Abstract: The paper describes PCDM and PCDM4MP as new tools and commands capable of exploring large datasets. They select variables based on identifying the absolute values of Pearson’s pairwise correlation coefficients between a chosen response variable and any other existing in the dataset. In addition, for each pair, they also report the corresponding significance and the number of non-null intersecting observations, and all this reporting is performed in a record-oriented manner (both source and output). Optionally, using threshold values for these three as parameters of PCDM, any user can select the most correlated variables based on high magnitude, significance, and support criteria. The syntax is simple, and the tools show the exploration progress in real-time. In addition, PCDM4MP can trigger different instances of Stata, each using a distinct class of variables belonging to the same dataset and resulting after simple name filtering (first letter). Moreover, this multi-processing (MP) version overcomes the parallelization limitations of the existing parallel module, and this is accomplished by using vertical instead of horizontal partitions of large flat datasets, dynamic generation of the task pattern, tasks, and logs, all within a single execution of this second command, and the existing qsub module to automatically and continuously allocate the tasks to logical processors and thereby emulating with fewer resources a cluster environment. In addition, any user can perform further selections based on the results printed in the console. The paper contains examples of using these tools for large datasets such as the one belonging to the World Values Survey and based on a simple variable naming practice. This article includes many recorded simulations and presents performance results. They depend on different resources and hardware configurations used, including cloud vs. on-premises, large vs. small amounts of RAM and processing cores, and in-memory vs. traditional storage.

Keywords: data mining; pairwise correlation; support for filtering and parallel processing; regression analysis; large data sets; real-world example; triangulation; cross-validation; full support for replication of results

MSC: 33-04

1. Introduction

Recently, many concerns regarding the replicability of scientific findings as results of experiments and data analysis reported in various publications emerged. There are many cases in which other researchers have to re-implement and adapt to validate the findings or even replicate the data analysis or the computation using the same data, procedure, methodology, and even code or script sequences [1,2].

Nowadays, many statistical tools (SPSS, R, Matlab, Minitab, SAS, Stata, etc.) encourage replicability through consistent support for data analysis, statistical calculations, visualizations, advanced tests, and automatic reporting of results and aid for community contributions and versioning. The latter concerns both the main software version for which a certain command was written (https://www.stata.com/features/overview/
tegrated-version-control/, accessed on 1 June 2022) and the release marker telling the program’s version in the proprietary tracking scheme (https://www.stata.com/support/faqs/programming/release-marker-versus-version-number/, accessed on 1 June 2022). Stata (https://www.stata.com, accessed on 1 June 2022) benefits from all of these [3–7] and it successfully combines a friendly user interface with support for power users and programmers. There are many new Stata programs and commands introduced to serve different purposes. Among them, those used for data mining (as a crucial component of business intelligence [8,9] or even dedicated Cross-Industry Standard Process model—CRISP-DM [10]) and variable selection such as: stepwise [11] with forward and backward selection, or the LASSO package [12]. The latter has different components. For instance, CVLASSO can perform cross-validations on random subsamples. More, RLASSO places a high priority on controlling overfitting [13,14]. In addition, the calculation of shrinkage statistics to measure overfitting using overfit.ado [15]. Moreover, PCA stands for Principal Components Analysis [16] and allows the estimation of parameters for principal-component models. Even more, it is worth mentioning here the Bayesian Model Averaging (BMA) and weighted-average least-squares (wals) for estimating linear regression models with uncertainty about the choice of the explanatory variables [17]. The Boosting technique for decision tree classifiers [18] also has a well-defined place in the list of exploratory ones. Still, the boost plugin in Stata is too time-consuming in terms of execution, and it has limited capabilities such as automatic variable selection and treatment of missing values [19].

In terms of parallel approaches, we mention early contributions focused on computing the “information gain” using MapReduce jobs executed on Hadoop Clusters [21,22] or the open-source distributed machine learning library, namely MLlib [23] and other more recent methods and techniques in Apache Spark [24,25] and Mahout [26–28]. In addition, it is worth referring to other new approaches that focus in particular on computing Pearson’s correlation coefficients, such as ForkJoinPcc, which uses the parallel MATLAB APIs [29] to mimic the well-known parallel programming model, namely the fork-join model.

In this paper, we describe new exploratory tools, namely PCDM and PCDM4MP. They serve data-mining and variable selection purposes being also two new dedicated commands for Stata. They rely on pairwise correlation computation and printing easy-to-copy and filterable results in the console. Their design enables them to support the rapid selection of most correlated variables with the one specified right after the command name (the target), and this is even without knowing and stating the name of the rest of the variables in any dataset. The latter is an advantage that makes them reliable data-mining tools. For PCDM, we also considered a direct but more complex filtering scenario. The latter considers a set of three most important values as parameters. The first corresponds to certain thresholds for the correlation coefficients (minimum accepted absolute value). The second describes the minimum accepted number of valid observations at the intersection of every single pair of two variables meaning the target one and each of the remaining. The third is afferent to the maximum accepted p-value [30,31]. For PCDM4MP, the focus was on speed via multi-processing.

2. Materials and Methods

Both a persistent online Google Drive folder (https://drive.google.com/drive/folders/1kC2lwD3v9sD9kePHMDq8xJr4xu_VdK, accessed on 1 June 2022) and a GitHub repository (https://github.com/danhomocianu/PCDM-PCDM4MP, accessed on 1 June 2022) served to keep both commands (.ado files), other processing and selection script sequences (mostly. do files in the “additional scripts” section), and many demos (the “recorded simulations” section) acting as short tutorials [32,33] supporting this research.

A dataset from the World Values Survey (WVS) (The Data Availability Statement at the end of this manuscript and the video instructions in the 1st recorded simulation, namely,
A dataset from the World Values Survey (WVS) (The Data Availability Statement at https://drive.google.com/u/0/uc?id=1wiwHo1gYrmccZYojB4y1kjgcVQdVfZwE&export=download, accessed on 1 June 2022) proves the usefulness of PCDM in real-world scenarios with large amounts of historical data [34–36]. The WVS is one of the biggest cross-national, non-commercial, time-series empirical surveys of human beliefs and values ever conducted. It is also a representative comparative social investigation conducted globally in over 100 countries and includes seven waves applied once every five years (from 1981 to 2020). WVS served many kinds of research and studies [37–43]. The starting point was the entire set of variables (1045) and observations (426,452) in this dataset which was also loaded and exported as .csv using Stata (line 8—Figure 1). In addition, a simple binary derivation of the variable to analyze (C033, Job satisfaction) considering the symmetric split of the original scale. C033 (original scale of 1 = Dissatisfied up to 10 = Satisfied) was the starting point to generate C033_bin. This binary form has the value of 1 for all non null initial values greater than or equal to 6 and 0 otherwise (but still not null original values—lines 3–5, Figure 1—preProcessingScript.do (https://drive.google.com/u/0/uc?id=1sQntMANwM3DzP5CAI2u3lo-xD6f5_4xW&export=download, accessed on 1 June 2022).

The first thing to do was to intersect the results obtained using both variable selections corresponding to those two forms of the outcome and the method based on PCDM and PCDM4MP in Stata (versions 16.0 and 17.0, MultiProcessing, x64, StataCorp, College Station, TX, USA). It meant computing and filtering on absolute values of pairwise correlation coefficients, their significance, and the corresponding number of observations. Additional filters served the latter after copying the results from the console in a spreadsheet tool. The alternative, also considered, was to demonstrate the use of optional arguments. Further selections relied on the LASSO pack and BMA (in both Stata versions above).

In addition, we tried to find the most resilient predictors by using another method based on the Adaptive Boosting technique and to show which of them are among those obtained using the approach based on PCDM and PCDM4MP. Therefore, we first loaded the .csv dataset in the Rattle (https://rattle.togaware.com, accessed on 1 June 2022) (version 5.4.0) interface from R. Then, we used this technique for decision tree classifiers as an alternative data-mining round, considering the following default settings: Trees——50, Max Depth—6, Min Split—20, Complexity—0.01, Learning Rate—0.3, Threads—2, Iterations—50, Objective—binary logistic/logit). It benefited from the support of a virtual machine available in a private cloud described below.

Other correlation commands (e.g., correlate) further served each tested regression model (logit and OLS-Ordinary Least Squares). This time, they are critical concerning just the resulting and intersecting predictors as maximum absolute values from their matrices with correlation coefficients (maxAbsVFMCC). In addition, the highest ones for the computed Variance Inflation Factor or OLSmaxComputationVIF were subject to assessment against (no more than) the maximum acceptable ones (Equation (1)) or OLSmaxAcceptVIF [44–46] for each OLS regression model. The measurements also concerned accuracy, as AUCROC.
(better for larger values). The latter means the Area Under the Curve of Receiving Operators Characteristics [47,48]. The same for the information gain and model fitness via AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) values [49–51], meaning more information gain and a better fit for lower such values.

\[ \text{OLSmaxAcceptVIF} = \frac{1}{(1 - \text{the R-squared value of the model})} \]  

PCDM (https://drive.google.com/u/0/uc?id=1hRBn0tv5wSXFjUbzVumIfqvRasOCCWGcY&export=download, accessed on 1 June 2022) is installable (download and copy to one of the ado directories (https://www.stata.com/manuals13/u17.pdf, accessed on 1 June 2022, Section 17.5.2 of the previous online .pdf manual)—e.g., C:\ado\personal). The source script and syntax of PCDM (Listing A1, Appendix A) are easy to understand and allow two main types of use.

![Figure 2. Simple usage scenario involving a single logical processing core (PCDM) with the real-time reporting of execution progress for a 1045 variables dataset (WVS). Notes: The asterisk (*) stands for all variables in the dataset. The first dot (.) is automatically generated by Stata after entering the command (pcdm C033 *). The subsequent occurrences of dots (PCDM’s feedback in Stata’s console) followed by numerical values indicate zeros (0) followed by their decimal parts (e.g., .065 is actually 0.065 while -.1175 is actually −0.1175). The “e” followed by the minus (“−”) and numbers indicate the E notation corresponding to the scientific one (1.4e-16 is actually 1.4 × 10^{-16}).](image)

The first is simple meaning without optional parameters by specifying only the variable considered for analysis and the rest of the variables available in the dataset using a generic symbol (e.g., PCDM C033 *) or explicitly—e.g., PCDM C033 A170 C006 C031. The second means a more complex scenario (e.g., if_plus_mix_of_if_and_3arg.do, https://drive.google.com/u/0/uc?id=17HNpLZypin dqT8hZarZjv3O180jn0z3&export=download, accessed on 1 June 2022) when benefiting from the use of the if data subset filtering option (supported by PCDM—lines 6 and 52—Listing A1) for filtering the dataset (e.g., on a certain country code: PCDM C033 * if S003==840) and three optional parameters (line 6—Listing A1, between square brackets, and Figure A1, Appendix A), namely
• minacc—the minimum accepted absolute value (lines 21–29 and 59, Listing A1) of the correlation coefficient (its default value is 0—line 18, Listing A1);
• minn—the minimum accepted number of observations (lines 30–38 and 59, Listing A1) for each response-predictor pair (its default value is 1—line 19, Listing A1);
• maxp—the maximum tolerated \( p \)-value (lines 39–47 and 59, Listing A1) for a significance threshold, usually 0.05 or less (therefore, its default value is 0.05—line 20, Listing A1).

A simple use case relies on a single logical processing core. It also involves the real-time reporting of the number of execution steps out of the total number (the same as the total number of variables in the dataset) along with the execution percentage (lines 61 and 62, Listing A1) and printing of all results or only the ones satisfying those three constraints above (if specified as arguments) in the Stata console (Figure 2).

In a more advanced scenario, the PCDM command (which should work on many platforms, only depending on the location of the personal .ado directory (https://www.stata.com/support/faqs/mac/personal-ado-directory/, accessed on 1 June 2022) appears as invoked inside another one (PCDM4MP, https://drive.google.com/u/0/?id=1_GzZgykKWoZ0j8JuEZ7eiQ4mwaG&export=download accessed on 1 June 2022). The latter was designed for multi-processing purposes (the video instructions in the 2nd recorded simulation, namely, https://drive.google.com/u/0/?id=1_M-LdWMEtfw75zIg8a541V5sc7rk66&export=download, accessed on 1 June 2022) in Stata but only on a Windows physical or virtual machine (reading the number of existing logical cores only considered the case of a Windows OS—local nproc: env NUMBER_OF_PROCESSORS). Using the latter (Figure 3) involves only the target variable and two optional parameters (number of logical cores and destination disk for temporary results) without the rest of the variables and the optional arguments of PCDM. PCDM4MP is optimized for Windows, and it invokes the qsub parallel processing module in Stata [52]. Therefore, qsub is a prerequisite in the sense that it must be installed (ssc install qsub, replace) first. PCDM4MP first displays the starting time (Listing A2, Appendix A, lines 7 and 8). The same when finishing (Listing A2, lines 152 and 153). In addition, it checks many things. One is the number of the total existing (Listing A2, line 33) vs. allocated logical cores (Listing A2, lines 6, 34–43, and 136–138). The latter is optimized (if ‘xc’ > ‘k’ local xc = ‘k’, https://drive.google.com/u/0/?id=1_GzZgykKWoZ0j8JuEZ7eiQ4mwaG&export=download, accessed on 1 June 2022) in order not to overpass the number of vertical splits of the dataset (k groups of variables, according to the starting letter, upper or lower case in their names). Other checks mean verifying whether more than one variable/no variable is used in the command call or simply checking the number of variables in the dataset, its path, and the path of the Stata tool. PCDM4MP also creates a structure of folders on the root of a specified partition/disk (by default C—Listing A2, lines 6, and 44–53, C:\StataMPTasks\queue; C:\StataMPTasks\logs—Listing A2, lines 56–64) and a template file (C:\StataMPTasks\main_do_file.do—Listing A2, lines 65–93) working with two arguments: 1-the task number (Listing A2, lines 72, 124 and 128) in maximum two digits, 2-the starting capital or small letter (Listing A2, lines 80, 89, 124, and 128) for a group of variables to consider in a PCDM correlation command. At runtime, PCDM4MP (Figure 3) will start from this template and will also dynamically generate as many .do files/tasks (maximum 52 in the “queue” subfolder—Listing A2, lines 112–133) as there are variable groups starting with a given letter (upper or lower case), and this was considered because there are many other organizations collecting large datasets (e.g., SHARE-ERIC, http://www.share-project.org/home0.html, accessed on 1 June 2022, e.g., all the variables about work quality start with “wq.”) that use category coding of variables that start with a particular letter or combination of letters. All these tasks will be managed by qsub, which is automatically used (Listing A2, line 139) by PCDM4MP. Consequently, there is no need for further user/custom scripts or setups to generate the template and the tasks, as indicated in the documentation of qsub. When generating tasks, PCDM4MP will also include log generation commands (Listing A2, lines 68–76, and 92), which are necessary to retrieve the results obtained in a parallel manner. Finally, PCDM4MP will
print all the logs (previously generated in the logs subfolder—Listing A2, lines 140–151) in the Stata console. Any user should further copy all this content into a spreadsheet tool, split it into columns using the programmatically generated space separator, and filter it to keep only the correlation results, including additional conditions for minacc, minn, and maxp. This fact (the user is already being asked to copy and filter) is the reason why these three were no longer considered arguments (not even optional) when dealing with multi-processing tasks (PCDM4MP). PCDM4MP is not optimized to support filtering on data subsets (if) either, but this option remains easily available with the aid of a simple script pattern, namely use_filtering_script.do (https://drive.google.com/u/0/uc?id=1yjGsW0fwU1i-PZgv1lnA5Kmy9Gx9U40SaK&export=download, accessed on 1 June 2022) (as six simple command lines) able to extract, export, and reload only a data subset starting from the initial dataset and depending on one or more conditions.

Table 1. The best execution time (approximation in sec.) of PCDM and PCDM4MP for different hardware configurations on WVS data.

| Platform/No.of Allocated Logical Cores (nalc) | Intel Xeon Gold 6240 CascadeLake, 2.6 GHz (VM), SCSI Disk | Intel Xeon Gold 6240 CascadeLake, 2.6 GHz (VM), ImDisk RAMdisk | Intel Core i7 4710HQ, 3.5 GHz (PM), SSD | Intel Core i7 4710HQ, 3.5 GHz (PM), ImDisk RAMdisk | Atom N550, 1.5 GHz (PM), SATA HDD |
|---------------------------------------------|----------------------------------------------------------|----------------------------------------------------------|---------------------------------------|---------------------------------------|----------------------------------|
| 1 (PCDM)                                   | 115 **                                                   | 85                                                       | 85                                    | 800                                    |                                  |
| 2                                           | 50 ***                                                   | 38                                                       | 29                                    | 421                                    |                                  |
| 4                                           | 33 (between 00:04:23 and 00:04:56 in the 4th recorded simulation, namely 4.pcdm4mp-RaaS-IS(4x).ramdisks.mp4 ****) | 30                                    | 28                                    | N/A                                    |                                  |
| 8                                           | 47                                                       | 31                                                       | 29                                    | N/A                                    |                                  |
| 10                                          | 64                                                       | N/A                                                      | N/A                                   | N/A                                    |                                  |
| 12                                          | 74                                                       | N/A                                                      | N/A                                   | N/A                                    |                                  |
| 14                                          | 86                                                       | N/A                                                      | N/A                                   | N/A                                    |                                  |
| 16 (15 really used)                         | 92                                                       | N/A                                                      | N/A                                   | N/A                                    |                                  |

Source: own measurements in Stata 16.0. ** https://drive.google.com/file/d/1coXtC4p6yKo5yfO5fLr5Sr8/view, accessed on 1 June 2022. *** https://drive.google.com/file/d/1oyZ_pyp9sU_G9tiQfPfXVN9_EQD8b1xq/view, accessed on 1 June 2022. **** https://drive.google.com/u/0/?ui=1nN09Syha_-W-B9MfY-J6nmp7Erv0Yy45X&export=download, accessed on 1 June 2022. ***** https://drive.google.com/u/0/?ui=1Md1wDj_iTVcL-Le1vF0ROJmNj7&export=download, accessed on 1 June 2022.

The tests used the above versions of Stata (“Run as Administrator” mode mandatory only for PCDM4MP) and three hardware architectures:

- Intel Xeon Gold 6240 CascadeLake CPU (Central Processing Unit) with 36 virtual processors/logical cores/threads and 18 physical ones, Socket 3647 LGA, 14 nm technology, 2.6 GHz and 32 GB of RAM (Random Access Memory), SCSI Disk, on a Windows Server Datacenter 2019 Virtual Machine (VM—CPU’s bus/core ratio/clock multiplier locked inside the VM, and maximum 32 virtual processors (https://drive.google.com/file/d/1LbbB9Jz3C9SYJH5sRUCkwmSREKoI-_ej/view, accessed on 1 June 2022, configured for use) in a private cloud (https://cloud.raas.uaic.ro, accessed on 1 June 2022) managed using OpenStack on Ubuntu.
- Intel Core i7–4710HQ CPU (8 logical cores, 4 physical ones), Socket 1366 BGA, 22 nm technology, up to 3.5 GHz and 32 GB of RAM, SSD, on a Physical Machine (PM—CPU’s bus/core ratio not locked) using Windows 8.1 Professional x64.
• Intel Atom N550 dual-core CPU (4 logical cores), Socket 559 FCBGA8, 45 nm technology, 1.5 GHz and 2 GB of RAM, SATA HDD, on a PM using the same type of Windows 8.1 above.

Figure 3. More advanced usage scenario involving its version for multi-processing (PCDM4MP) and six logical processing cores on the 2nd hardware configuration described in this paper (Table 1, 4th column).

Notes: Only the first two commands (those two lines starting with “use” and “pcdm4mp”) are the responsibility of the user, while the rest is feedback from the PCDM4MP command in Stata’s console. Otherwise, the same notes as in Figure 2.
3. Results and Discussion

The goal here is to demonstrate the usefulness of the PCDM and PCDM4MP commands, and this is mostly in terms of simplicity and increased support for variable selection. These are based on the results of some tests with both PCDM and PCDM4MP intersected with the ones obtained using other tools and techniques.

Although essentially based on pwcorr (https://www.stata.com/manuals/rcorrelate.pdf, accessed on 1 June 2022) (the pairwise correlation command starting from Pearson’s product-moment method [53,54]—line 52, Listing A1), PCDM has clear advantages over the already existing correlate or pwcorr. The latter is due to its filterable results in a tabular format (Listing A1—the space separators programmatically generated at lines 17 and 59 using the display/di command, and Figure 2) vs. matrices with two headers (Figure A3, Appendix A). This applies in all cases, meaning when considering two or more variables for these already existing correlation commands.

Another advantage of PCDM over other selection methods (e.g., Stepwise, CVLASSO, RLASSO, or BMA) is given even by its specific way of taking pairs of two variables (the chosen one—e.g., C033 or C033_bin and each of the remaining ones). By doing this and reporting and filtering on the number of not null intersecting observations, PCDM can avoid an annoying error, namely No Observations, r(200) other methods confront (Figure A2, Appendix A). The latter is clearly due to non-existent cases/observations at the intersection of all included variables. Therefore, the impossibility of performing statistical computations and the resulting error is expectable. In such cases, PCDM skips the pairs with such problems by using the error capture clause and error type checking with the aid of the _rc (http://www.stata.com/manuals/perror.pdf, accessed on 1 June 2022) (return code) built-in variable (lines 52 and 53, Listing A1).

The first tests of PCDM concerned a simple scenario (the command used was: pcdm C033 *) with those three hardware configurations already mentioned using only a single logical processor core. The whole exploration of the same WVS dataset took between 85 and 124 s, depending on the hardware used (the second line in Table 1).

PCDM also resisted some tests in another more advanced scenario with the command invoked inside the other one, which is optimized for multi-processing (PCDM4MP, Figure 3) on the same three hardware configurations above. PCDM4MP uses PCDM many times (different sessions of Stata) and consequently involves many data loads. This means that PCDM4MP keeps track of the original location and number of variables of the last dataset loaded in the main session of Stata (Listing A2, Appendix A, lines 18–31), and it will send these details (the “main_do_file.do” multi-processing template/pattern—Listing A2, Appendix A, lines 77 and 84) to automatically triggered sessions. The whole parallel exploration of the same dataset comprised 15 distinct unbalanced tasks/jobs (first column and last line in Table 1 and third column in Table 2) corresponding to the same number of variable groups starting with a distinct letter. It took between 36 and 112 s, between 29 and 38, or between 380 and 421 s, depending on the hardware used and the number of logical processor cores allocated (nalc, lines 3–10, and columns 2, 4, and 6 in Table 1). In most cases, this took more than the theoretical nalc part of the previous amount consumed in the single-core approach (the second line in Tables 1 and 2). The exception was the unexpected speed-up (more than double) when going from one to two logical cores for the first two configurations. However, the parallel processing was fast enough. For instance, when using the first configuration (Xeon Gold 6240 CascadeLake, on a VM—Table 1, second column), the execution in the best performing parallel approach (four or six logical cores) was almost 3.5 times (=124/36) faster than using a single-core. A lower ratio (~3) was recorded (=85/29) for the second configuration (Core i7 4710HQ, on a PM—Table 1, fourth column). Moreover, we tried to find out if the specific optimum of four or six logical cores is also due to lower transfer speeds beyond a certain number of concurrent reads for the SSD, SCSI, or SATA storage devices used in these tests. The dataset used occupies 553 MB on all NTFS partitions, and we previously optimized the algorithm behind PCDM4MP to load only each vertical chunk (group of variables) used for computing the correlation coef-
ficients and not the entire dataset (“use <var-list> using <path/dataset-file>” (Listing A2, lines 78–85) instead of just “use <path/dataset-file>” for each different job running on a particular logical core). We noticed that for simultaneous uses of the same storage device (when loading a different part of the same data source into RAM) by each logical CPU (in all tested configurations), an unexpectedly increasing processing time corresponds to increasing parallelism (six or more logical processing cores used). This was more pronounced for the VM (lower CPU frequency and storage devices that involve rotating disks—Table 1, second column) than for the second configuration with a PM (higher CPU frequency and SSDs—Table 1, fourth column). For the latter (based on SSD), the load speed (from disk to RAM) is theoretically divided by the number of concurrent reads, while for the former, this division rule does not apply. This is primarily due to the impossibility of simultaneous access of a read head to several areas on a specific platter of the rotating disc. This translates into dramatic decreases in data loading speed and processing delays. However, in order to eliminate these differences while benefiting from the maximum possible reading speed, we also tested PCDM4MP on the first two configurations using one of the fastest RAM Disk tools, namely ImDisk (https://sourceforge.net/projects/imdisk-toolkit, accessed on 1 June 2022), and two so-called “in RAM” partitions (first—R, of 640 MB, hosting the WVS dataset file, and meant for improving the read speed, and second—Z, of 64 MB, hosting the StataMPtasks temporary folder containing the .do task pattern file, the queue subfolder, and the one with log files, meant for improving the write speeds). As expected, some improvements in the processing time are easily noticeable (Table 1, third and fifth columns). However, its evolution with the increase in the logical parallelism invalidates, beyond a certain threshold (six logical cores, as reported in Table 1), the inverse relationship between the two. To demonstrate that this evolution is not substantially influenced by the behavior of the qsub command on which PCDM4MP is based, we performed an additional simulation (The 6th recorded simulation, namely 6.pcdm-RaaS-IS(15x)RAMdisks-own MPsim without QSUB(same increased time).mp4 (https://drive.google.com/u/0/uc?id=1ij-C4HLXVIAUO-9yF5Ne4Sr7KrtxLdd&export=download, accessed on 1 June 2022)). It used 15 cores simultaneously (the first hardware configuration and using ImDisk), each for every task of those 15 corresponding to the variable groups. This time the corresponding scripts (namely own_sim-autorun15do_files.do (https://drive.google.com/u/0/uc?id=1rcB1MFN5gDMKrKaff11KzFwesrMy8k5Qr&export=download, accessed on 1 June 2022), and own_sim-print15logs.do (https://drive.google.com/u/0/uc?id=1UTNFQb75dfFn2oOKE mwUP61t9NEbVTPv9&export=download, accessed on 1 June 2022)) together with the folder structure to copy on the target disk (in the archive StataMPtasks.zip (https://drive.google.com/u/0/uc?id=1ZXvnGSPQQT4Qi-cTkpfxBMyl3pltsezh&export=download, accessed on 1 June 2022)) were generated without relying on qsub. The results were comparable to those (Table 1, the last line for columns 2 and 3) obtained using PCDM4MP, which finally invokes qsub. Still, for this case (15 logical cores working in parallel and covering all 15 tasks in one execution round), they are far from the theoretical optimum (the maximum of 28 s for the most consuming job/the last one that ends—task no.5, the fourth column and sixth line in Table 2). The closest value when using the same hardware configuration (1st) is obtained with just four cores (Table 1, the third column and fourth line, namely 32 s).
Table 2. The best execution time (approximation in sec.) of PCDM (single logical core) on variable chunks as depending on the starting letter in the name.

| Task No. | Var.Chunk (Starting Letter in var. Names) | No.of.Vars. in the Chunk | Processing Time (Xeon CPU, 1st Config.) | Processing Time (Core i7 CPU, 2nd Config.) | Processing Time (Atom CPU, 3rd Config.) |
|----------|-------------------------------------------|--------------------------|----------------------------------------|----------------------------------------|----------------------------------------|
| 1        | A                                         | 204                      | 25                                     | 20                                     | 173                                     |
| 2        | B                                         | 25                       | 2                                      | 2                                      | 14                                      |
| 3        | C                                         | 43                       | 6                                      | 5                                      | 45                                      |
| 4        | D                                         | 56                       | 6                                      | 6                                      | 49                                      |
| 5        | E                                         | 305                      | 28                                     | 21                                     | 184                                     |
| 6        | F                                         | 129                      | 22                                     | 13                                     | 117                                     |
| 7        | G                                         | 124                      | 13                                     | 9                                      | 88                                      |
| 8        | H                                         | 30                       | 1                                      | 1                                      | 10                                      |
| 9        | I                                         | 2                        | 0                                      | 0                                      | 1                                       |
| 10       | S                                         | 20                       | 4                                      | 3                                      | 27                                      |
| 11       | T                                         | 1                        | 0                                      | 0                                      | 1                                       |
| 12       | V                                         | 7                        | 0                                      | 0                                      | 2                                       |
| 13       | W                                         | 11                       | 1                                      | 1                                      | 3                                       |
| 14       | X                                         | 51                       | 7                                      | 6                                      | 50                                      |
| 15       | Y                                         | 37                       | 8                                      | 6                                      | 58                                      |
| Total    | -                                         | 1045                     | 123                                    | 93                                     | 822                                     |

Source: own measurements in Stata 16.0.

In both cases above (single-core and parallel processing with up to 15 logical cores used), the results are identical (the file 8xCORES-result.xlsx, https://docs.google.com/spreadsheets/d/1KnDQB767F1UHJ4rE2HhS2n8yCtXn0bgT/view, accessed on 1 June 2022) in all tests performed with this WVS dataset. The latter means 332 still filterable lines (excluding the header) when using a single logical processing unit and the same number of lines above if using many logical cores (bottom of Figure 3). In this second case (parallel processing), the same results emerged after filtering on the first column, preserving only two specific entries, namely, the target variable (C033) and the first part of the header (Outcome(y)), then sorting on both this header part above and another one (Input(X)) and finally removing the header duplicates at the end.

Two similar tests using the corr_var function (the commands inside R(corr_var).txt, https://drive.google.com/file/d/174HdKZ5jy02ICQUWMrHefQMJjmaRg6l/view, accessed on 1 June 2022) from the Lares package in R, version 4.1.3 x64, and the .rds (R) format of the same dataset took much more time (between 30 and 45 min) when compared with the PCDM (Stata, single-core mode, 85 sec., Table 1, fourth col.). By contrast, both tests using corr_var ended with errors related to memory allocation (Error: cannot allocate vector of size 1.6 MB, Error: cannot allocate vector of size 18.6 GB) even if using the same (second) hardware configuration (Intel Core i7). In terms of resources consumed, corr_var and R used up to 15 GB of RAM (the first test) and 24 GB (the second), while PCDM and Stata just up to 900 MB.

In addition, the Rattle library of R was considered for performing correlations (the explore section). The main drawback consisted of the impossibility to set a target. Rattle was able to identify (Rattle-explore-correl.png, https://drive.google.com/file/d/1MCX6RDe_U3KAbm0LOMly7JheV5A5Uuz/view, accessed on 1 June 2022) from the Lares package in R, version 4.1.3 x64, and the .rds (R) format of the same dataset took much more time (between 30 and 45 min) when compared with the PCDM (Stata, single-core mode, 85 sec., Table 1, fourth col.). By contrast, both tests using corr_var ended with errors related to memory allocation (Error: cannot allocate vector of size 1.6 MB, Error: cannot allocate vector of size 18.6 GB) even if using the same (second) hardware configuration (Intel Core i7). In terms of resources consumed, corr_var and R used up to 15 GB of RAM (the first test) and 24 GB (the second), while PCDM and Stata just up to 900 MB.

Moreover, in Weka (both versions 3.8.6 and 3.9.6), we selected the CorrelationAttributeEval technique and the Pairwise CorrelationAttributeEval (both requiring the use of a Ranker search method), but these two techniques could not be applied (Weka-correl nominal var.png, https://drive.google.com/file/d/1En5xrlHgb_ZSfo9SE0hPMF276MZGWd/view, accessed on 1 June 2022) to any of the nominal and binary forms of the
target variable (C033, C033_bin), although the modules have been enabled in the Package Manager and the variables and their values met the requirements (Weka-correl and other req.png, https://drive.google.com/file/d/1GQBqomT7_dEVCBTSjWvUmgqY7DdKTyK/view, accessed on 1 June 2022). By contrast, the ClassifierAttributeEval (https://drive.google.com/file/d/1RnMZZ4dYenii2gxXITscqqP8_FEyFh-W/view, accessed on 1 June 2022) (not focused on correlations), which is also using the Ranker search method (on the full training set), was successfully applied, although it lasted tens of minutes using both the first and the second hardware configuration. The same for ClassifierSubsetEval (https://drive.google.com/file/d/1iLkRw3CtwG9j-XuQ6eVheBdgtgsa-/view, accessed on 1 June 2022) (also not focused on correlations) using the GreedyStepwise method and ten folds cross-validation. The last two confirmed the validity of the data set used but not of the two correlation packages in Weka (Explorer application module).

In the case of using larger datasets (e.g., the default maximum number of variables in both versions of Stata used is 5000, and 120,000 as the maximum possible), because the console (results) window should handle a considerable output, a dedicated command (e.g., set scrolbufsize 2048000) to increase it (maximum size in bytes) is needed (https://stats.ora.cc.ucla.edu/stata/faq/how-can-i-make-the-results-window-hold-more-results, accessed on 1 June 2022).

Some attempts to automatically parallelize the execution of PCDM using another module for multi-processing (the parallel command (https://github.com/gvegayon/parallel, accessed on 1 June 2022) in Stata) succeeded but with different correlation results than in the single-core mode (when using the entire dataset), and this is because the parallel command is optimized to work just with horizontal subsets/data chunks as groups of records but not with vertical ones (groups of fields/variables) as required by PCDM. Or, unless dealing with rare exceptions (an additionally recorded simulation, namely 7.pcdm8xRaaS(exception for parallel).avi (https://drive.google.com/u/0/uc?id=1zoD5ji_dOGNQEBW6SO4E93LsQjuBx4bXl&export=download, accessed on 1 June 2022) about the inappropriate use of the parallel module (horizontal splits) via two custom scripts, namely sim.do (https://drive.google.com/file/d/1ilx02KsYyzEY_gYGh6QQZPUakjwnjirP/view, accessed on 1 June 2022) and parallel_sim.do (https://drive.google.com/file/d/1EF2GqFFh0nBDPMutSUxVsS1-mm81gUZ/view, accessed on 1 June 2022)), e.g., all non-null records accidentally included in a single horizontal chunk up to the threshold of eight logical cores used for this example of this specific WVS dataset), this approach using the parallel command is doomed to end up with different correlation coefficients than those resulting when considering the whole dataset. Even so, similar parallelization attempts of pwcorr (the command PCDM relies on “pwcorr C033 *, obs sig” instead of “pcdm C033 *” in a .do file) using the parallel command failed (a seemingly endless execution loop). This difference occurred because the atomic tasks in the existing pwcorr command (viewsource pwcorr.ado) are inside some while loops with a priori-unknown iteration space. Therefore, they are hard to parallelize [35]. By contrast, PCDM uses invocations of pwcorr only on pairs of variables (less time consuming and less likely to encounter situations without common observations/“No Observations” error) as most atomic operations in a finite loop with a priori-known iteration space (forvalues—Listing A1, lines 49–63). Under these circumstances, using different Stata instances with distinct classes of variables from the same dataset and resulting from intuitive name filtering (simply using * after one or more than one common initial letter provided) remains a handy and feasible parallelization approach. These are strong arguments in favor of using PCDM and PCDM4MP instead of the consecrated command pwcorr (bottom of Figure A3, the Appendix A) and relying on qsub instead of parallel when it comes to time-consuming data-mining tasks in Stata to parallelize by extracting vertical chunks of data.

The pre-processing responsible for generating the binary form of the variable to analyze before exporting the Stata native .dta format to .csv (necessary for comparatively testing using other tools and techniques) is also available (preProcessingScript.do, https://drive.go
The alternative selection stage based on Adaptive Boosting and some tuning parameters [56,57] in the Rattle library of R served the triangulation [58] as a scientific principle. It discovered in a ranked way (Figure 4) the most important variables related to the one to analyze in its binary form (C033_bin).

Figure 4. Alternative results as obtained using the Adaptive Boosting technique in the Rattle library of R.

Additional filters (Figure 5 and the practical example at the end of the fifth recorded simulation, namely 5.pcdm4mp-RaaS-IS(16x).mp4 (https://drive.google.com/u/0/uc?id=1iMdiIwDR_iiVv0C-Le1vF0lROJmvNjJ7&export=download, accessed on 1 June 2022)) on the results obtained in the console further served analysis purposes. Such results came after simple invocations of PCDM for both forms of the outcome (“pcdm C033 *” and “pcdm C033_bin *”) and were previously copied in a spreadsheet file. The first was the exclusion of C033 and C033_bin from the list of values for the input (general common-sense condition). Next came the specification of the first constraint, namely ≥ 0.2 for ACC [59,60]. Another restriction (≥ 10,000) was a subjective one for the number of observations, meaning ~2/3 or more of the entire support for the variable to analyze (15,968 valid records—top of Figure 6). More, an additional one for the p-values (≤ 0.001) followed. After checking the results, only the following list of seven strong intersecting influences and corresponding variables emerged: A008, A170, A173, C006, C031, C034, and D002.

The nearest equivalent of the PCDM commands (“pcdm C033 *, minacc(0.2) minn(10000) maxp(0.001)” and “pcdm C033_bin *, minacc(0.2) minn(10000) maxp(0.001)” for the user-mode visual filters (Figure 5) is also available, and this comes together with the corresponding results. For instance, in Figure 6, both resulting lists except the autocorrelation in the first reported line with values (just below the header as the second one printed in the console, after invoking the PCDM command) and the seventh line with printed values (bottom of Figure 6) showing the correlation with the source variable, namely between C033_bin and C033. The difference of one unit (1045 vs. 1046) between the total amounts of steps needed (Figure 6—corresponding to the total number of variables) is
due to the derivation of the binary form of the outcome (lines 3–5 in Figure 1) performed between the first and the second invocation of PCDM. The reason for an additional test clause (if) to remove autocorrelation that is not available in the source script of PCDM is an efficiency-related one. The latter means not oversizing the processing time (already large when using just a single logical processing core and large datasets such as this time series version of WVS).

| Outcome(y) | Input(x) | Correl.Cof.(CC) | Abs.Val.CC(ACC) | No.Obs.(Nobs) | Signif.(p) |
|-----------|----------|----------------|----------------|--------------|------------|
| C033      | A008     | -0.275599473   | 0.275599473     | 15566        | 2.45E-269  |
| C033      | A009     | -0.21503478    | 0.21503478      | 14907        | 1.79E-155  |
| C033      | A170     | 0.448367207    | 0.448367207     | 15848        | 0          |
| C033      | A173     | 0.267281432    | 0.267281432     | 15680        | 1.10E-254  |
| C033      | C006     | 0.392538393    | 0.392538393     | 15811        | 0          |
| C033      | C031     | -0.548267082   | 0.548267082     | 14900        | 0          |
| C033      | C034     | 0.457145428    | 0.457145428     | 15811        | 0          |
| C033      | D002     | 0.393719848    | 0.393719848     | 15752        | 0          |
| C033      | G006     | -0.211638309   | 0.211638309     | 15576        | 3.21E-157  |
| C033      | Y010     | -0.205139136   | 0.205139136     | 15782        | 1.48E-149  |
| C033      | Y011B    | -0.211512282   | 0.211512282     | 15576        | 4.96E-157  |

| Outcome(bin) | Input(x) | Correl.Cof.(CC) | Abs.Val.CC(ACC) | No.Obs.(Nobs) | Signif.(p) |
|---------------|----------|----------------|----------------|--------------|------------|
| C033_bin      | A008     | -0.232290544   | 0.232290544     | 15566        | 9.35E-190  |
| C033_bin      | A170     | 0.371341451    | 0.371341451     | 15848        | 0          |
| C033_bin      | A173     | 0.220887675    | 0.220887675     | 15680        | 1.46E-172  |
| C033_bin      | C006     | 0.332663899    | 0.332663899     | 15811        | 0          |
| C033_bin      | C031     | -0.458191035   | 0.458191035     | 14900        | 0          |
| C033_bin      | C034     | 0.367811471    | 0.367811471     | 15811        | 0          |
| C033_bin      | D002     | 0.317981159    | 0.317981159     | 15752        | 0          |

Figure 5. Seven intersecting results after two selection rounds using PCDM in its simple format for both forms of the outcome and further visual filters in spreadsheet tools (Microsoft Office Excel). Note: The “E” followed by the minus (“−”) and numbers indicate the E notation corresponding to the scientific one (2.45E-269 is actually 2.45 × 10^{-269}).

If further applying CVLASSO and RLASSO selection techniques in many rounds until no loss (2xLASSO.do, https://drive.google.com/u/0/uc?id=1Lw4mjmX1Ua2QDLaVZxRWxE9dwNi0aRJ&export=download, accessed on 1 June 2022) for both forms of the outcome (C033 and C033_bin), they converge to a shortlist of just five intersecting variables, namely A170, C006, C031, C034, and D002. All these five are also available in the list returned after using the Adaptive Boosting technique in the Rattle library of R (Figure 4). If additionally using BMA (2xBMA.do, https://drive.google.com/u/0/uc?id=1j8uK8EGxLcrolWIrUsxilPKEZ9h_Yb&export=download, accessed on 1 June 2022) as Bayesian Model Averaging [17] and considering A008 and A173 as auxiliary predictors, the posterior inclusion probability (pip) for these two seems to be lower than 5%, while for the rest of the predictors is close to 99.99% when considering both forms of the outcome. When considering their pairwise correlation with both formats of the variable to analyze, these two auxiliary predictors are the only ones having an absolute value of the correlation coefficient below 0.3 from the previous list of seven common possible predictors (Figure 5). This value above is considered by many authors [61–63] a low one.
https://drive.google.com/u/0/uc?id=1j8uK8EGxLEcroLWIrUsxIiPKEZS9h_Yb&export=download, accessed on 1 June 2022) as Baye-
sonian Model Averaging [17] and consid-
ering A008 and A173 as auxiliary predictors, the posterior inclusion probability (pip) for these two seems to be lower than 5%, while for the rest of the predictors is close to 99.99% when considering both forms of the outcome. When considering their pairwise correlation with both formats of the variable to analyze, these two auxiliary predictors are the only ones having an absolute value of the correlation coefficient below 0.3 from the previous list of seven common pos-
sible predictors (Figure 5). This value above is considered by many authors [61–63] a low one.

Figure 6. Similar intersecting results using PCDM on a single logical processing core for both forms of the outcome and all three optional arguments for specifying the minimum/maximum limits. Notes: The same as in Figure 2.

Of course, for obtaining robust regression models, further checks are required to elimi-
nate reverse causality (Table A3, rev_cause_checks_logit.do, https://drive.google.com/u/0/
uc?id=1cWajOE8ylkoy3gzKdpPBwR4mqqBnuO00&export=download, accessed on 1 June 2022) and collinearity issues (Table A4, collin_rem_and_comp_perf_checks.do, https://driv-
e.google.com/u/0/?c?id=181SbannhNljr9vgrE6jmsycmNwxwL_Fgf&export=download, ac-
essed on 1 June 2022), but not before performing additional derivations for these five variables (Table A1, additional_processing_script.do, https://drive.google.com/u/0/u
uc?id=1WRwIdmiB3uBC66c6y-WMwAX74BqSkcV&export=download, accessed on 1 June 2022, and Table A2). However, all previous selections are easy to perform with the aid of PCDM, and this is obvious if comparing this with the scenario when start-
ing from all variables and using only CVLASSO, RLASSO, or BMA in Stata. In the latter case, the corresponding commands will return the same error mentioned above (No observations—BMA_and_LASSO_NoObsErrs.do, https://drive.google.com/u/0/
uc?id=1ZHT4Ge7WhFjD8y8qUFK1BYuNk5jyLN7n&export=download, accessed on 1 June 2022). Moreover, PCDM also supports cross-validations on well-established criteria [64] or targeted ones [65,66] via the mix of using both the if statement for filtering the dataset and those three arguments presented above and meant for filtering the correlation results obtained (if_plus_mix_of_if_and_3arg.do, https://drive.google.com/u/0/?c?id=17HNpLZypindqT8hZarZjv3O1B80jn0z3&export=download, accessed on 1 June 2022).

Figure 6. Similar intersecting results using PCDM on a single logical processing core for both forms of the outcome and all three optional arguments for specifying the minimum/maximum limits. Notes: The same as in Figure 2.

Of course, for obtaining robust regression models, further checks are required to elimi-
nate reverse causality (Table A3, rev_cause_checks_logit.do, https://drive.google.com/u/0/
uc?id=1cWajOE8ylkoy3gzKdpPBwR4mqqBnuO00&export=download, accessed on 1 June 2022) and collinearity issues (Table A4, collin_rem_and_comp_perf_checks.do, https://driv-
e.google.com/u/0/?c?id=181SbannhNljr9vgrE6jmsycmNwxwL_Fgf&export=download, ac-
essed on 1 June 2022), but not before performing additional derivations for these five variables (Table A1, additional_processing_script.do, https://drive.google.com/u/0/u
uc?id=1WRwIdmiB3uBC66c6y-WMwAX74BqSkcV&export=download, accessed on 1 June 2022, and Table A2). However, all previous selections are easy to perform with the aid of PCDM, and this is obvious if comparing this with the scenario when start-
ing from all variables and using only CVLASSO, RLASSO, or BMA in Stata. In the latter case, the corresponding commands will return the same error mentioned above (No observations—BMA_and_LASSO_NoObsErrs.do, https://drive.google.com/u/0/
uc?id=1ZHT4Ge7WhFjD8y8qUFK1BYuNk5jyLN7n&export=download, accessed on 1 June 2022). Moreover, PCDM also supports cross-validations on well-established criteria [64] or targeted ones [65,66] via the mix of using both the if statement for filtering the dataset and those three arguments presented above and meant for filtering the correlation results obtained (if_plus_mix_of_if_and_3arg.do, https://drive.google.com/u/0/?c?id=17HNpLZypindqT8hZarZjv3O1B80jn0z3&export=download, accessed on 1 June 2022).
The methodology used in this paper also stands on the scientific principle of triangulation [58,67,68]. The latter means to use various methods, techniques, and tools and obtain results that agree across all of them, for instance, data mining based on pairwise correlation coefficients, Adaptive Boosting, BMA, LASSO variable selection techniques, reverse causality and collinearity checks, different regressions, post-estimations of accuracy and goodness of fit, maximum absolute values for correlation coefficients among influences, and predictors, and dynamic thresholds for variance inflation factors.

4. Conclusions

Although essentially based on pairwise correlations, PCDM and its version for multiprocessing (PCDM4MP) are new tools compared with the existing ones (not just in Stata but also in R or Python). Both bring additional functionalities and serve for selecting the most important influences to include in regression and classification models. They also report the exploration progress in real-time depending on the hardware processing power (in most cases, the CPU specifications, and the RAM and storage amount and speed) together with the number of variables existing and specified from a dataset. PCDM4MP also supports parallelism and emulates a cluster environment up to a certain level by triggering different instances of Stata using distinct classes of variables resulting from intuitive name filtering (first letter). The paper also describes this parallel version supporting an approach oriented towards time-consuming data-mining tasks in Stata and some benchmark results against different hardware configurations used for processing. The description includes the automatic generation of a dynamic task pattern, tasks, and logs. The main consequence is that these tools reduce the time needed to generate filterable tabular results based on absolute values of correlation coefficients and their corresponding significance and support, all reported in a record-oriented and transparent manner. In addition, they successfully overcome annoying errors such as „No observations” by their pairs of variables-oriented nature. The paper describes both tools and brings real-world examples of using large datasets to prove the support provided by PCDM and PCDM4MP for exploring reliable influences and even determinants of different variables to analyze.

Author Contributions: Conceptualization, D.H. and D.A.; methodology, D.H.; software, D.H.; validation, D.H. and D.A.; formal analysis, D.A.; investigation, D.H. and D.A.; resources, D.H.; data curation, D.H.; writing—original draft preparation, D.H.; writing—review and editing, D.H. and D.A.; visualization, D.H.; supervision, D.A.; project administration, D.A.; funding acquisition, D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any funding in terms of publishing fees. Still, it benefited from the infrastructure purchased via the project mentioned in the Acknowledgments section below.

Institutional Review Board Statement: The data used in this study belong to the World Values Survey, which conducted surveys following the Declaration of Helsinki.

Informed Consent Statement: The World Values Survey obtained informed consent from all subjects involved in the study.

Data Availability Statement: The dataset used in this study and belonging to the World Values Survey is the .dta file inside the “WVS TimeSeries 1981 2020 Stata v1 6.zip” archive (https://www.worldvaluessurvey.org/WVSDocumentationWVL.jsp, accessed on 1 June 2022, the “Data and Documentation” menu, the “Data Download” option, the “Timeseries (1981–2022)” entry).

Acknowledgments: For allowing the exploration of the dataset and the agreement to publish the research results, we would like to thank the World Values Survey and supporting projects. As technical assistance (https://cloud.raas.uaic.ro, accessed on 1 June 2022), as a private cloud of the Alexandru Ioan Cuza University of Iasi, Romania), this paper benefited from the support of the Competitiveness Operational Programme Romania. More precisely, project number SMIS 124759—RaaS-IS (Research as a Service Iasi) id POC/398/1/124759, coordinated by Marin Fotache, to whom we are grateful. We would also like to thank Cristina Tirnauca, Department of Mathematics, Statistics, and Computation, Faculty of Sciences, University of Cantabria, Santander, Spain, for her useful advice.
Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Listing A1. The source script of PCDM with numbered lines—numbers displayed separately, as when opened with the Stata editor.

1 *! version 1.1 12July2022
2 *Authors: Daniel HOMOCIANU & Dinu AIRINEI
3 *Ex1.: pcdm C033 * *Ex2.: pcdm C033 C031 C034 C036 C037 C038 C039 if S003==840
4 *Ex3.: pcdm C033 * if S003==840, minacc(0.15) minn(1000) maxp(0.001)
5 program define pcdm
6 syntax varlist [if] [ , minacc(real 0) minn(real 1) maxp(real 0.05)]
7 local datetime = `c(current_date)’ `c(current_time)’
8 di “PCDM STARTED AT: ’datetime’”
9 local k : word count ’varlist’
10 if ’k’ < 2 {
11  di as error “ Error: Provide at least 2 variables!”
12  exit
13 }
14 local y : word 1 of ’varlist’
15 local xvarlist: list varlist -y
16 local npred = ’k’−1
17 di “Outcome(y) Input(x) Correl.Coeff.(CC) Abs.Val.CC(ACC) No.Obs.(Nobs) Signif.(p)”
18 local macc=0
19 local mn=1
20 local mp=0.05
21 if !missing(“’minacc’”) {
22  if ’minacc’>=0 & ’minacc’<=1 {
23    local macc=’minacc’
24  }
25  else {
26    di as err “Error: parameter minacc(min.ACC) must be >=0 and <=1!”
27    exit
28  }
29 }
30 if !missing(“’minn’”) {
31  if ’minn’>=1 {
32    local mn=’minn’
33  }
34  else {
35    di as err “Error: parameter minn(min.Nobs.) must be an integer >=1!”
36    exit
37  }
38 }
39 if !missing(“’maxp’”) {
40  if ’maxp’>=0 & ’maxp’<=0.05 {
41    local mp=’maxp’
42  }
43  else {
44    di as err “Error: parameter maxp(max.p) must be >=0 and <=0.05!”
45    exit
46  }
47 }
48 local k=0
49 forvalues i = 1(1) ‘npred’ {
50 local k =’k’ + 1
51 local x : word ‘i’ of ‘xvarlist’
52 capture pwcorr ‘y’ ‘x’ ‘if’, sig
53 if _rc==0 {
54 matrix crlv=vec(r(C))
55 local CC=crlv [1,2]
56 local ACC=abs(‘CC’)
57 local Nobs = r(N)
58 local p = r(sig) [1,2]
59 if ‘ACC’>‘macc’ & ‘p’<=‘mp’ & ‘Nobs’>‘mn’ di “‘y’ ‘x’ ‘CC’ ‘ACC’ ‘Nobs’ ‘p’”
60 }
61 local perc=int(‘k’/‘npred’*100)
62 window manage maintitle “Step ‘k’ of ‘npred’ (‘perc’% done)”
63 }
64 window manage maintitle “Stata”
65 local datetime = “‘c(current_date)’ ‘c(current_time)’”
66 di “PCDM FINISHED AT: ‘datetime’”
67 end

Listing A2. The source script of PCDM4MP with numbered lines.

1 */! version 1.1 12July2022
2 *Authors: Daniel HOMOCIANU & Dinu AIRINEI
3 *Ex1.: pcdm4mp C033 *Ex2.: pcdm4mp C033, xcpu(4) *Ex3.: pcdm4mp wq727_,
4 program define pcdm4mp
5 version 16.0
6 syntax varlist [‘, xcpu(real 2) disk(string)]
7 local datetime = “‘c(current_date)’ ‘c(current_time)’”
8 di “PCDM4MP STARTED AT: ‘datetime’”
9 local k : word count ‘varlist’
10 if ‘k’ < 1 {
11 di as error “ Error: Provide the target variable!”
12 exit
13 }
14 if ‘k’>1 {
15 di “ Warning: For MP tasks only the 1st variable (target) will be considered!”
16 }
17 local Y : word 1 of ‘varlist’
18 ***get the path of the current dataset and its no.of vars.***
19 local dataset=”‘c(filename)’”
20 local dsetnvars=’c(k)’+150
21 if ‘dsetnvars’ < 2048 {
22 local dsetnvars=2048
23 }
24 if ‘dsetnvars’ > 120,000 {
25 di as error “ Error: The dataset is too large (>120,000 vars.)!”
26 exit
27 }
28 if missing(“‘dataset’”) {
29 di as error “ Error: First you must open a dataset!”
30 exit
31 }
32 ***check the CPU config.***
33 local nproc : env NUMBER_OF_PROCESSORS
34 local xc=2
35 if !missing("'xcpu'") {
36 if 'xcpu'>=2 & 'xcpu'<= 'nproc' {
37 local xc=int('xcpu')
38 }
39 else {
40 di as error " Error: Provide at least 2 logical CPU cores (but no more than 'nproc') for MP tasks!"
41 exit
42 }
43 }
44 local dsk="C"
45 if !missing("'disk'") {
46 if "'disk'"<="Z" | "'disk'"<="Z" {
47 local dsk="'disk'"
48 }
49 else {
50 di as error " Error: Provide a valid disk letter!"
51 exit
52 }
53 }
54 di "pcdm4mp will save temporary results at 'dsk':\StataMPtasks and also below!"
55 ***Generating the “main_do_file.do” MP template***
56 local smpt_path="'dsk':\StataMPtasks"
57 shell rd "'smpt_path'/s/q
58 qui mkdir "'smpt_path'
59 local full_do_path="'smpt_path'\main_do_file.do"
60 local q_subdir="queue"
61 qui mkdir "'smpt_path'/'q_subdir'"
62 local queue_path="'smpt_path'\q_subdir"
63 local l_subdir="logs"
64 qui mkdir "'smpt_path'/'l_subdir'"
65 local logs_path="'smpt_path'\l_subdir"
66 qui file open mydofile using "'full_do_path'", write replace
67 file write mydofile "clear all" _n
68 file write mydofile "log using "
69 file write mydofile "'logs_path'\log"
70 file write mydofile "'logs_path'\log"
71 file write mydofile "'logs_path'\log"
72 file write mydofile "1" _n
73 file write mydofile "set maxvar 'dsetnvars'" _n
74 file write mydofile "use 'Y' " _n
75 file write mydofile "80 file write mydofile "2" _n
76 file write mydofile "82 file write mydofile "'dataset'" _n
77 file write mydofile "'dataset'" _n
78 file write mydofile "use 'Y' " _n
79 file write mydofile "83 file write mydofile "'dataset'" _n
80 file write mydofile "'dataset'" _n

86 file write mydofile "pcdm"
87 file write mydofile "Y"
88 file write mydofile "z"
89 file write mydofile "2"
90 file write mydofile "*"
91 file write mydofile "_n"
92 file write mydofile "log close"
93 qui file close mydofile
94 ***Finding the Stata dir.***
95 local _sys='c(sysdir_stata)'
96 local exec : dir ""_sys"" files "Stata*.exe", respect
97 foreach exe in `exec' {
98 if inlist('exe','Stata.exe','Stata-64.exe','StataMP.exe','StataMP-64.exe',
StataSE.exe','StataSE-64.exe') {
99 local curr_st_exe 'exe'
100 continue, break
101 }
102 }
103 local st_path='""_sys""'"'curr_st_exe"
104 capture confirm file ""_sys""curr_st_exe""
105 if _rc !=0 {
106 di as error "Stata’s sys dir and executable NOT found!"
107 exit
108 }
109 else {
110 di "!!!Stata’s sys dir and executable found: 'st_path' !!!"
111 }
112 ***Creating and configuring .do files***
113 clear all
114 set maxvar ‘dsetnvars’
115 use ‘dataset’
116 local k=0
117 foreach letter in 'c(alpha)' & 'c(ALPHA)' {
118 if 'letter'<="z" | 'letter'<="Z" {
119 capture ds ‘letter’*
120 if !_rc {
121 local k = 'k' + 1
122 if ‘k’<10 {
123 qui file open mydofile using ‘queue_path’job0’k’.do, write replace
124 qui file write mydofile "'dsk':StataMPtasks\main_do_file.do" 0’k’ ‘letter’
125 }
126 if ‘k’>10 {
127 qui file open mydofile using ‘queue_path’job’k’.do, write replace
128 qui file write mydofile "'dsk':StataMPtasks\main_do_file.do" ’k’ ‘letter’
129 }
130 file close mydofile
131 }
132 }
133 }
134 ***Allocating .do tasks to CPU using qsub v.13.1 (06/10/2015), created by Adrian
Sayers.***
135 *ssc install qsub, replace
136 if ‘xc’>'k' {
137 local xc='k'
138 }
139 qsub , jobdir('queue_path') maxproc('xc') statadir('st_path') deletelogs
140 ***Printing logs for all .do tasks in the main session’s console***
141 local mylogs : dir ""logs_path"" files ".txt"
142 local k=0
143 foreach entry in 'mylogs' {
144 local k =`k’ + 1
145 if ‘k’<10 {
146 type ""logs_path”’log0`k’’.txt"
147 }
148 if ‘k’>=10 {
149 type ""logs_path”’log`k’’.txt"
150 }
151 }
152 local datetime = "’c(current_date) ’c(current_time)’"
153 di “PCDM4MP FINISHED AT: `datetime’"
154 end

Figure A1. Errors when not providing enough variables or exceeding the minimum/maximum thresholds of those three PCDM parameters.
Notes: The same as the first two in Figure 2.

Figure A2. Discovery limitations when using the cvlasso, rlasso, and bma commands in Stata.
Note: The same as the first one in Figure 2.
Figure A3. Discovery and filtering limitations when using the `correlate` and `pwcorr` commands in Stata.

Notes: The same as the first two in Figure 2. The “e” followed by the plus (“+”) and numbers indicate the E notation corresponding to the scientific one (4.3e+05 is actually 4.3 × 10^5).
Table A1. The outcome and the most resilient five possible predictors selected after using PCDM, LASSO, and BMA.

| Variable | Question                                                                 | Coding                                                                 |
|----------|--------------------------------------------------------------------------|------------------------------------------------------------------------|
| C033     | Job satisfaction—DEPENDENT VARIABLE                                      | 1-Dissatisfied . . . 10-Satisfied                                      |
| C033_bin | Job satisfaction (binary format)—DEPENDENT VARIABLE                     | 1 if C033!=. & C033>=6 0 if C033!=. & C033<6 & C033>0                  |
| A170     | Satisfaction with your life                                              | 1-Dissatisfied . . . 10-Satisfied                                      |
| A170_bin | Satisfaction with your life (binary format)                              | 1 if A170!=. & A170>=6 0 if A170!=. & A170<6 & A170>0                 |
| C006     | Satisfaction with the financial situation of household                  | 1-Dissatisfied . . . 10-Satisfied                                      |
| C006_bin | Satisfaction with the financial situation of household (binary format)   | 1 if C006!=. & C006>=6 0 if C006!=. & C006<6 & C006>0                  |
| C031     | Degree of pride in your work                                             | 1-A great deal . . . 4-None                                           |
| C031_bin | Degree of pride in your work (binary format)                             | 1 if C031!=. & C031<=2 & C031>0 0 if C031!=. & C031>2                  |
| C034     | Freedom of decision taking in the job                                   | 1-Not at all . . . 10-A great deal                                    |
| C034_bin | Freedom of decision taking in the job (binary format)                    | 1 if C034!=. & C034>=6 0 if C034!=. & C034<6 & C034>0                  |
| D002     | Satisfaction with home life                                             | 1-Dissatisfied . . . 10-Satisfied                                      |
| D002_bin | Satisfaction with home life (binary format)                              | 1 if D002!=. & D002>=6 0 if D002!=. & D002<6 & D002>0                 |

Source: WVS data and own calculations in Stata using the following commands: label list, generate, and replace.

Table A2. Descriptive statistics for the variable to analyze and those most resilient five possible predictors selected after using PCDM, LASSO, and BMA.

| Variable | N     | Mean | Std.Dev. | Min | Median | Max |
|----------|-------|------|----------|-----|--------|-----|
| C033     | 15,968| 7.27 | 2.31     | 1   | 8      | 10  |
| C033_bin | 15,968| 0.77 | 0.42     | 0   | 1      | 1   |
| A170     | 420,669| 6.7  | 2.42     | 1   | 7      | 10  |
| A170_bin | 420,669| 0.69 | 0.46     | 0   | 1      | 1   |
| C006     | 411,461| 5.75 | 2.58     | 1   | 6      | 10  |
| C006_bin | 411,461| 0.54 | 0.5      | 0   | 1      | 1   |
| C031     | 14,988| 1.73 | 0.87     | 1   | 2      | 4   |
| C031_bin | 14,988| 0.51 | 0.5      | 0   | 1      | 1   |
| C034     | 17,900| 6.54 | 2.79     | 1   | 7      | 10  |
| C034_bin | 17,900| 0.65 | 0.48     | 0   | 1      | 1   |
| D002     | 25,653| 7.72 | 2.24     | 1   | 8      | 10  |
| D002_bin | 25,653| 0.83 | 0.38     | 0   | 1      | 1   |

Source: Own calculations in Stata using the univar command.
Table A3. Reverse causality checks using binary logistic regressions for job satisfaction and each potential predictor from those five resulting after using PCDM, LASSO, and BMA.

| Model | (1)     | (2)     | (3)     | (4)     | (5)     | (6)     | (7)     | (8)     | (9)     | (10)    |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Predictors/Response var. | C033_bin | A170_bin | C033_bin | C006_bin | C033_bin | C031_bin | C033_bin | C034_bin | C033_bin | D002_bin |
| A170  | 0.3973 *** | 0.3300 *** | (0.0097) | (0.0084) |         |         |         |         |         |         |
| C006  |         |         |         |         | −1.2461 *** | (0.0263) |         |         |         |         |
| C031  |         |         |         |         |         |         |         |         | 0.3233 *** | (0.0077) |
| C034  |         |         |         |         |         |         |         |         |         |         |
| D002  |         |         |         |         |         |         |         |         | 0.3264 *** | (0.0092) |
| C033  | −1.3840 *** | −1.2868 *** | (0.0646) | (0.0618) | −1.8448 *** | (0.0473) | −1.8024 *** | (0.0640) | −0.7322 *** | (0.0475) |
| _cons |         |         |         |         | 3.5825 *** | (0.0576) |         |         | −1.9542 *** | (0.0638) |
| N     | 15,848  | 15,848  | 15,811  | 15,811  | 14,900  | 14,900  | 15,811  | 15,811  | 15,752  | 15,752  |
| chi²  | 1681.6969 | 1511.4602 | 1558.7477 | 1406.8425 | 2237.2495 | 2034.7577 | 1771.5264 | 1851.9322 | 1253.1181 | 1212.2401 |
| p     | 0.0000  | 0.0000  | 0.0000  | 0.0000  | 0.0000  | 0.0000  | 0.0000  | 0.0000  | 0.0000  | 0.0000  |
| pseudo R² | 0.1258 | 0.1060 | 0.1046 | 0.0800 | 0.1833 | 0.2168 | 0.1244 | 0.1204 | 0.0880 | 0.0993 |
| AUCROC | 0.7433 | 0.7129 | 0.7272 | 0.6797 | 0.7667 | 0.8095 | 0.7377 | 0.7280 | 0.6912 | 0.7193 |
| AIC   | 14,832.3486 | 15,908.2089 | 15,176.6194 | 19,733.4346 | 13,249.4465 | 10,656.0603 | 14,786.6002 | 17,607.5324 | 15,391.2067 | 12,641.8199 |
| BIC   | 14,847.6902 | 15,923.5505 | 15,191.9563 | 19,748.7715 | 13,264.6647 | 10,671.2786 | 14,801.9971 | 17,622.8693 | 15,406.5362 | 12,657.1493 |

Source: Own calculations in Stata.
Notes: Robust standard errors are between parentheses; all raw coefficients above parentheses emphasized using *** are significant at 1‰; green vs. red means better comparative performance and variables that are more likely to be predictors (green in the 1st column) rather than response ones (red).
### Table A4. Comparative regression models for predicting job satisfaction (C033_bin) after removing reverse causality and collinearity issues and performing additional checks.

| Model | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) | (11) | (12) |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| **Regression Type** | logit | OLS | logit | OLS | logit | OLS | OLS | logit | logit | OLS | OLS | OLS |
| **Filter Condition** | N/A | N/A | N/A | N/A | if C006!=. | if A170!=. | if C006!=. | if A170!=. | N/A | N/A | N/A | N/A |
| **Predictors/Response var.** | C033_bin | A170 | C006 | C031 | C034 | D002 | _cons | N | chi2 | p | R² | RMSE |
| Filter Condition | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| **Model (1)** | 0.1867 *** | 0.0258 *** | 0.2667 *** | 0.0416 *** | 0.3433 *** | 0.0535 *** | 0.3441 *** | 0.0536 *** | 0.0132 | 0.0111 | 0.0102 | 0.0115 | 0.0115 |
| **Model (2)** | 0.1423 *** | 0.0169 *** | 0.1851 *** | 0.0254 *** | 0.2780 *** | 0.0409 *** | 0.2776 *** | 0.0409 *** | 0.0115 | 0.0102 | 0.0115 | 0.0115 | 0.0115 |
| **Model (3)** | 0.1286 *** | 0.0258 *** | 0.1579 *** | 0.0254 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.0127 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |
| **Model (4)** | 0.1045 ** | 0.0258 *** | 0.1376 *** | 0.0401 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.019 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |
| **Model (5)** | 0.0907* | 0.0258 *** | 0.1173 ** | 0.0401 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.0208 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |
| **Model (6)** | 0.0764 | 0.0258 *** | 0.0970 | 0.0401 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.0225 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |
| **Model (7)** | 0.0622 | 0.0258 *** | 0.0767 | 0.0401 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.025 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |
| **Model (8)** | 0.0480 | 0.0258 *** | 0.0564 | 0.0401 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.0275 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |
| **Model (9)** | 0.0338 | 0.0258 *** | 0.0361 | 0.0401 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.0305 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |
| **Model (10)** | 0.0206 | 0.0258 *** | 0.0168 | 0.0401 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.0335 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |
| **Model (11)** | 0.0074 | 0.0258 *** | 0.0074 | 0.0401 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.0365 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |
| **Model (12)** | -0.0006 | 0.0258 *** | -0.0107 | 0.0401 *** | 0.2784 *** | 0.0409 *** | 0.2791 *** | 0.0409 *** | 0.0409 | 0.0115 | 0.0115 | 0.0115 | 0.0115 |

**Source:** Own calculations in Stata.  
**Notes:** Robust standard errors are between parentheses; all raw coefficients above parentheses emphasized using *** are significant at 1‰; green vs. red indicates better comparative performance and, consequently, better models; red alone indicates unacceptable collinearity (OLSmaxComputVIF>OLSmaxAcceptVIF) or moderate correlation between predictors (maxAbsVPMCC).
References

1. Baker, M. Why scientists must share their research code. *Nature* 2016. [CrossRef]

2. Matareasa, V. Kinds of replicability: Different terms and different functions. *Axiomathes* 2022, 1–24. [CrossRef]

3. Homocianu, D.; Ploepeau, A.-P.; Janole-Calin, R. A Robust Approach for Identifying the Major Components of the Bribery Tolerance Index. *Mathematics* 2021, 9, 1570. [CrossRef]

4. Rajah, K.; Sivarasa, S.; Maharajian, M.K. Impact of Pharmacists’ Interventions and Patients’ Decision on Health Outcomes in Terms of Medication Adherence and Quality Use of Medicines among Patients Attending Community Pharmacies: A Systematic Review. *Int. J. Environ. Res. Public Health* 2021, 18, 4392. [CrossRef] [PubMed]

5. Sadeghi, A.R.; Bahadori, Y. Urban Sustainability and Climate Issues: The Effect of Physical Parameters of Streetscape on the Thermal Comfort in Urban Public Spaces; Case Study: Karimkhan-e-Zand Street, Shiraz, Iran. *Sustainability* 2021, 13, 10886. [CrossRef]

6. Thanh, M.T.G.; Toan, N.; Toan, D.T.T.; Thang, N.P.; Dung, N.Q.; Dung, N.T.; Hang, P.T.T.; Anh, L.Q.; Tra, N.T.; Ngoc, V.T.N. Diagnostic Value of Fluorescence Methods, Visual Inspection and Photographic Visual Examination in Initial Caries Lesion: A Systematic Review and Meta-Analysis. *Dent. J.* 2021, 9, 30. [CrossRef]

7. Wang, L.; Ling, C.-H.; Lai, P.-C.; Huang, Y.-T. Can The ‘Speed Bump Sign’ Be a Diagnostic Tool for Acute Appendicitis? Evidence-Based Appraisal by Meta-Analysis and GRADE. *Life* 2022, 12, 138. [CrossRef] [PubMed]

8. Damasceno, E.; Azevedo, A.; Pérez-Cota, M. Data mining, business intelligence, grid and utility computing: A bibliometric review of the literature from 2015 to 2020. In Proceedings of the 23rd International Conference on Enterprise Information Systems, Prague, Czech Republic, 26–28 April 2021; Volume 1, pp. 367–373. [CrossRef]

9. Kopf, O.; Homocianu, D. The Business Intelligence Based Business Process Management Challenge. *Inform. Econ. J.* 2016, 20, 7–19. [CrossRef]

10. Studer, S.; Bui, T.B.; Drescher, C.; Hanuschkin, A.; Winkler, L.; Peters, S.; Müller, K.-R. Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Mach. Learn. Knowl. Extr.* 2021, 3, 392–413. [CrossRef]

11. Bendel, R.B.; Afifi, A.A. Comparison of stopping rules in forward “stepwise” regression. *J. Am. Stat. Assoc.* 1977, 72, 46. [CrossRef]

12. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Methodol.* 1996, 58, 267–288. [CrossRef]

13. Sanchez, J.D.; Régo, L.C.; Ospina, R. Prediction by Empirical Similarity via Categorical Regressors. *Mach. Learn. Knowl. Extr.* 2019, 1, 641–652. [CrossRef]

14. Ahrens, A.; Hansen, C.B.; Schaffer, M.E. Lassopack: Model selection and prediction with regularized regression in Stata. *Stata J.* 2020, 20, 176–235. [CrossRef]

15. Bilger, M. Overfit: Stata module to calculate shrinkage statistics to measure overfitting as well as out- and in-sample predictive bias. *Stat Soft. Comp.* 2015, S457950. Available online: https://EconPapers.repec.org/RePEc:boc:bocode:s457950 (accessed on 1 June 2022).

16. Gao, Y.; Cowling, M. Introduction to Panel Data, Multiple Regression Method, and Principal Components Analysis Using Stata: Study on the Determinants of Executive Compensation—A Behavioral Approach Using Evidence from Chinese Listed Firms; SAGE Publications Ltd.: Thousand Oaks, CA, USA, 2019. [CrossRef]

17. De Luca, G.; Magnus, J.R. Bayesian model averaging and weighted-average least squares: Equivariance, stability, and numerical issues. *Stata J.* 2011, 11, 518–544. [CrossRef]

18. Karabulut, E.M.; Ploepeau, A.-P.; Janole-Calin, R. A Robust Approach for Identifying the Major Components of the Bribery Tolerance Index. *Mathematics* 2021, 9, 1570. [CrossRef]

19. Carabulut, E.; Ibrìkçi, T. Analysis of cardiotocogram data for fetal distress determination by decision tree based adaptive boosting approach. *J. Comput. Commun.* 2014, 2, 32–37. [CrossRef]

20. Schönlieu, M. Boosted regression (boosting): An introductory tutorial and a Stata plugin. *Stata J.* 2005, 5, 330–354. [CrossRef]

21. Zlotnik, A.; Abraira, V. A general-purpose nomogram generator for predictive logistic regression models. *Stata J.* 2015, 15, 537–546. [CrossRef]

22. Zdravevski, E.; Lameski, P.; Kulakov, A.; Trajov, D.; Jakimovski, S. Parallel computation of information gain using Hadoop and mapreduce. *Ann. Comput. Sci. Inf. Syst.* 2015, 10, 43–53. [CrossRef]

23. Oancea, B.; Dragoescu, R.M. Integrating R and Hadoop for Big Data Analysis, Romanian Statistical Review. arXiv 2014, arXiv:1407.4908. [CrossRef]

24. Meng, X.; Bradley, J.; Yavuz, B.; Sparks, E.; Venkataraman, S.; Liu, D.; Freeman, J.; Tsai, D.B.; Amde, M.; Owen, S.; et al. MLlib: Machine Learning in Apache Spark. arXiv 2015, 2015:06807. [CrossRef]

25. Fotache, M.; Cuci, M.-I. Big Data Performance in private clouds. In Some initial findings on Apache Spark Clusters deployed in OpenStack; In Proceedings of the 2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet), Iasi, Romania, 4–6 November 2021. [CrossRef]

26. Li, J.; Zhang, C.; Zhang, J.; Qin, X.; Hu, L. MICS-P:parallel mutual-information computation of big categorical data on Spark. *J. Parallel Distrib. Comput.* 2022, 161, 118–129. [CrossRef]

27. Khoshaba, F.; Kareem, S.; Awla, H.; Mohammed, C. Machine learning algorithms in Bigdata analysis and its applications: A Review. In Proceedings of the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 9–11 June 2022; pp. 1–8. [CrossRef]

28. Murty, C.S.; Saradhi Varma, G.P.; Satyanarayana, C. Content-based collaborative filtering with hierarchical agglomerative clustering using user/item based ratings. *J. Interconnect. Netw.* 2022. [CrossRef]
28. Aldabbas, H.; Albashish, D.; Khatatneh, K.; Amin, R. An architecture of IOT-aware healthcare smart system by leveraging machine learning. *Int. Arab. J. Inf. Technol.* **2019**, *19*, 160–172. [CrossRef]

29. Alhussan, A.A.; AIEisa, H.N.; Attea, G.; Solouna, N.H.; Seoud, R.A.; Ayoub, O.S.; Ghoneim, V.F.; Samee, N.A. ForkJoinPcc algorithm for computing the PCC matrix in gene co-expression networks. *Electronics* **2022**, *11*, 1174. [CrossRef]

30. Huckvale, E.D.; Hodgman, M.W.; Greenwood, B.B.; Stucki, D.O.; Ward, K.M.; Ebbert, M.T.; Kauwe, J.S.; Miller, J.B. Pairwise Correlation Analysis of the Alzheimer’s disease neuroimaging initiative (ADNI) dataset reveals significant feature correlation. *Genes* **2021**, *12*, 1661. [CrossRef][PubMed]

31. Ye, R.; Fang, B.; Du, W.; Luo, K.; Lu, Y. Bootstrap Tests for the Location Parameter under the Skew-Normal Population with Unknown Scale Parameter and Skewness Parameter. *Mathematics* **2022**, *10*, 921. [CrossRef]

32. Airinei, D.; Homocianu, D. The Importance of Video Tutorials for Higher Education—The Example of Business Information Systems. In Proceedings of the 6th International Seminar on the Quality Management in Higher Education, Tulcea, Romania, 8–9 July 2010. Available online: https://ssrn.com/abstract=2381817 (accessed on 1 June 2022).

33. Michelucci, U.; Venturini, F. Estimating Neural Network’s Performance with Bootstrap: A Tutorial. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 357–373. [CrossRef]

34. Airinei, D.; Homocianu, D. The Geographical Dimension of DSS Applications. *Sci. Ann. Alexandru Ioan Cuza Univ. Iasi* **2009**, *56*, 637–642. Available online: https://econpapers.repec.org/RePEc:cai:journl:y:2009:v:56:p:637-642 (accessed on 1 June 2022).

35. Goodwin, J.L.; Williams, A.L.; Snell Herzog, P. Cross-Cultural Values: A Meta-Analysis of Major Quantitative Studies in the Last 50 Years. *Religions* **2020**, *11*, 396. [CrossRef]

36. Ortega-Gil, M.; Mata García, A.; ElHichou-Ahmed, C. The Effect of Ageing, Gender and Environmental Problems in Subjective Well-Being. *Land* **2021**, *10*, 1314. [CrossRef]

37. Miniesy, R.S.; AbdelKarim, M. Generalized Trust and Economic Growth: The Nexus in MENA Countries. *Economies* **2021**, *9*, 39. [CrossRef]

38. Lim, S.B.; Malek, J.A.; Yigitcanlar, T. Post-Materialist Values of Smart City Societies: International Comparison of Public Values for Good Enough Governance. *Future Internet* **2013**, *5*, 201. [CrossRef]

39. Vo, T.T.D.; Tuliao, K.V.; Chen, C.-W. Work Motivation: The Roles of Individual Needs and Social Conditions. *Behav. Sci.* **2022**, *12*, 49. [CrossRef]

40. Sánchez-García, J.; Gil-Lacruz, A.I.; Gil-Lacruz, M. The influence of gender equality on volunteering among European senior citizens. *Volunt. Int. J. Volunt. Nonprofit Organ.* **2022**. [CrossRef]

41. Fakhri, A.; Makdissi, P.; Marrouch, W.; Tabri, R.V.; Yazbeck, M. A stochastic dominance test under survey nonresponse with an application to comparing trust levels in Lebanese public institutions. *J. Econom.* **2023**, 228, 342–358. [CrossRef]

42. Freund, R.J.; Wilson, W.J. *Regression Analysis: Statistical Modeling of a Response Variable*, 2nd ed.; Academic Press: Cambridge, UK, 2006.

43. Batcheva, P.K.; Lee, M.; McCormick, J.B.; Rahbar, M.H. Multicollinearity in regression analyses conducted in epidemiologic studies. *Epidemiol. Sunnyvale Open Access* **2015**, *6*, 227. [CrossRef]

44. Arabameri, A.; Asadi Nalivian, O.; Chandra Pal, S.; Chakrabortty, R.; Saha, A.; Lee, S.; Pradhan, B.; Tien Bui, D. Novel Machine Learning Approaches for Modelling the Gully Erosion Susceptibility. *Remote Sens.* **2020**, *12*, 2833. [CrossRef]

45. Pepe, M.S.; Cai, T.; Longton, G. Combining predictors for classification using the area under the receiver operating characteristic curve. *Biometrics* **2005**, *62*, 221–229. [CrossRef]

46. Carreras, J.; Hamoudi, R. Artificial Neural Network Analysis of Gene Expression Data Predicted Non-Hodgkin Lymphoma Subtypes with High Accuracy. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 720–739. [CrossRef]

47. Espinheira, P.L.; da Silva, L.C.M.; Silva, A.d.O.; Ospina, R. Model Selection Criteria on Beta Regression for Machine Learning. *Mach. Learn. Knowl. Extr.* **2019**, *19*, 427–449. [CrossRef]

48. Dziak, J.J.; Coffman, D.L.; Lanza, S.T.; Li, R.; Jermiin, L.S. Sensitivity and specificity of information criteria. *Brief. Bioinform.* **2019**, *21*, 553–565. [CrossRef]

49. Jimenez, J.; Navarro, L.; Quintero, M.C.G.; Pardo, M. Multivariate Statistical Analysis for Training Process Optimization in Neural Networks-Based Forecasting Models. *Appl. Sci.* **2021**, *11*, 3552. [CrossRef]

50. Pearson, K. Mathematical contributions to the theory of evolution—III. Regression, heredity, and panmixia. *Philos. Trans. R. Soc. Lond. Ser. A* **1896**, *187*, 253–318.

51. Pearson, K.; Filon, L.N.G. Mathematical contributions to the theory of evolution. IV. On the probable errors of frequency constants and on the influence of random selection on variation and correlation. *Philos. Trans. R. Soc. Lond. Ser. A* **1898**, *191*, 229–311.

52. Rauchwerger, L.; Padua, D. Parallelizing while loops for multiprocessor systems. In Proceedings of the 9th International Parallel Processing Symposium, Santa Barbara, CA, USA, 25–28 April 1995; pp. 347–356. [CrossRef]
56. Chen, Y.-K.; Li, W.; Tong, X. Parallelization of AdaBoost algorithm on multi-core processors. In Proceedings of the 2008 IEEE Workshop on Signal Processing Systems 2008, Washington, DC, USA, 8–10 October 2008; pp. 275–280. [CrossRef]

57. Williams, G. Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery; Springer: Berlin/Heidelberg, Germany, 2011; pp. 269–291.

58. Munafò, M.R.; Smith, G.D. Robust research needs many lines of evidence. Nature 2018, 553, 399–401. [CrossRef] [PubMed]

59. Schober, P.; Boer, C.; Schwarte, L.A. Correlation coefficients. Anesth. Analg. 2018, 126, 1763–1768. [CrossRef]

60. Mukaka, M.M. Statistics corner: A guide to appropriate use of correlation coefficient in medical research. Malawi Med. J. 2012, 24, 69–71.

61. Corlett, M.T.; Pethick, D.W.; Kelman, K.R.; Jacob, R.H.; Gardner, G.E. Consumer Perceptions of Meat Redness Were Strongly Influenced by Storage and Display Times. Foods 2021, 10, 540. [CrossRef]

62. Lace, J.W.; Handal, P.J. Psychometric Properties of the Daily Spiritual Experiences Scale: Support for a Two-Factor Solution, Concurrent Validity, and Its Relationship with Clinical Psychological Distress in University Students. Religions 2017, 8, 123. [CrossRef]

63. Berthold, D.P.; Morikawa, D.; Muench, L.N.; Baldino, J.B.; Cote, M.P.; Creighton, R.A.; Denard, P.J.; Gobezie, R.; Lederman, E.; Romeo, A.A.; et al. Negligible Correlation between Radiographic Measurements and Clinical Outcomes in Patients Following Primary Reverse Total Shoulder Arthroplasty. J. Clin. Med. 2021, 10, 809. [CrossRef] [PubMed]

64. Roberts, D.R.; Bahn, V.; Ciuti, S.; Boyce, M.S.; Elith, J.; Guillera-Arroita, G.; Hauenstein, S.; Lahoz-Monfort, J.J.; Schröder, B.; Thuiller, W.; et al. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography 2017, 40, 913–929. [CrossRef]

65. Link, W.A.; Sauer, J.R. Bayesian Cross-Validation for Model Evaluation and Selection, with Application to the North American Breeding Survey. Ecology 2015, 97, 1746–1758. [CrossRef] [PubMed]

66. Bayerl, P.S.; Akhgar, B. Surveillance and falsification implications for open source intelligence investigations. Commun. ACM 2015, 58, 62–69. [CrossRef]

67. Giacomello, G.; Martinelli, D. Crystal Clear: Investigating Databases for Research, the Case of Drone Strikes. Data 2021, 6, 124. [CrossRef]

68. Sierras-Davo, M.C.; Lillo-Crespo, M.; Verdu, P.; Karapostoli, A. Transforming the Future Healthcare Workforce across Europe through Improvement Science Training: A Qualitative Approach. Int. J. Environ. Res. Public Health 2021, 18, 1298. [CrossRef]