DOI: 10.47743/eufire-2024-1-9

REMOVING DK/NA VALUES IN SHARE, WVS, OR SIMILAR DATASETS. EFFECTS ON THE EXPLORATION OF PREDICTIVE MODELS

DANIEL HOMOCIANU
Alexandru Ioan Cuza University of Iasi
Iasi, Romania
daniel.homocianu@uaic.ro

Abstract
This paper describes the effects of using a tool capable of automatically removing DK/NA (Do Not Know/No Answer) values from some tabular datasets. For these values, the original encoding performed by some providers of significant survey datasets (e.g., SHARE, WVS, etc.) is as negative numbers. To leave them as are means to accept an artificial increase of scales. Or that translates into dramatic changes in feature selection, exploration tasks, and performance measurements of the resulting models. The tool discussed in this paper helps avoid manually recoding or deriving cleaned replicas of the existing variables in such datasets. In a transparent manner (progress tracking), this tool automatically detects all variables specified, treats each of them, and generates immediate results corresponding to the treatment status (including exceptions for string ones). The paper also brings examples of using real-world data (World Values Survey-WVS, Time-series, v4.0).

Keywords: *SHARE, WVS, or similar datasets; DK/NA coded as negative values; effects on regression and classification models; feature selection steps; performance metrics.*
JEL Classification: L63, C31, C55, D83.

1. INTRODUCTION

Nowadays large datasets have become increasingly prevalent in social sciences and other research domains. Moreover, many statistical tools such as SPSS, R, MATLAB, Minitab, SAS, Stata, facilitate data analysis, statistical computations, visual representations, advanced tests, and automate the entire process of generating the results, many under the form of regression models with coefficients, errors, and other performance metrics.

Some of the owners of the datasets the research community (mostly those based on surveys) uses sometimes not so common ways of encoding variable values. One example is that of some scales configured in the reverse order of the natural intensity of the corresponding responses translated into values related to certain variables. For instance, the five points scale of 5 - 1 (decreasing) for something that varies between Very Weak and Very Strong and should have an inverse correspondence (e.g., 1 for Very Weak, 2 for Weak, 3 for So and So, 4 for Strong, and 5 for Very Strong). Another example (in fact, the one to which this paper is devoted) concerns DK/NA values coded as "Do not Know" (DK), "No Answer" (NA), "Not Asked" (NA), "Not Applicable" (NA) or Missing/Unknown, which should typically be considered missing values. These values represent respondents' lack of knowledge or refusal to provide an answer to specific survey questions or simply the impossibility of collecting a valid answer due to other reasons (Couper *et al.,* 2001). DK/NA values are typically treated as missing and coded accordingly (Williams *et al.,* 2018). The latter applies because many researchers aim for clear and trustful answers, and the accuracy of the classification models obtained critically depends on the treatment procedures for missing values (Acuña and Rodriguez, 2004). However, some significant providers of survey datasets adopt an alternative approach and encode these values as negative ones, introducing a notable challenge in data analysis by leading to artificial inflation of scales, potential distortions in the estimation of correlation coefficients, generation of statistical models, and interpretation of statistical coefficients together with affecting measures of multi-collinearity and accuracy. Moreover, such a coding scheme can complicate feature selection efforts, hinder the robust exploration of models, lead to biased conclusions, and hinder the generalizability of research findings. Therefore, addressing this issue and restoring the integrity of the data is crucial for model exploration, feature selection tasks, accurate data analysis, and reliable results.

Traditional approaches to address this problem involve data imputation techniques, where missing values turn into estimated values based on various statistical methods and observed patterns in the data. Various imputation methods (e.g., mean imputation, hot-deck imputation, or multiple imputation) evolved to handle missing values effectively (Farhangfar *et al.,* 2007). These imputation methods require careful consideration and can introduce additional uncertainties and biases into the data. However, in the case of DK/NA values encoded as negative numbers, the usual imputation approaches may not be appropriate. Imputing these negative values using standard imputation techniques can introduce further biases and distort the nature of the data. Thus, a tailored approach is required to address this specific coding scheme and accurately handle DK/NA values.

REMDKNA (a new Stata command) tackle such issues being designed to automatically remove DK/NA values from large survey datasets, avoid consuming extra time and effort to clean the dataset, and obtain cleaner datasets ensuring the integrity, validity, and reliability of the statistical analyses and resulting models. The tool was inspired by working with large data sets, such as the ones belonging to significant providers as the European Values Study a large-scale, cross-national, and longitudinal survey of attitudes, opinions and values produced by Tilburg University and partners; GESIS of the Leibniz

Institute for Social Sciences; the World Values Survey, a global research project that explores people's values and beliefs; the Survey of Health Ageing and Retirement in Europe / SHARE of the Munich Center for the Economics of Aging (MEA), a former division of the Max Planck Institute for Social Law and Social Policy; and the Life in Transition Survey/LITS of the European Bank for Reconstruction and Development. The idea came after carefully identifying the procedures for coding the values of the fields/variables related to the questions in the questionnaires used by these providers, *label list* and *tabulate* commands in Stata, as used for the dta form of the datasets.

Previous research (Zhang, 2011; Liu *et al.,* 2012) emphasized the challenges of using DK/NA values in large survey datasets and their impact on data analysis and modeling. Traditional approaches often involve laborious manual recoding or imputation techniques (Young *et al.,* 2011), which can be time-consuming and prone to errors (Assale *et al.,* 2019). Addressing DK/NA values as negative ones is not something new. It stands behind studies utilizing large survey datasets like the World Values Survey (WVS), SHARE, or others indicated in the previous section. Such datasets provide valuable insights into societal trends and attitudes over time. However, the artificially encoded negative values can introduce bias and affect the reliability of statistical analyses.

REMDKNA offers a promising solution to this issue. By automatically identifying variables specified in the command and processing them in real-time, REMDKNA efficiently removes DK/NA values. The tool provides information on the treatment of such values for each variable, including numerical success or string exception codes and survey item descriptions/labels. Moreover, the paper includes real-world examples starting from a WVS dataset and illustrating the impact of REMDKNA on model evaluation metrics, predictor independence tests, and classification model accuracy.

2. DATA AND METHODS

REMDKNA in Stata is recommended to be used together with "*label list*" and "*tabulate*" (existing commands) that can support an easy check of the original coding logic and frequency of values for any variable. For all three, the results are presented in the console of Stata. Moreover, other community-based ones served to generate comparative predictive model representations, with and without DK/NA value treatment, such as probability/risk-prediction nomograms based on *nomolog*, in the context of binary logistic regressions (Zlotnik and Abraira, 2015). Such prediction nomograms were also necessary for two things. First, it is about the selection between collinear/redundant variables. Second, the representation of the final binary logistic models working also as prediction instruments. Some of the additional installations performed were those of the *estout* package, including support for using the *eststo*, *esttab* (Jann, 2005; Jann and Long, 2010), PCDM (Homocianu and Airinei, 2022) and MEM commands. (Homocianu and Tîrnăucă, 2022). Estout served to automatically generate tables with coefficients and errors corresponding to the regression models by printing them in the console or even exporting them as .csv files. PCDM was meant for filtering on magnitude, support, and significance after performing calculations regarding Pearson Pairwise Correlations, while MEM complemented *estout* with additional performance metrics - e.g., AUC-ROC (Jiménez-Valverde, 2012).

The original design of REMDKNA was as a .do (batch) script file. In this case, its execution depended on its download path. Now, it serves as a command publicly available for Stata under this peculiar name (REMDKNA), and it also supports a set of variables or the Asterix (*) instead of all existing variables, as the only parameter. To install it is enough to download and copy the remdkna.ado addition to Stata file into one of the ado directories, e.g., C:\ado\personal. When designing our own Stata processing files, .do extension to automate the derivation and analysis steps or even the generation of tables with classification and regression models and their performance statistics, we can rely on the simple logic of invoking REMDKNA right after opening the original dataset.

REMDKNA can dynamically and efficiently deal with all existing variables in a dataset, use of * instead of specifying a long list of space-separated variable names, which is also possible. It also looks for exceptions when proving no variable and prevents fatal interruptions in execution when detecting other issues, e.g., string variables - easy to find later by searching after the EXCEPTION keyword, and easy to remove by relying on the use of the *drop* command. REMDKNA also measures the execution step (variable) and percentage, which is useful when dealing with time-consuming tasks involving many variables to clean. Because of capturing both the job start and job end timestamps, it also supports measuring the time needed to complete such tasks. By that it simultaneously acts as a benchmarking instrument for different hardware and software configurations.

A subsequent scenario of finding string variables generating exceptions and dropping them from the dataset is worth mentioning when further performing selections, e.g., CVLASSO and RLASSO in the LASSO pack (Tibshirani, 1996) and not wanting to explicitly specify so many variables from the dataset but use instead the Asterix (*) to refer them all at once. Therefore, the data treatment effort is minimized as much as possible by relying on such dynamic and real-time treatment and reporting transparently performed by REMDKNA in the case of datasets like the ones mentioned in the Introduction part of this article.

For other cases, datasets in the native .dta format of Stata or imported files such as .csv or .xls/.xlsx) in which the negative values of some variables do not correspond to DK/NA values, the use of the REMDKNA command should be performed with a lot of care, in order not to alter/destruct the original scales of these variables. Data from the World Values Survey (WVS) prove the usefulness of REMDKNA for real-world examples. All variables (1,045) and observations (450,869) in a WVS dataset (the file named WVS_TimeSeries_4_0.dta available in the .zip archive, namely WVS TimeSeries 1981 2022 Stata v4 0.zip, supported the tests. Data was exported then as .csv in two forms (two forms depending on whether we have previously applied or not REMDKNA) using Stata. This export took place only after a simple binary derivation of the variable to analyze (D002, Satisfaction with home life) considering the symmetric split of the original scale. Therefore, starting from it (original scale of 1=Dissatisfied to 10=Satisfied), D002bin was derived. The latter contains one for all original values greater than or equal to 6 and 0 otherwise. Other commands (label list, tabulate, generate, and replace) seemed handy when checking the original scales or the frequency of values and performing the derivations. Then, both forms of the .csv file, exported dataset acted as input in the Rattle version 5.5.1, interface of R, version 4.1.3, x64.

Next, the Adaptive Boosting (Ada Boost) technique for decision tree classifiers (Vadivukkarasi and Santhi, 2020) as the 1st part of the 1st selection round was applied with default settings Trees:50, Max Depth:6, Min Split:20, Complexity:0.01, Learning Rate:0.3, Threads:2, Iterations:50, Objective: binary logistic. The reason for not using the boost plugin in Stata is related to its time-consuming execution coupled with limited capabilities in terms of automatic variable selection and treatment of missing values (Schonlau, 2005). Simultaneously, 2nd part of the 1st selection round, a filter based on Pearson correlation coefficients, between the target variable in its scale format and all other variables was applied using PCDM (Homocianu and Airinei, 2022) and two types of threshold conditions, 0.2 for the minimum value of the correlation coefficient as absolute value or modulus, and 0.001 for the maximum accepted p-value for which the correlation coefficient is significant.

Another selection stage (2nd round) focused on discovering the intersection of the selections previously performed using Ada Boost and Pearson correlation coefficients.

In the 3rd round, we successively invoked two powerful commands in the LASSO package (Tibshirani, 1996) in Stata for both forms of the outcome (binary and scale) and both forms of the .dta dataset (when previously using or not REMDKNA) until observing no loss in selections. The list of predictors obtained in the 2nd round served as input for this one, round 3. To perform such consecutive selections, two powerful commands in this LASSO pack of Stata served, namely *rlasso* – responsible for controlling overfitting (Sanchez *et al.,* 2019), and *cvlasso* – performing cross-validations on random subsamples (Ahrens *et al.,* 2020).

In the 4th round (performed only starting from the previous findings in the scenario when initially removing the DK/NA values), the selections based on checking for reverse causality issues using ordered logit and ordered probit regressions, the target (D002) and just one predictor, each from those identified at the previous step, in every single regression plus interchanging the roles (D002 as the predictor and each of the others as input) and comparing some metrics, e.g., Pseudo R-squared – the larger, the better, and AIC and BIC – the smaller, the better, in each direction.

The 5th round, also performed only starting from the previous findings in the scenario when initially removing the DK/NA values, selected variables using risk prediction nomograms and the results of Ada Boost (frequency of splits for each input variable) only after identifying collinearity issues among predictors based on R-squared and VIF comparisons in OLS regressions with two (x1 x2) and three (y x1 x2) variables specified, and Pearson correlations coefficients (Liveris *et al.,* 2014) not overpassing (in their absolute values) a maximum of 0.4, this time computed only for pairs of predictors and not between the target and each potential predictor – the case of the 2nd part of the 1st selection round.

Stata 17.0 MP 2021 64-bit was behind most of the derivation, selection, and analysis steps preceded by using or not the REMDKNA tool proposed in this paper.

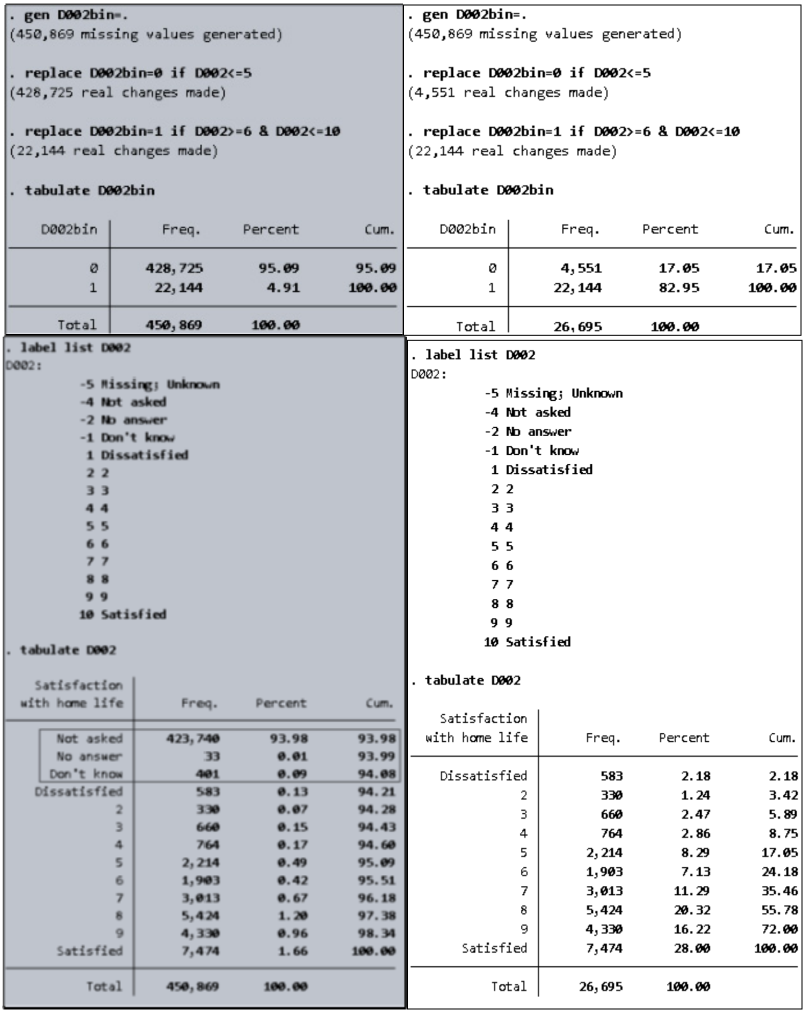
3. RESULTS AND DISCUSSION

The goal of this section is to demonstrate the usefulness of the REMDKNA command in terms of increased support for treatments of DKNA values of most numerical variables and the consequences of not applying such treatment in terms of differences in descriptive statistics (Tables A1 and A2), resulting sets of variables corresponding to various selection steps, collinearity and reverse causality measurements, and accuracy of other performance indicators for final models. Moreover, after identifying peculiar influences at the end of the first three selection rounds mentioned in the previous section, additional tests and analyses were performed considering both forms of the outcome variable (D002 for the scale and D002bin for the binary form).

As noticed in Figure 1, the result of the 2nd tabulation on center-left, the derivation process when not previously using REMDKNA is doomed to failure unless explicitly specifying a secondary condition (>0) for the lower half (1-5) of the original ten-points scale when creating the 0 (zero) values of the binary derivation. The latter is due to the negative values (e.g., -5 for Missing/Unknown, -4 for Not asked, -2 for No answer, and -1 for Don't know, where only the last three have a frequency for the target variable, D002) used by the owners of this dataset to code the DK/NA values.

In the case of using REMDKNA and automatically dropping all missing values for all variables, coded as negative ones, this will result in fewer valid values, only the ones corresponding to the scale used) for each variable and a much lighter (right of Figure 2) exported data set (.csv format to use with other tools, such as Rattle in R).

When performing the first selection based on Ada Boost in Rattle with the target variable set in its binary format, this tool selected only 123 variables (Figure 3B) from all existing in the dataset if previously not dropping the DK/NA values and only 65 (Figure 3A) if using REMDKNA immediately after opening the dataset.



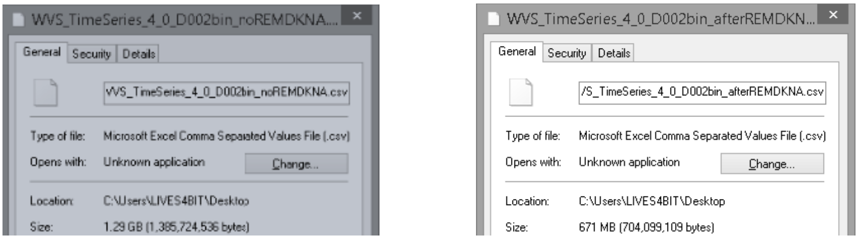
```
. gen D002bin=.
(450,869 missing values generated)

. replace D002bin=0 if D002<=5
(428,725 real changes made)

. replace D002bin=1 if D002>=6 & D002<=10
(22,144 real changes made)

. tabulate D002bin

    D002bin |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |    428,725       95.09       95.09
          1 |     22,144        4.91      100.00
------------+-----------------------------------
      Total |    450,869      100.00
```

```
. gen D002bin=.
(450,869 missing values generated)

. replace D002bin=0 if D002<=5
(4,551 real changes made)

. replace D002bin=1 if D002>=6 & D002<=10
(22,144 real changes made)

. tabulate D002bin

    D002bin |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      4,551       17.05       17.05
          1 |     22,144       82.95      100.00
------------+-----------------------------------
      Total |     26,695      100.00
```

```
. label list D002
D002:
          -5 Missing; Unknown
          -4 Not asked
          -2 No answer
          -1 Don't know
           1 Dissatisfied
           2 2
           3 3
           4 4
           5 5
           6 6
           7 7
           8 8
           9 9
          10 Satisfied

. tabulate D002

 Satisfaction |
with home life |      Freq.     Percent        Cum.
--------------+-----------------------------------
    Not asked |    423,740       93.98       93.98
    No answer |         33        0.01       93.99
   Don't know |        401        0.09       94.08
 Dissatisfied |        583        0.13       94.21
            2 |        330        0.07       94.28
            3 |        660        0.15       94.43
            4 |        764        0.17       94.60
            5 |      2,214        0.49       95.09
            6 |      1,903        0.42       95.51
            7 |      3,013        0.67       96.18
            8 |      5,424        1.20       97.38
            9 |      4,330        0.96       98.34
    Satisfied |      7,474        1.66      100.00
--------------+-----------------------------------
        Total |    450,869      100.00
```

```
. label list D002
D002:
          -5 Missing; Unknown
          -4 Not asked
          -2 No answer
          -1 Don't know
           1 Dissatisfied
           2 2
           3 3
           4 4
           5 5
           6 6
           7 7
           8 8
           9 9
          10 Satisfied

. tabulate D002

 Satisfaction |
with home life |      Freq.     Percent        Cum.
--------------+-----------------------------------
 Dissatisfied |        583        2.18        2.18
            2 |        330        1.24        3.42
            3 |        660        2.47        5.89
            4 |        764        2.86        8.75
            5 |      2,214        8.29       17.05
            6 |      1,903        7.13       24.18
            7 |      3,013       11.29       35.46
            8 |      5,424       20.32       55.78
            9 |      4,330       16.22       72.00
    Satisfied |      7,474       28.00      100.00
--------------+-----------------------------------
        Total |     26,695      100.00
```

Figure 1. Comparative results of applying tabulations for the target variable in both forms (binary derivation and original form)



WVS_TimeSeries_4_0_D002bin_noREMDKNA....

General | Security | Details

WVS_TimeSeries_4_0_D002bin_noREMDKNA.csv

Type of file: Microsoft Excel Comma Separated Values File (.csv)

Opens with: Unknown application

Location: C:\Users\LIVES4BIT\Desktop

Size: 1.29 GB (1,385,724,536 bytes)

WVS_TimeSeries_4_0_D002bin_afterREMDKN...

General | Security | Details

/S_TimeSeries_4_0_D002bin_afterREMDKNA.csv

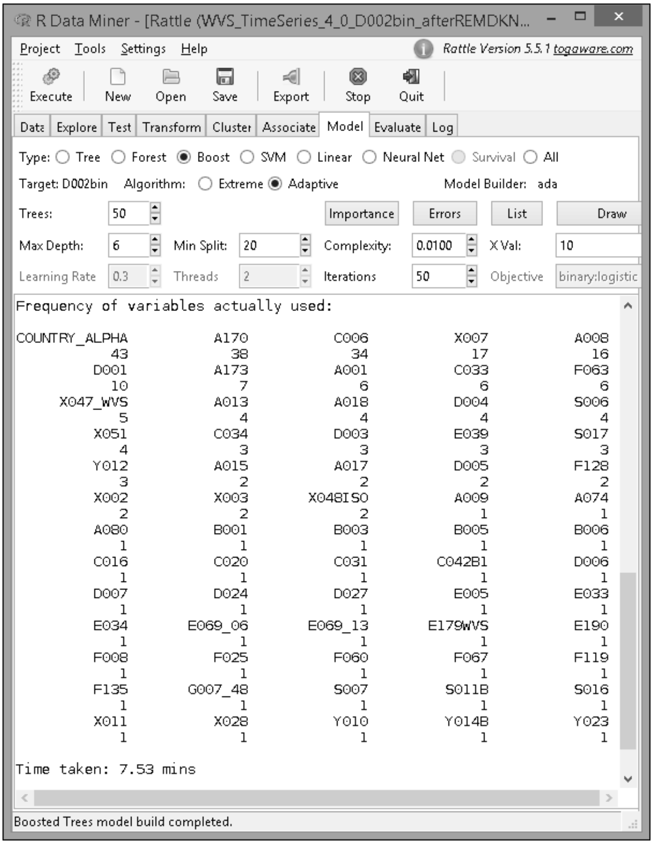
Type of file: Microsoft Excel Comma Separated Values File (.csv)

Opens with: Unknown application

Location: C:\Users\LIVES4BIT\Desktop

Size: 671 MB (704,099,109 bytes)

Figure 2. Comparative resulting .csv exports if previously using (right – smaller size) or NOT (left – larger size) the REMDKNA command



R Data Miner - [Rattle (WVS_TimeSeries_4_0_D002bin_afterREMDKN...

Rattle Version 5.5.1 togaware.com

Type: ○ Tree ○ Forest ● Boost ○ SVM ○ Linear ○ Neural Net ○ Survival ○ All

Target: D002bin Algorithm: ○ Extreme ● Adaptive Model Builder: ada

Trees: 50

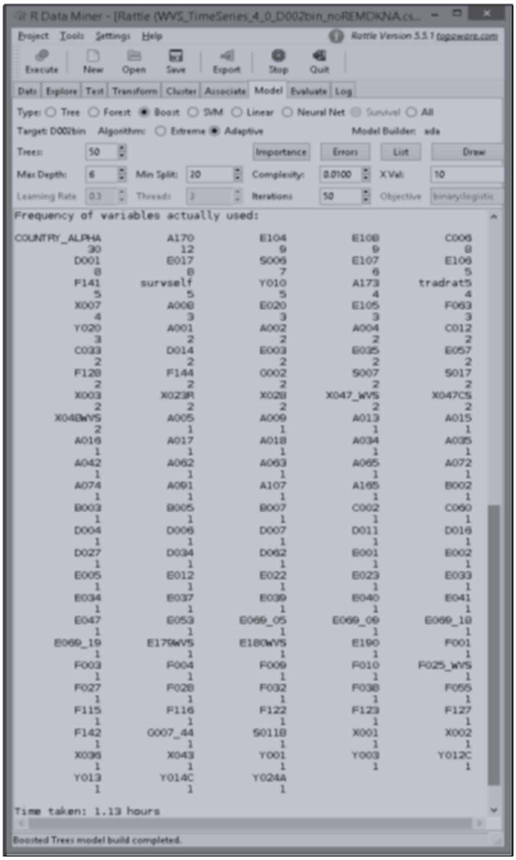
Max Depth: 6 Min Split: 20 Complexity: 0.0100 X Val: 10

Learning Rate 0.3 Threads 2 Iterations 50 Objective binary:logistic

```
Frequency of variables actually used:

COUNTRY_ALPHA          A170          C006          X007          A008
           43            38            34            17            16
         D001          A173          A001          C033          F063
           10             7             6             6             6
      X047_WVS          A013          A018          D004          S006
            5             4             4             4             4
         X051          C034          D003          E039          S017
            4             3             3             3             3
         Y012          A015          A017          D005          F128
            3             2             2             2             2
         X002          X003       X048ISO          A009          A074
            2             2             2             1             1
         A080          B001          B003          B005          B006
            1             1             1             1             1
         C016          C020          C031        C042B1          D006
            1             1             1             1             1
         D007          D024          D027          E005          E033
            1             1             1             1             1
         E034       E069_06       E069_13       E179WVS          E190
            1             1             1             1             1
         F008          F025          F060          F067          F119
            1             1             1             1             1
         F135       G007_48          S007         S011B          S016
            1             1             1             1             1
         X011          X028          Y010         Y014B          Y023
            1             1             1             1             1

Time taken: 7.53 mins
```

Boosted Trees model build completed.

Figure 3A. Results of applying the Ada Boost technique (in the Rattle library of R) if previously removing the DK/NA values using REMDKNA (less time required)



R Data Miner - [Rattle (WVS_TimeSeries_4_0_D002bin_noREMDKNA.cs...
Project Tools Settings Help
Rattle Version 5.5.1 togaware.com
Execute New Open Save Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Importance Errors List Draw

Type: ○ Tree ○ Forest ● Boost ○ SVM ○ Linear ○ Neural Net ○ Survival ○ All

Target: D002bin Algorithm: ○ Extreme ● Adaptive Model Builder: ada

Trees: 50

Max Depth: 6 Min Split: 20 Complexity: 0.0100 X Val: 10

Learning Rate 0.3 Threads 2 Iterations 50 Objective binary:logistic

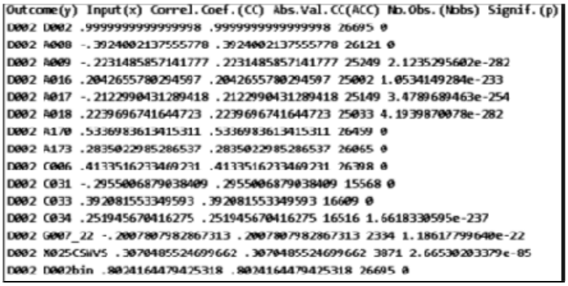
```
Frequency of variables actually used:

COUNTRY_ALPHA          A170          E104          E108          C006
           30            12             9             9             8
         D001          E017          S006          E107          E106
            8             8             7             6             5
         F141      survself          Y010          A173      tradrat5
            5             5             5             4             4
         X007          A008          E020          E105          F063
            4             3             3             3             3
         Y020          A001          A002          A004          C012
            3             2             2             2             2
         C033          D014          E003          E035          E057
            2             2             2             2             2
         F128          F144          G002          S007          S017
            2             2             2             2             2
         X003         X023R          X028      X047_WVS        X047CS
            2             2             2             2             2
      X048WVS          A005          A009          A013          A015
            2             1             1             1             1
         A016          A017          A018          A034          A035
            1             1             1             1             1
         A042          A062          A063          A065          A072
            1             1             1             1             1
         A074          A091          A107          A165          B002
            1             1             1             1             1
         B003          B005          B007          C002          C060
            1             1             1             1             1
         D004          D006          D007          D011          D016
            1             1             1             1             1
         D027          D034          D062          E001          E002
            1             1             1             1             1
         E005          E012          E022          E023          E033
            1             1             1             1             1
         E034          E037          E039          E040          E041
            1             1             1             1             1
         E047          E053       E069_05       E069_09       E069_18
            1             1             1             1             1
      E069_19       E179WVS       E180WVS          E190          F001
            1             1             1             1             1
         F003          F004          F009          F010      F025_WVS
            1             1             1             1             1
         F027          F028          F032          F038          F055
            1             1             1             1             1
         F115          F116          F122          F123          F127
            1             1             1             1             1
         F142       G007_44         S011B          X001          X002
            1             1             1             1             1
         X036          X043          Y001          Y003         Y012C
            1             1             1             1             1
         Y013         Y014C         Y024A
            1             1             1

Time taken: 1.13 hours
```

Boosted Trees model build completed.

Figure 3B. Results of applying the Ada Boost technique (in the Rattle library of R) if previously NOT removing the DK/NA values using REMDKNA (more time needed)

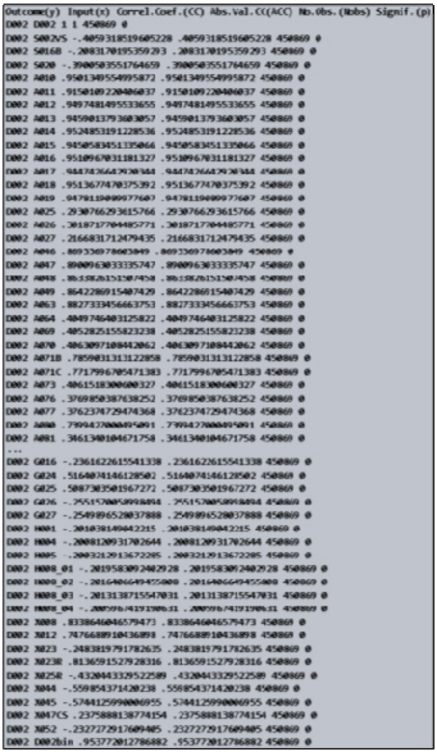
When using a selection based on the absolute values of the Pearson Pairwise Correlation Coefficients between the outcome, the scale form of D002 and each predictor, using PCDM, Figures 3A&3B, a minimum accepted magnitude of 0.2 (minacc=0.2) for such coefficients was considered, negligible correlation. The value of 0.2 is a reconciliation average (0 to 0.2 or −0.2 to 0 for Very Weak / Very Low / Negligible; 0.2 to 0.4 or −0.4 to −0.2 for Weak / Low; 0.4 to 0.6 or −0.6 to −0.4 for Moderate / Intermediate; 0.6 to 0.8 or −0.8 to −0.6 for High/Strong, and 0.8 to 1 or −1 to −0.8 for Very High/Very Strong Correlation) between the two versions: the first 0.1 (Schober *et al.,* 2018) and the second 0.3 (Mukaka, 2012).

Moreover, 0.2 represents a consistent step (0.2x5=1 or -0.2x5=-1) given those five categories: negligible, weak/low, moderate, strong, and very firm/strong correlation (those five intervals above). We can notice that the 3rd state (moderate/intermediate) is symmetric if considering the middle values of -0.5 and 0.5 for -1 to 0 and 0 to 1 as negative and positive subranges of the correlation coefficients. In addition, the minimum significance of one per thousand was considered in this selection scenario (maxp of 0.001 – Figures 4A and 4B).



```
Outcome(y) Input(x) Correl.Coef.(CC) Abs.Val.CC(ACC) No.Obs.(Nobs) Signif.(p)
D002 D002 .9999999999999998 .9999999999999998 26695 0
D002 A008 -.3924002137555778 .3924002137555778 26121 0
D002 A009 -.2231485857141777 .2231485857141777 25249 2.1235295602e-282
D002 A016 .2042655780294597 .2042655780294597 25002 1.0534149284e-233
D002 A017 -.2122990431289418 .2122990431289418 25149 3.4789689463e-254
D002 A018 .2239696741644723 .2239696741644723 25033 4.1939870078e-282
D002 A170 .5336983613415311 .5336983613415311 26459 0
D002 A173 .2835022985286537 .2835022985286537 26065 0
D002 C006 .4133516233469231 .4133516233469231 26398 0
D002 C031 -.2955006879038409 .2955006879038409 15568 0
D002 C033 .392081553349593 .392081553349593 16609 0
D002 C034 .251945670416275 .251945670416275 16516 1.6618330595e-237
D002 G007_22 -.2007807982867313 .2007807982867313 2334 1.18617799640e-22
D002 X025CSWVS .3070485524699662 .3070485524699662 3871 2.66530203379e-85
D002 D002bin .8024164479425318 .8024164479425318 26695 0
```

Figure 4A. Results of applying filters based on Pearson Pairwise Correlation Coefficients if previously removing the DK/NA values using REMDKNA



```
Outcome(y) Input(x) Correl.Coef.(CC) Abs.Val.CC(ACC) No.Obs.(Nobs) Signif.(p)
D002 D002 1 1 450869 0
D002 S002VS -.4059318519605228 .4059318519605228 450869 0
D002 S016B -.2083170195359293 .2083170195359293 450869 0
D002 S020 -.3900503551764659 .3900503551764659 450869 0
D002 A010 .9501349554995872 .9501349554995872 450869 0
D002 A011 .9150109220406037 .9150109220406037 450869 0
D002 A012 .9497481495533655 .9497481495533655 450869 0
D002 A013 .9459013793603057 .9459013793603057 450869 0
D002 A014 .9524853191228536 .9524853191228536 450869 0
D002 A015 .9450583451335066 .9450583451335066 450869 0
D002 A016 .9510967031181327 .9510967031181327 450869 0
D002 A017 .9447426647920344 .9447426647920344 450869 0
D002 A018 .9513677470375392 .9513677470375392 450869 0
D002 A019 .9478119000977607 .9478119000977607 450869 0
D002 A025 .2930766293615766 .2930766293615766 450869 0
D002 A026 .3018717704485771 .3018717704485771 450869 0
D002 A027 .2166831712479435 .2166831712479435 450869 0
D002 A046 .8693369786003049 .8693369786003049 450869 0
D002 A047 .8900963033335747 .8900963033335747 450869 0
D002 A048 .8633826151507458 .8633826151507458 450869 0
D002 A049 .8642286915407429 .8642286915407429 450869 0
D002 A063 .8827333456663753 .8827333456663753 450869 0
D002 A064 .4049746403125822 .4049746403125822 450869 0
D002 A069 .4052825155823238 .4052825155823238 450869 0
D002 A070 .4063097108442062 .4063097108442062 450869 0
D002 A071B .7859031313122858 .7859031313122858 450869 0
D002 A071C .7717996705471383 .7717996705471383 450869 0
D002 A073 .4061518300600327 .4061518300600327 450869 0
D002 A076 .3769850387638252 .3769850387638252 450869 0
D002 A077 .3762374729474368 .3762374729474368 450869 0
D002 A080 .7999427000095091 .7999427000095091 450869 0
D002 A081 .3461340104671758 .3461340104671758 450869 0
...
D002 G016 -.2361622615541338 .2361622615541338 450869 0
D002 G024 .5164074146128502 .5164074146128502 450869 0
D002 G025 .5087303501967272 .5087303501967272 450869 0
D002 G026 -.2551570050998494 .2551570050998494 450869 0
D002 G027 -.2549896528037888 .2549896528037888 450869 0
D002 H001 -.2010381490042215 .2010381490042215 450869 0
D002 H004 -.2008120931702644 .2008120931702644 450869 0
D002 H005 -.2003212913672285 .2003212913672285 450869 0
D002 H008_01 -.2019583092402928 .2019583092402928 450869 0
D002 H008_02 -.2016406649455000 .2016406649455000 450869 0
D002 H008_03 -.2013138715547031 .2013138715547031 450869 0
D002 H008_04 -.2005967419190631 .2005967419190631 450869 0
D002 X008 .8338646046579473 .8338646046579473 450869 0
D002 X012 .7476688910436898 .7476688910436898 450869 0
D002 X023 -.2483819791782635 .2483819791782635 450869 0
D002 X023R .8136591527928316 .8136591527928316 450869 0
D002 X025R -.4320443329522589 .4320443329522589 450869 0
D002 X044 -.559854371420238 .559854371420238 450869 0
D002 X045 -.5744125990006955 .5744125990006955 450869 0
D002 X047CS .2375888138774154 .2375888138774154 450869 0
D002 X052 -.2327272917609405 .2327272917609405 450869 0
D002 D002bin .9537720127806882 .9537720127806882 450869 0
```

Figure 4B. Results of applying filters based on Pearson Pairwise Correlation Coefficients if previously NOT removing the DK/NA values using REMDKNA

An extra argument for a balanced and conciliatory approach for Pearson Pairwise Correlation Coefficients, those five intervals above, is the existing balanced approach of interpreting AUC-ROC classification accuracy values, five equidistant intervals of step 0.1 between 0.5 and 1 corresponding (Nahm, 2022) to Fail, Poor, Fair, Good, Excellent or, in other approaches: Fail, Worthless, Poor, Good, Excellent (Polo and Miot, 2020) or Bad, Poor, Satisfactory, Good, Excellent (Bogale *et al.,* 2022), or Unsatisfactory, Satisfactory, Poor, Good, Excellent (Trifonova *et al.,* 2014). Filters based on Pearson Pairwise Correlation Coefficients selected only a few predictors, 13 items excluding the target: D002 &D002bin, when using REMDKNA, Figure 4A. A less consistent filtering, 335 items out of 1045) occurred in the other case, Figure 4B.

Next, the intersection between the results provided by Ada Boost and those returned by the approach considering Pearson correlation coefficients, Pearson Pairwise Correlation Coefficients using PCDM in Stata, represents ten vs. 52 predictors depending on previously dropping the DK/NA values, right side of Table 1, or not - left of Table 1.

After performing both CVLASSO and RLASSO selections on both forms of the target variables, 29 of 52 variables resulted (left of Table 2) if not removing the DK/NA values and just eight of 10 (right of Table 2) if previously using REMDKNA and dropping these values.

Table 1. Comparative intersecting results of applying both the Ada Boost technique in Rattle and filters based on Pearson Pairwise Correlation Coefficients (PCDM in Stata)

Intersection between Ada Boost and Pearson Pairwise Correlation Coefficients if NOT using REMDKNA						Intersection between Ada Boost and Pearson Pairwise Correlation Coefficients if using REMDKNA
A013	B007	D016	E104	F027	X023R	A008
A015	C012	D027	E105	F032	X047CS	A009
A016	C033	D034	E106	F055		A017
A017	C060	D062	E107	F127		A018
A018	D001	E017	E108	F128		A170
A063	D004	E020	E190	F141		A173
A091	D006	E047	F003	F142		C006
A107	D007	E053	F004	F144		C031
B003	D011	E057	F009	G002		C033
B005	D014	E069_09	F010	G007_44		C034

Table 2. Comparative results of applying both CVLASSO&RLASSO (Stata) in cascade

CVLASSO and RLASSO if NOT using REMDKNA			CVLASSO and RLASSO if using REMDKNA
A016	E020	F004	A008
A017	E047	F009	A017
A018	E053	F027	A170
B007	E057	F127	A173
C033	E104	F128	C006
D001	E105	F142	C031
D004	E106	F144	C033
D016	E107	G007_44	C034
D034	E108	X047CS	
E017	E190		

The step that followed was to check for reverse causality issues using ordered logit and ordered probit regressions and each variable selected in the previous stage as input and the target in the scale form (D002) and vice versa by interchanging the roles (D002 as input and each of those eight predictors on the right side of Table 2 as output). In addition, some model performance metrics such as Pseudo R-squared, AIC, and BIC automatically resulted (computed and reported) to conclude which variable of those eight is more appropriate to act as input and which as a target in connection with D002. This selection stage only took place in the case of previously dropping the DKNA values, and it filtered the previous set of possible predictors and preserved only five, namely A170, A173, C006, C033, and C034. The corresponding results are available in Tables A3 and A4.

The next step was to remove the collinearity issues in the second scenario (if cleaning the dataset using the REMDKNA command). After identifying the collinear pairs, a prediction nomogram (Figure 5) resulted after performing a logit regression, including all five remaining predictors. The removal considered only those variables involved in collinear pairs ((A170, C033), (C006, C033), and (C033, C034) – top of Table 3, together with (A170, A173), and (A170, C006) - right of Figure 6). It also dropped the ones with lower values for the magnitude (smaller overall bars and the right edge corresponding to a lower score – Figure 5). The first three collinear pairs (top of Table 3) resulted when considering all possible OLS regressions with D002 or D002bin set as outcomes (y=D002bin or y=D002) and each pair of those five remaining predictors above (after performing reverse causality checks), and observing that R^2 in OLSx1x2 > R^2 in OLSyx1x2 or Maximum Computed VIF (using the VIF command after OLSyx1x2) > Maximum Accepted VIF=1/(1-R^2) for OLSyx1x2 (bottom of Table 3). Moreover, an *OLS Maximum Computed VIF>OLS Maximum Accepted*

VIF means that the correlation between the predictors is stronger than the regression relationship (Vatcheva *et al.,* 2016), and multicollinearity can affect their coefficient estimates.

Moreover, two matrices with Pearson Pairwise Correlation Coefficients (Figure 6 - only for those remaining five predictors above) served to identify additional collinearity issues. In this second approach, a maximum accepted magnitude of 0.4 (Schober et al., 2018) for such correlation coefficients was considered (negligible and weak correlation), and two other collinear pairs have been identified (A170 and A173, A170, and C006). The latter finding regarding these two additional collinear pairs couples with the fact that using a similar approach as the one depicted in this article (including DK/NA treatments using REMDKNA and similar selection steps), A008, A009, A173, C006 emerged as the most resilient determinants of life satisfaction (A170). Therefore, A173 and C006 cannot co-exist with A170 in the model in Figure 5 (D002bin set as target). The first two are also predictors of A170 (a clear indication of redundancy affecting A170). After removing all collinearity issues based on the logic above, the remaining list of predictors comprises only three (A173, C006, and C034) from those five above, and, eventually, A008 and A009 (the predictors of A170). Still, the latter two did not pass reverse causality checks (like those described in Tables A3 &A4) when considering D002 (homelife satisfaction) as the target. Consequently, they are not confirmed when all previous selection steps are activated.

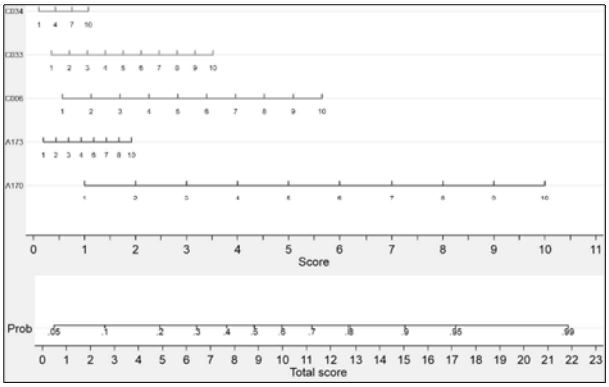
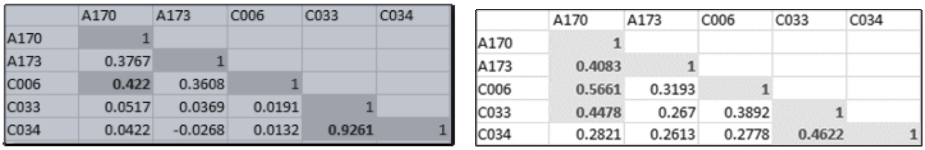


Figure 5. Nomogram (the *nomolog* tool in Stata) for collinearity removal purposes

Moreover, a comparison between C033 and C006 took place. C033 was dropped (smaller overall bar and a lower score corresponding to the value on the right edge – Figure 5, and, in addition, a tinier frequency in the results of Ada Boost than the one of C033: 6 vs. 34 – top of Figure 2). The same applies to the pair C033 and A170. This removal solved most critical collinearity issues, as shown at the top of Table 3. Additionally, it excluded half of the ones in Figure 6 (right side).



	A170	A173	C006	C033	C034
A170	1				
A173	0.3767	1			
C006	0.422	0.3608	1		
C033	0.0517	0.0369	0.0191	1	
C034	0.0422	-0.0268	0.0132	0.9261	1

	A170	A173	C006	C033	C034
A170	1				
A173	0.4083	1			
C006	0.5661	0.3193	1		
C033	0.4478	0.267	0.3892	1	
C034	0.2821	0.2613	0.2778	0.4622	1

Figure 6. Comparative matrices with Pearson Pairwise Correlation Coefficient

It is also worth mentioning the differences in correlations among the reliable predictors identified (if previously removing DK/NA values or not – Figure 6) and those in VIF and R^2 (top of Table 3 vs. bottom of Table 3), which do not point out the same collinearity issues.

Table 3. Collinearity checks for each pair of those five predictors resisting reverse causality checks if previously removing DK/NA values (top) or NOT (bottom)

y	x1	x2	R^2 for OLSx1x2	R^2 for OLS yx1x2	Max.Comput.VIF for OLS yx1x2 (estat vif)	Max.Accept.VIF for OLS yx1x2 (=1/(1-R^2))
D002bin	A170	C033	0.1999	0.1959	1.2499	1.2436
D002bin	C006	C033	0.1517	0.1486	1.1789	1.1745
D002bin	C033	C034	0.2121	0.0979	1.2691	1.1085
D002	C033	C034	0.2121	0.1585	1.2691	1.1883
D002bin	A170	A173	0.1419	0.0056	1.1654	1.0057
D002bin	A170	C006	0.1781	0.0071	1.2167	1.0071
D002bin	A173	C006	0.1302	0.0047	1.1496	1.0047
D002bin	C033	C034	0.8576	0.5475	7.0244	2.2097
D002	A170	A173	0.1419	0.0036	1.1654	1.0036
D002	A170	C006	0.1781	0.0046	1.2166	1.0046
D002	A173	C006	0.1302	0.0032	1.1496	1.0032
D002	C033	C034	0.8576	0.6195	7.0244	2.6281

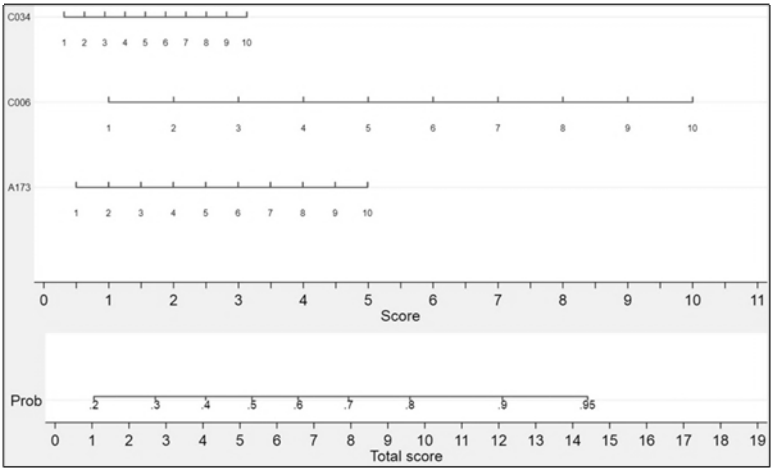


Figure 7A. Prediction nomogram if previously removing the DK/NA values

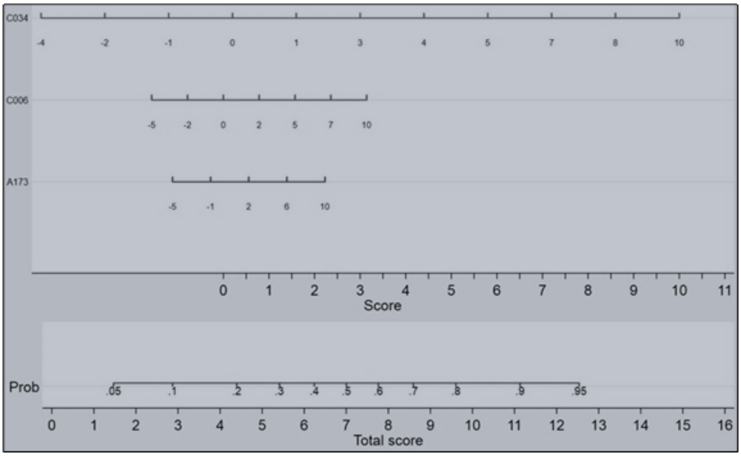


Figure 7B. Prediction nomogram if previously NOT removing the DK/NA values

The next step was to compare different regression models and prediction nomograms containing these three remaining predictors also resisting collinearity checks (A173, C006, and C034) and considering previous treatment of DK/NA values or not.

By performing side-by-side comparisons between those six models in tables A5, when cleaning DK/NA values by using the REMDKNA command, and A6, when not performing such data cleaning tasks, we can see that all R-squared values for all six models are exaggerated (much higher) in the second case (Table A6). The same applies to the accuracy of classification (AUC-ROC of

0.7832 vs. 0.9182 for model 1-Logit and 0.7832 vs. 0.9374 for model 2-Probit). The RMSE for OLS regression models (3 and 6) was also higher when previously applying REMDKNA. Instead, the AIC and BIC information criteria, an indication of the goodness of fit, recorded better/smaller values for all six models when previously performing data cleaning tasks (Table A5). The maximum absolute values of the Pearson correlation coefficients between all pairs of predictors (maxAbsVPMCC of 0.2835 vs. 0.3608) indicate higher values (slightly more collinearity) when not performing the removal of DK/NA values (Table A6). All these couples with the overrated number of valid observations (16160-Table A5 vs. 450869-Table A6). Moreover, they are specific to the dataset used, the outcome variable, and the most robust corresponding predictors identified. Still, they offer a clear picture of the magnitude of distortions obtained if dealing with the source data, but not adequately.

It is easily noticeable that when allowing for inflated scales, the entire selection process (including all checks) dramatically complicates (Figure 3A vs. Figure 3B, Figure 4A vs. Figure 4B, Tables 2, 3 and 4A vs. 4B). And that comes second if considering the lack of correctness of an approach allowing for such not treated messy data. In addition, if we focus on the most robust predictors, the noticeable differences in their compared magnitudes (an overturning of the ranking – Figure 7A vs. Figure 7B) when considering these two scenarios (previously cleaning the original data vs. not) should raise tough question marks. Moreover, the significant differences in the performance metrics (support, accuracy, R-squared, fit, etc.) of the resulting models in these two scenarios are another solid argument for using such automatic tools dedicated to the correct data treatment.

As already demonstrated, REMDKNA successfully simplifies the task of automatically cleaning some dataset types (e.g., large files designed by considerable project owners such as WVS or SHARE, including the .dta format for Stata) before starting to perform feature selection tasks and also contributes to obtaining realistic models.

4. CONCLUSIONS

This article describes the effects of using an automated solution for removing DK/NA values encoded as negative ones, in some large survey datasets. First, it is about minimizing the manual intervention required for data cleaning. Second, the solution demonstrates the effectiveness of improving the feature selection and robust model exploration steps, including the tests of predictor independence, and obtaining realistic model evaluation metrics. And these stand on real-world datasets analysis such as the World Values Survey or similar. By removing the artificially inflated scales caused by DK/NA values, the integrity and reliability of statistical models are ensured. Moreover, the real-time progress monitoring and reporting feature of such a solution, including easy tracking of DK/NA value treatment and insights into the success in case of numerical variables or exceptions encountered for string ones, enhances its usability and efficiency.

References

1) Acuña, E. and Rodriguez, C. (2004). The Treatment of missing values and its effect on classifier accuracy. In: Banks, D., McMorris. F.R., Arabie. P., Gaul, W., (eds.), *Classification, Clustering, and data mining applications. Studies in classification, Data Analysis, and Knowledge Organisation*, Berlin: Springer, Berlin, pp. 639-647.
2) Ahrens, A., Hansen, C.B. and Schaffer, M.E. (2020). Lassopack: Model selection and prediction with regularized regression in Stata. *The Stata Journal: Promoting communications on statistics and Stata*, 20, pp. 176-235.
3) Assale, M., Dui, L.G., Cina, A., Seveso A. and Cabitza, F. (2019). The Revival of the Notes Field: Leveraging the Unstructured Content in Electronic Health Records. *Front. Med.*, 6, pp. 66.
4) Bogale, L., Anley, D.T., Tsegaye, T., Abdulkadir, M. and Akalu, T.Y. (2022). A score to predict the risk of major adverse drug reactions among Multi-Drug-Resistant tuberculosis patients in Southern Ethiopia, 2014–2019. *Infection and Drug Resistance,* 15, pp. 2055-2065.
5) Couper, M.P., Traugott, M.W. and Lamias, M.J. (2001). Web Survey Design and Administration. *Public Opinion Quarterly,* 65(2), pp. 230-253.
6) Farhangfar, A., Kurgan, L.A. and Pedrycz, W. (2007). A Novel Framework for Imputation of Missing Values in Databases. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans,* 37(5), pp. 692-709.
7) Homocianu, D. and Airinei, D. (2022). PCDM and PCDM4MP: New Pairwise Correlation-Based Data Mining Tools for Parallel Processing of Large Tabular Datasets. *Mathematics,* 10, pp. 2671.
8) Homocianu, D. and Tîrnăucă, C. (2022). MEM and MEM4PP: New Tools Supporting the Parallel Generation of Critical Metrics in the Evaluation of Statistical Models. *Axioms,* 11, pp. 549.
9) Jann, B. (2005). Making regression tables from stored estimates. *The Stata Journal* 5(3), pp. 288-308.
10) Jann, B. and Long, J.S. (2010). Tabulating SPost results using estout and esttab. *The Stata J.,* 10(1), pp. 46-60.
11) Jiménez-Valverde, A. (2012). Insights into the area under the receiver operating characteristic curve (AUC) as a discrimination measure in species distribution modelling. *Glob. Ecol. & Biogeo*., 21, pp. 498-507.
12) Liu, B., Zhou, X., Wang, Y., Hu, J., He, L., Zhang, R., Chen, S. and Guo, Y. (2012). Data processing and analysis in real-world traditional Chinese medicine clinical data: challenges and approaches*, Statistics in Medicine,* 31(7), pp. 653-660.
13) Liveris, A., Bello, R.A., Friedmann, P., Duffy, M.A., Manwani, D., Killinger, J.S., Rodriquez, D. and Weinstein, S. (2014). Anti-Factor Xa Assay Is a Superior Correlate of Heparin Dose Than Activated Partial Thromboplastin Time or

Activated Clotting Time in Pediatric Extracorporeal Membrane Oxygenation. *Pediatric Critical Care Medicine,* 15(2), pp. 72-79.

14) Mukaka, M.M. (2012). A guide to appropriate use of Correlation coefficient in medical research. *Malawi Med. J.*, 24(3), pp. 69-71.

15) Nahm, F.S. (2022). Receiver operating characteristic curve: overview and practical use for clinicians. *Korean Journal of Anesthesiology*, 75(1), pp. 25-36.

16) Polo, T.C.F. and Miot, H.A. (2020). Aplicações da curva ROC em estudos clínicos e experimentais. *Jornal Vascular Brasileiro*, 19, pp. 1-4.

17) Sanchez, J.D., Rêgo, L.C. and Ospina, R. (2019). Prediction by Empirical Similarity via Categorical Regressors. *Mach. Learn. Knowl. Extr.,* 1, pp. 641-652.

18) Schober, P., Boer, C. and Schwarte, L.A. (2018). Correlation coefficients: appropriate use and interpretation. *Anesth. & Analges.,* 126(5), pp. 1763-1768.

19) Schonlau, M. (2005). Boosted regression (boosting): An introductory tutorial and a Stata plugin. *The Stata J.,* 5(3), pp. 330-354.

20) Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *J. of Roy. Stat. Society, ser. B (meth.),* 58(1), pp. 267-288.

21) Trifonova, O P., Lokhov, P.G. and Archakov, A.I. (2014). Metabolic profiling of human blood. *Biomeditsinskaia Khimiia*, 60(3), pp. 281-294.

22) Vadivukkarasi, S. and Santhi, S. (2020). A novel hybrid learning based Ada Boost (HLBAB) classifier for channel state estimation in cognitive networks. *International Journal of Dynamics and Control*, 9(1), pp. 299-307.

23) Vatcheva, K.P., Lee, M.J., McCormick, J.B. and Rahbar, M.H. (2016). Multicollinearity in Regression Analyses Conducted in Epidemiologic Studies. *Epidem. (Sunnyvale),* 6(2), pp. 227.

24) Williams, P., Alessa, L., Abatzoglou, J.T. et al. (2018). Community-based observing networks and systems in the Arctic: Human perceptions of environmental change and instrument-derived data. *Reg Environ Change*, 18, pp. 547-559.

25) Young, W., Weckman, G. and Holland, W. (2011). A survey of methodologies for the treatment of missing values within datasets: limitations and benefits, *Theoretical Issues in Ergonomics Science,* 12(1), pp. 15-43.

26) Zhang, S. (2011). Information enhancement for data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(4), pp. 284-295.

27) Zlotnik, A. and Abraira, V. (2015). A general-purpose nomogram generator for predictive logistic regression models. *The Stata J.,* 15(2), pp. 537-546.