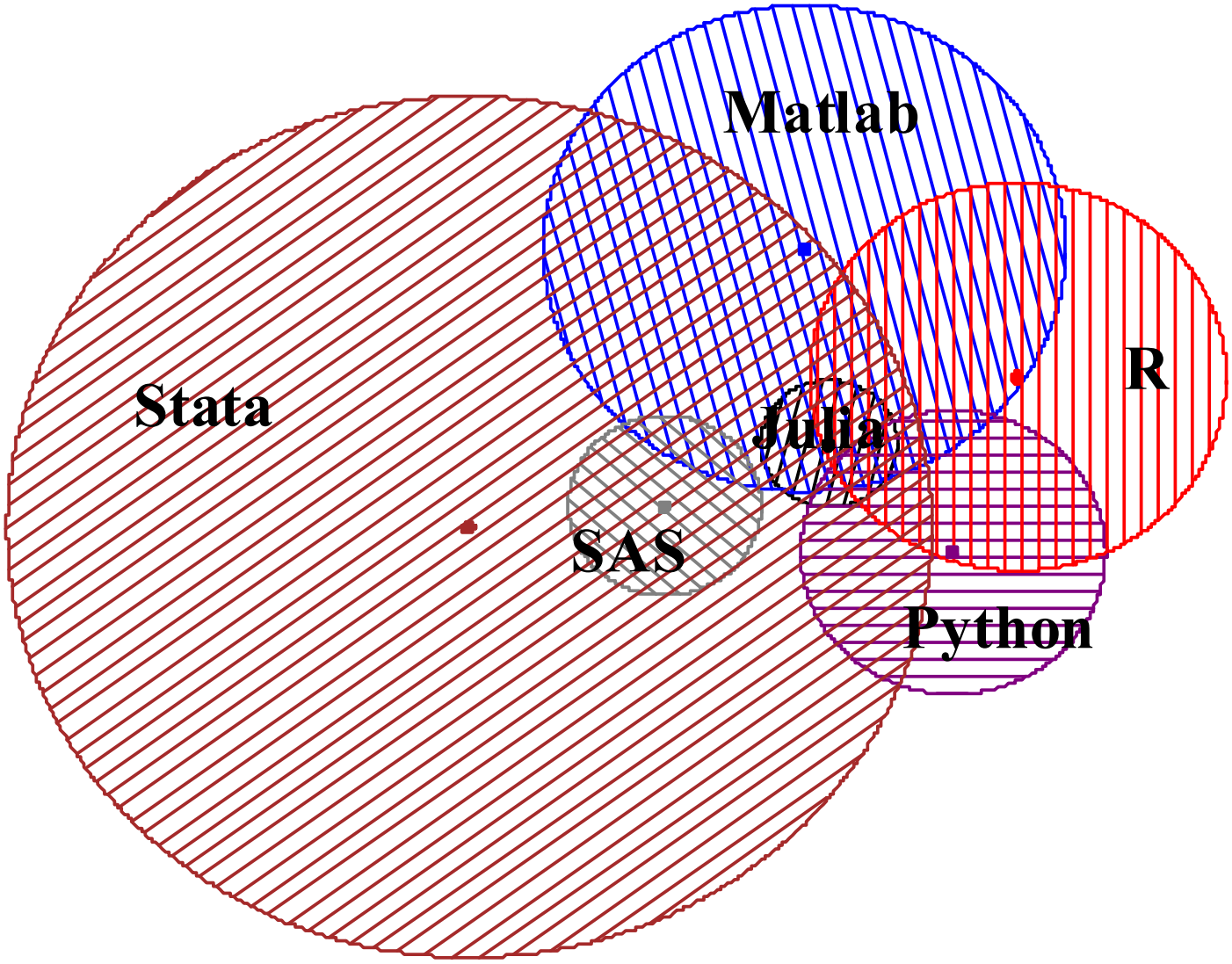
We find that 32% of papers used more than one additional software routine and 8% used two or more. That brings up the question as to: What combinations of different software are most common?

Table 3. Distribution of multiple software used by 171 papers with code.

Software	***Software***					
	Julia	*Matlab*	*Python*	*R*	*SAS*	*Stata*
Julia	**3**	1	1	1	0	1
Matlab	1	**44**	3	6	2	27
Python	1	3	**15**	6	0	11
R	1	6	6	**28**	1	16
SAS	0	2	0	1	**6**	6
Stata	1	27	11	16	6	**138**

Note that in addition to these software packages we found 75 papers that included files in *Microsoft Excel* format in their replication folders as well.

Software use combinations by the 171 papers



3. A survey of *Stata* coding practice.

We examined the *Stata* code submitted for 135 papers that appeared in the 2020 *American Economic Review* and the *Papers and Proceedings* of the 2020 annual meeting of the *American Economic Association*. This constituted 1,785 scripts that amounted to 528,191 lines of code. The table below summarises the number of lines of code in this sample of scripts.

Table 5 The number of lines of *Stata* code over the 135 papers that uses *Stata* in either the *AER* or the *AEA P&P*.

Issue	Number	Mean	Median	Max	Min
AEA P&P	65	846.72	378	6,650	6
AER	70	6,801.99	4,419	42,661	84

Table 6 The program elements that appear at least once in each of the 1,785 scripts.

Program Feature	% of scripts
A Blank line	98.38
Equation is used	97.82
A Comment	91.09
An Equation that is spaced	82.80
Use of *,= , # for separator	75.52
Tab in a line	70.70
Start a line with either a tab or at least 2 spaces	68.68
There is a loop defined	49.92
Use of a continuation "///" or a delimiter	35.80
Definition of a variable label	22.86
At least one Line > 320 characters	13.95
Install special routine via "ssc install"	7.34
Make an assertion with "assert"	5.94
There is "global" or a "local" defined	1.74
Set the version of *Stata*	1.01

Means of average values per script

Characteristic	Scripts	Mean	St D	Min	Max
The line length	1785	35.07	15.11	2.00	268
Number of lines in program	1785	333.49	1303.98	1.00	42946
Number of words used	1669	92.66	949.17	1.00	36382
Number of words per line	1785	5.09	3.78	0.67	47.07
Word size	1669	10.55	3.20	6	31
# spaces to start	1785	0.20	0.74	0.00	13.85
% of equations w space	1746	48.47	34.67	0.00	100.00

Econometric practice

In current econometric practice there is widespread use of regressions with clustered standard error estimates. Note this includes all *Stata* scripts although not all scripts are used for estimation. Some are only designed to prepare data for analysis.

Proportion of all *Stata* programs with at least one

Use of a regression	64.50%
Use of a regression with clustered SE	15.92%
Use of clustered SE in any procedure	24.42%

4. Recommendations for Code

Why a style guide for code?

- A well written program is easier to debug and maintain and is more useful to others who may want to replicate your work, extend it, to speed it up, or borrow from it. Good style helps the reader to concentrate on parts of code elements that can be checked separately.
- The code for research papers is often used for pedagogical purposes. The availability of well written code has become a useful tool in the training of future researchers.
- The rise of "Research Teams" in economics. Jones (2021) documents the rise of multiple author contributions in economics over solo author papers. When multiple authors work on a project it is often necessary for authors to share code.[1]

---

[1] Jones, B. (2021), "The Rise of Research Teams: Benefits and Costs in Economics", *Journal of Economic Perspectives*, 35, 191-216.

Some of the suggestions for preparing code we recommend the following:

- Do not sacrifice clarity in your program over speed or memory.
- Adopt a consistent style for your program.
- Use margins – don't start all code in the first column.
- Dense code is very hard to read – use banks to space out equations.
- Add liberal comments – you're not going to remember the details.
- Only write a new program when it is necessary.
- Begin by getting a version of your program working first.
- If possible, always test a smaller version of your program first.
- Write a program that only does what it needs to do.
- Separate tasks should be put into different programs.
- Adopt the "*Don't Repeat Yourself*" or DRY principle.
- Keep different versions of your program.
- Use of combinations of different software when needed.

5 A categorization of *Stata* code

To demonstrate that one could categorise the quality of code a hierarchical cluster analysis was performed based on several measures taken from each script.

- the average number of comments per line,
- the average the use of spaces in equations,
- the proportions of the characters each line that is blanks,
- the proportions of lines that have tabs, and
- the proportion of the code that has blank lines.

To provide a comparison we constructed two example codes that epitomize the most readable and the most obtuse code to perform the same function in *Stata*. To make this a comparative sample, we perform this comparison to scripts in papers with 3 or fewer scripts.

Two archetypic *Stata* codes that perform a typical regression.

Script B.1. Readable style code (1,213 characters)

```
/*
        name.do
        Do-file to run a regression and to test the hypothesis that
        the price parameters sum to zero.
*/
/*
        Make sure that there are no data sets in the memory
*/
        clear all
/*
        Read the comma delimitated data set beer.txt and label the variables
*/
        insheet using https://www.XXX/beer.txt

                label data "Beer demand and prices in logs"
                label variable q "Log of Beer consumed"
                label variable m "Log of income"
                label variable pb "Log of price of beer"
                label variable pl "Log of price of liquor"
                label variable pr "Log of price of other goods"
/*
        Make a scatter plot matrix of the data to check if there are outliers or other
        potential problems.
*/
        graph matrix q pb pl pr m
/*
        Run a regression on the log quantity
*/
        regress q pb pl pr m
/*
        Test if the demand equation is homogeneous of degree 1
*/
        test pb + pl + pr + m = 0
/*
        We can alternatively test the hypothesis by transforming the data so
        that a parameter is defined as the sum of the parameters - then the t-test
        if it is equal to zero should be equivalent to the test performed above.
*/
        gen z1 = pb
        gen z2 = pl - pb
        gen z3 = pr - pb
        gen z4 = m - pb
/*
        The coefficient for z1 will be the test statistic for the sum of the parameters
*/
        regress q z1 z2 z3 z4
```

Script B.2. Obtuse style code (165 characters)

```
#delimit;
clear all;insheet using "https://www.XXX/beer.txt";
gr mat q pb pl pr m;reg q pb pl pr m;te pb+pl+pr+m=0;
g z1=pb;g z2=pl-pb;g z3=pr-pb;g z4=m-pb;reg q z1-z4;
```

Dendrogram for averages from the AEA P&P paper's code and with up to 3 example routines

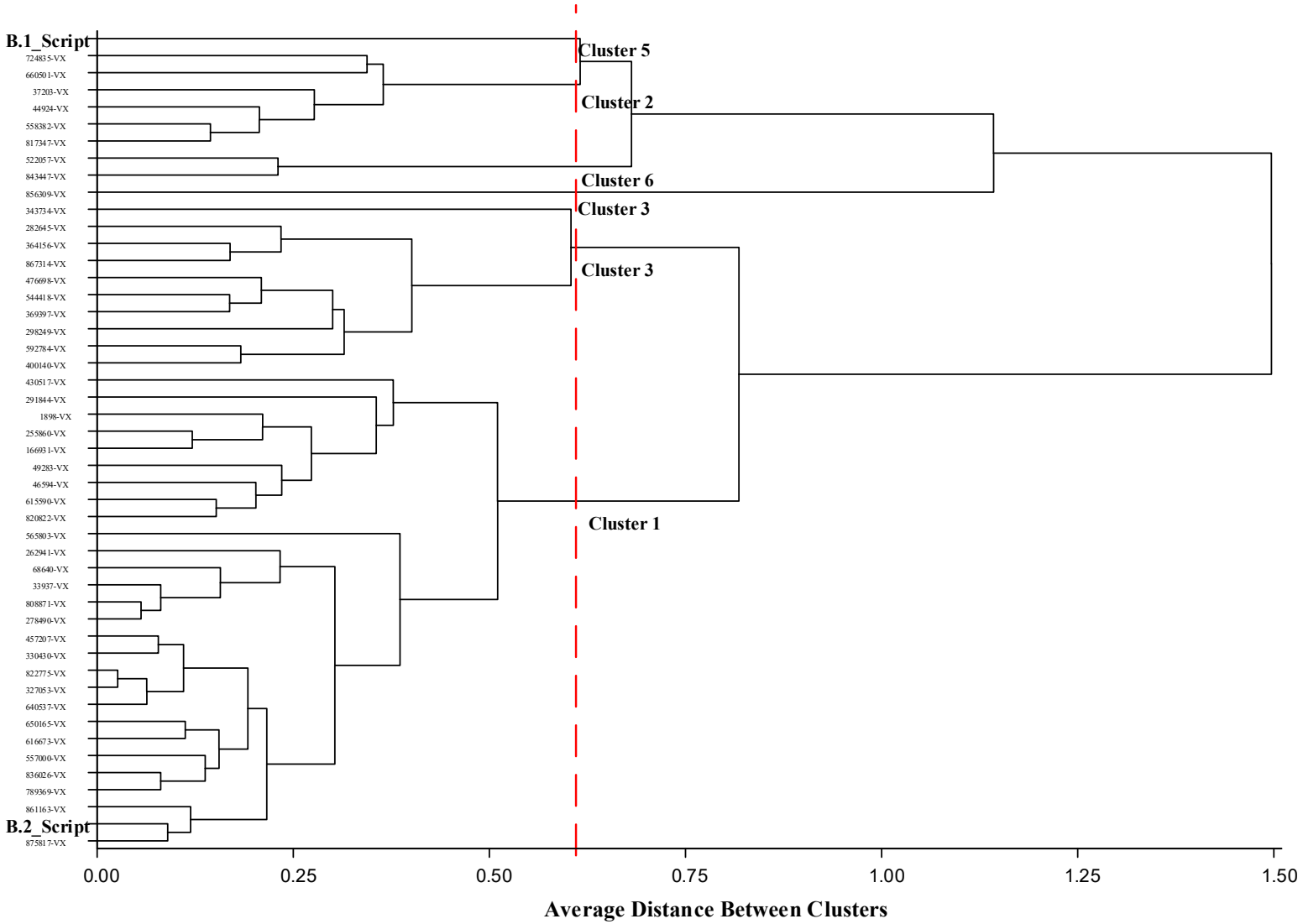
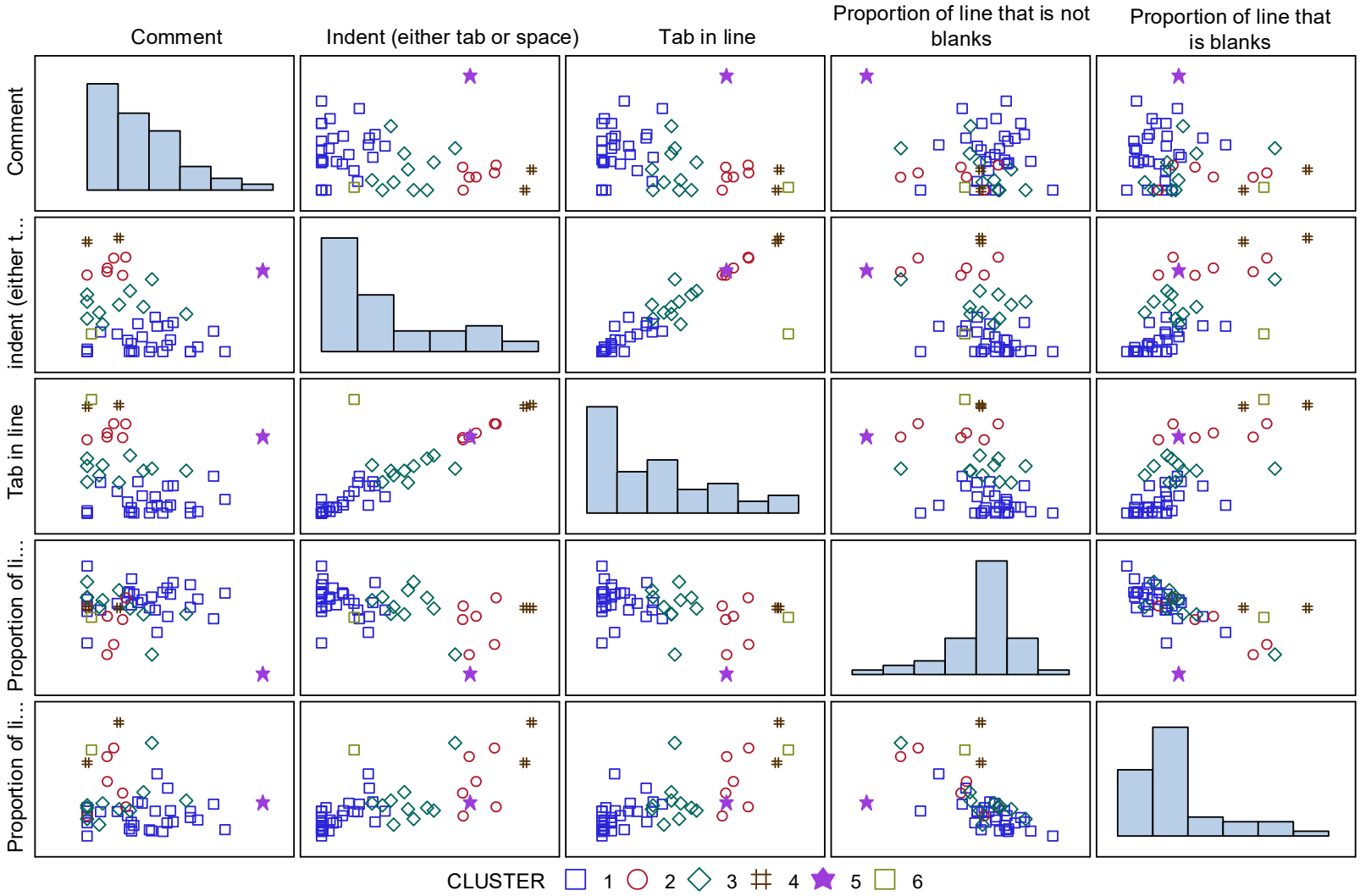


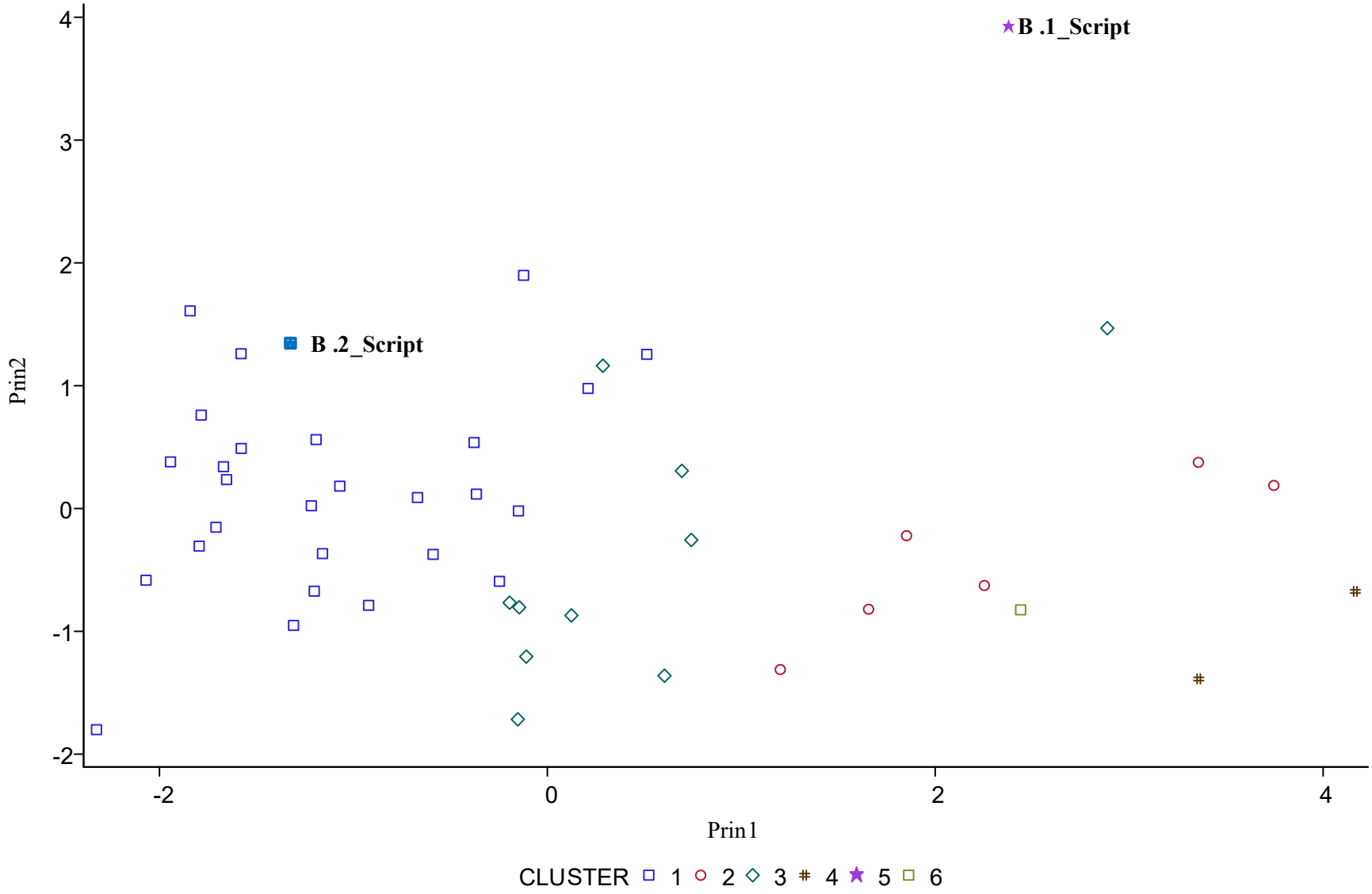
Figure 5 The scatter plot matrix of the values of the characteristics used in the cluster analysis with the allocation of cluster.



The correlation matrix of the characteristics.

		1	*2*	*3*	*4*	*5*
1	***Comment***	1.00	-0.17	-0.21	-0.16	-0.14
2	***Indent (either tab or space)***	-0.17	1.00	0.89	-0.45	0.68
3	***Tab in line***	-0.21	0.89	1.00	-0.41	0.73
4	***Proportion of line that is not blanks***	-0.16	-0.45	-0.41	1.00	-0.60
5	***Proportion of line that is spaces***	-0.14	0.68	0.73	-0.60	1.00

The scatter plot of the first two principal components.



Discussion

We survey the coding practice of research published in the 2020 volume of the *American Economic Review* and the *Papers and Proceedings of the American Economic Association* that have included software code files for replication.

We have determined the frequency of the use of different software packages and the combinations of software of different software used.

From this survey we have found that *Stata* is most widely used software for this research.

We have then read all *Stata* program scripts in the sample of papers, to determine the degree to which these programs use specific characteristics of program style.

Suggestions are then proposed as a potential code style guide.

We then demonstrate the possibility of a method for the categorisation of code. This is done by comparing a sample of these scripts to two model *Stata* routines we then demonstrate how one can use a computer algorithm to distinguish between coding styles.

More

What is the response to these proposals by the practitioners?

This method could lead to the development of an automated method for the evaluation of code style that could be used by journal editors and repositories of software code.

Is there a relationship between code style and the use of the code? Could this lead to the development of a new measure of publication impact in a similar manner to the existing citation statistics?

References

Broman, K. and K. Woo (2018), "Data Organization in Spreadsheets", *The American Statistician*, 72, 2-10.

Cox, N. (2005), "Suggestions on *Stata* programming style", *The Stata Journal*, 5, 560-566.

Cox, N. (2020), "Speaking *Stata*: Loops, again and again", *The Stata Journal*, 20, 999-1015.

Fiva, J., and T. Værøy (2019, Oct 11), "For applied economics @*Stata* is still the only game in town", https://twitter.com/JFiva/status/1182293282195460097?s=20

Fiva, J., T. Værøy and F. Herrera (2021, Aug 9), "For applied economics @*Stata* is still the only game in town", https://twitter.com/JFiva/status/1424679980538241025?s=20.

Gentzkow, M. and J. Shapiro,(2014), "Code and Data for the Social Sciences: A Practitioner's Guide", https://web.stanford.edu/~gentzkow/research/CodeAndData.pdf, downloaded 11/1/2021.

Hirschberg, J. and J. Lye, (2020) "Grading Journals in Economics: The ABCs Of The ABDC, *The Journal of Economic Surveys,* 34, 876-921.

Hirschberg, J, E. Maasoumi, and D. Slottje, (1991)"Cluster Analysis and the Quality of Life Across Countries", *Journal of Econometrics*, 50, 131-150.

_____, (2001), "Cluster of Attributes and Well-Being in the USA", *Journal of Applied Econometrics,* 16, 445-460.

Hunt, A. and D. Thomas (1999), *The Pragmatic Programmer: From Journeyman to Master*, USA: Addison-Wesley Professional.

Jones, B. (2021), "The Rise of Research Teams: Benefits and Costs in Economics", *Journal of Economic Perspectives*, 35, 191-216.

Kernighan, B. and P. Plauger,(1978), *The Elements of Programming Style*, New York: McGraw–Hill.

Nagler, J. (1995), "Coding style and good computing practices", *Political Science and Politics*, 28, 488–92.

Orozco, V., C. Bontemps, E. Maigné, V. Piguet, A. Hofstetter, A. Lacroix, F. Levert, J. Rousselle (2020), "How to make a Pie: Reproducible Research for Empirical Economics and Econometrics", *Journal of Economic Surveys*, 34, 1134-1169.

Press, W., S. Teukolsky, W. Vetterling, B. Flannery, and M. Metcalf, (2007), *Numerical Recipes: The Art of Scientific Computing,* 3rd ed, Cambridge University Press.

Sokal, R. and Michener, C. (1958), "A Statistical Method for Evaluating Systematic Relationships", *University of Kansas Science Bulletin,* 38,1409–1438.

Vilhuber, L., Turrito, J., & Welch, K. (2020), "Report by the AEA data editor", *AEA Papers and Proceedings*, 110, 764–75. https://doi.org/10.1257/pandp.110.764

Vilhuber, L. (2020). "Reproducibility and replicability in economics", *Harvard Data Science Review*, 2(4).https://doi.org/10.1162/99608f92.4f6b9e67

_____ (2021), "AEA Data and Code Availability Policy", *AEA Papers and Proceedings,* 111, 818-823. https://www.aeaweb.org/articles?id=10.1257/pandp.111.818

Wassberg, J. (2020), Computer Programming for Absolute Beginners: Learn essential computer science concepts and coding techniques to kick-start your programming c1212121212000areer, Packet Publishing.