Software Implementation of Application for Correlation Analysis

Vitalii Bazurin, Oleg Pursky, Ihor Tyschenko, Andrey Nechepourenko and Alexander Vishnevsky
Department of Computer Science and Information Systems, State University of Trade and Economics, Kyoto Str. 19, 02156 Kyiv, Ukraine
v.bazurin@knute.edu.ua, o.pursky@knute.edu.ua, i.tyshchenko@knute.edu.ua, a.nechepurenko@knute.edu.ua, o.vyshnivskyy@knute.edu.ua

Keywords: Correlation, Statistics, Software Library, Pearson Correlation, Spearman Correlation, Kendall Correlation, Algorithm, Application.

Abstract: The article contains the results of research on the development of an application for determining the correlation between two data samples. Such an application is relevant for correlation analysis. The central component of the applications the Correlation library, which contains classes and methods for calculating the Pearson, Spearman, and Kendall correlation coefficients. The library is implemented in Python. The created Correlation library utilizes built-in Python data types and does not rely on third-party libraries. The article describes the algorithms used to calculate these coefficients, as well as the Python implementations for calculating the specified correlation coefficients. For the practical implementation of the library, the PyCorrelation application has been developed in two versions. The interface of the first version of the application is implemented using the PyQT 5 library; the interface of the second version is implemented using the free Custom Tkinter library. The results of calculating the Pearson, Spearman, and Kendall correlation coefficients are verified using MS Excel spreadsheets. The developed Correlation library has significant prospects for modernization and use as a component of software tools for statistical processing of experimental data. It can also be integrated into future versions of the StatCriterion software tool, previously developed by the authors of the article.

1 INTRODUCTION

In the process of conducting many studies, including pedagogical ones, there is a need to establish the existence of a relationship between two phenomena. Quantitative characteristics of such phenomena are usually expressed in numerical arrays. For example, the relationship between voltage usage and current fluctuations, algebra grades and chemistry grades, algebra test results and physics test results, etc. To determine such relationships, there are methods of correlation analysis: Pearson, Spearman, and Kendall. According to R. Taylor, correlation analysis is of great importance in scientific research. All these methods differ in algorithm, but provide high reliability of the calculated results. The only drawback of these methods, in our opinion, is that these calculations are cumbersome, and it takes a lot of time to perform them manually. That is why it is advisable to develop a software tool that will automate correlation calculations and draw conclusions from the calculations performed. In the case of a component-oriented architecture of such a software tool, one of its main components will be a library of functions for calculating correlation coefficients. Such libraries for the Python language already exist: NumPy and Pandas. Using these libraries, you can create a program to determine the existence of a relationship between two data sets. However, here too, problems may arise in the further use of such a program. There is a possibility, albeit insignificant, that the release of the NumPy and Pandas libraries for new versions of Python may be discontinued. That is why, to develop and support software, it is advisable to first develop your own modules in the appropriate programming languages. The purpose of the article is to reveal the features of building algorithms and software code for a Python library for calculating Pearson, Spearman, and Kendall correlation coefficients.

2 LITERATURE REVIEW

The problem of applying correlation in the practice of scientific research has two aspects. The first is to develop correlation methods that will unambiguously show the presence or absence of a relationship between two data sets. R. Taylor reveals the role of correlation analysis in scientific research, the features of calculating Pearson's correlation [1]. The Pearson correlation coefficient varies within [-1;1] and characterizes the presence or absence of a relationship between two data sets. B. Ratner, in his article, presents an algorithm and basic formulas for calculating the Pearson correlation coefficient [2]. The article provides examples of calculating the Pearson correlation coefficient in the form of tables. D. Chicco and G. Jurman in their article [3] reveal the meaning of the Matthews correlation coefficient and give examples of using this coefficient to determine whether there is a relationship between two sets of features (even of different sizes). The Matthews coefficient takes into account true positives and negatives, false positives and false negatives. J.Jiang, X.Zhang, and Zhong Yuan in their article demonstrate an example of the application of Spearman correlation in the field of machine learning [4]. S.Chatterjee in his article states that the most popular classical correlation coefficients are Pearson, Spearman and Kendall coefficients [5]. S.Chatterjee developed his own correlation coefficient, which was called the Chatterjee coefficient. X.Su, S.Shang, Z.Xu, H.Quian and X.Pan conducted a study of risk factors using Pearson correlation [6]. In particular, they used Pearson correlation to determine the type and strength of the relationship between two factors of productivity formation. H.Yu and A.D.Hutson in their study check the validity of Spearman correlation for different samples [7]. Scientists prove that the test they developed to check the validity of Spearman correlation is valid for small and large data sets. F.Han and Z.Huang adapted Azadkia-Chatterjee's correlation coefficient for different data [8]. In their article, scientists prove the theorem and 11 lemmas. C.A.Metler, R.A.Vannatta and K.N.LaVenia in their book reveal the features of algorithms for calculating univariate and multivariate correlation [9].

The second direction of research is to create software tools that can automate the process of calculating the correlation coefficient and formulating a conclusion about the presence or absence of a relationship between two data sets. For example, V. Mikhalchuk in his study gives examples of creating pseudocode for calculating the t-test and p-value for two samples [10]. But to calculate the p-value, V. Mikhalchuk uses the function of the corresponding statistical library. H. Kang in his study uses the G* Power software tool to perform statistical data processing, including calculating the Pearson correlation coefficient [11]. Using the Pearson correlation, the presence or absence of a relationship between two samples is determined. The author recommends the G* Power software tool for performing statistical calculations. R.Bivand, G.Millo and G.Piras analyze the software tools for statistical data processing available in the R system [12]. Scientists have developed appropriate tests for evaluating statistical criteria, including correlation coefficients. L.Qi, W.Lin, X.Zhang, W.Dou, X.Xu and J.Chen investigate the issue of using the correlation graph in order to select and recommend web APIs for mobile applications [13]. S.Mguil-Touchal, F.Morestin and M.Brunei in their article demonstrate examples of creating experimental software for calculating correlation [14]. The article provides examples of using software tools for calculating the correlation coefficient in the process of studying the mechanical properties of solids. S.D.Rahmawati, A.F. Ihsan, F.Priadi and U.W.Siagian in their article [15] show the results of developing a software tool for studying the physical properties of hydrocarbon liquids. V.M. Bazurin, O.I. Pursky and O.S. Chashechnikova in their article demonstrate the structure and interface of an application for statistical processing of experimental results [16]. The developed application compares two samples of values using the Pearson, Whitney-Mann, and Student criteria. It would be advisable to supplement this application with correlation functions.

The analysis of research on the creation, justification of correlation coefficients and their use in practice shows that the problem of developing correlation methods is relevant and continues to be studied by many scientists.

3 OBJECTIVES AND METHODS

(RQ1) How to build algorithms for calculating Pearson, Spearman, and Kendall correlation coefficients and write the code as a software library in Python?

(RQ2) How to integrate the developed library into a Python application to provide a convenient interface and the necessary functionality?

Problem: How to build algorithms for calculating Pearson, Spearman, and Kendall correlation

coefficients and present them as a software library integrated into the application?

In the process of research, we used the following methods:

- general scientific analytical method - for analyzing the state of the problem, analyzing literary sources;
- modeling method - for developing a model for calculating correlation coefficients;
- algorithmic method - for developing algorithms for calculating correlation coefficients;
- testing method.

4 RESULTS

When starting to create a software library, we first define the functional and non-functional requirements for this library:

- the software library must be autonomous and operate without references to third-party modules (for example, NumPY);
- the software library must use an object-oriented programming paradigm. This will provide a clearer structure of the library and simplify further modernization of the library;
- the software library must use built-in Python data types and not use data types from third-party libraries. This will facilitate the autonomous operation of the library regardless of access to third-party modules;
- the software library must contain error prevention tools;
- the software library methods must return the value of the selected correlation coefficient and a conclusion about the similarity of two data sets;
- the software library must be quickly and easily integrated into the application.

The software library can be developed in different programming languages. We consider it advisable to write a software library in the Python language. Python language tools provide:

- the use of simple and complex data types when calculating correlation coefficients;
- ease of implementation of basic algorithmic structures;
- various platforms for the functioning of applications developed in Python;
- comparative ease of generating a report on correlation analysis.

Pearson's correlation coefficient is used to determine the linear relationship between two data sets. The value of the Pearson's correlation coefficient is within [-1;1]. The example of Pearson correlation calculation is shown in Table 1.

Table 1: Calculation of Pearson correlation coefficient.

Table	x	y	x_i^2	y_i^2	$x_i y_i$
1	3	5	9	25	15
2	4	2	16	4	8
3	5	8	25	64	40
4	7	9	49	81	63
5	2	4	4	16	8
6	6	9	36	81	54
7	4	8	16	64	32
Summ	31	45	155	335	220

The value of the Pearson's correlation coefficient is calculated by (1). The Pearson correlation coefficient is calculated using the formula [17]:

$$r_{xy} = \frac{n\cdot\sum_i(x_i\cdot y_i)-(\sum_i x_i)\cdot(\sum_i y_i)}{\sqrt{(n\cdot\sum_i x_i^2-(\sum_i x_i)^2)\cdot(n\cdot\sum_i y_i^2-(\sum_i y_i)^2)}}. \quad (1)$$

For two data sets (Table 1), the Pearson correlation coefficient is 0.728. This indicates the existence of a direct strong relationship between the samples. The algorithm for calculating the Pearson correlation coefficient is shown in the diagram (Fig.1).

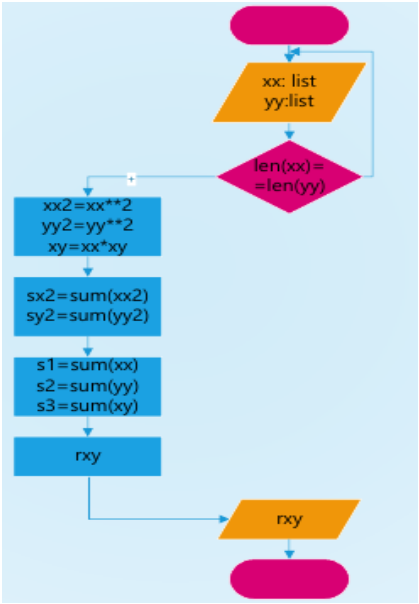

Figure 1: Schematic of the algorithm for calculating the Pearson correlation coefficient.

The algorithm for calculating the Pearson correlation coefficient is implemented as a Pearson

class. This class has the following fields: N - size of each array; xx - first list of values; uu - second list of values; xy - list of pairwise products of the elements of the first list by the elements of the second list; xx2 - list of squares of the numbers of the first list; yy2 - list of squares of the numbers of the second list; s1 - sum of the elements of the first list; s2 - sum of the elements of the second list; s3 - sum of the elements of the list xy; sx2 - sum of the squares of the elements of the first list; sy2 - sum of the squares of the elements of the second list, correl - Pearson correlation coefficient.

Methods of the Pearson class: init() - constructor; correlation - returns the Pearson correlation coefficient; concl() - returns a conclusion about the degree of connection between the initial lists of data. The generated PDF report contains the value of the correlation coefficient, a conclusion about the nature of the relationship between the samples, and a plot of the distribution of the samples being compared. The results of the Pearson class testing revealed that the algorithm was O(n) in complexity. The processing time for one pair of values did not change.

Spearman's correlation coefficient is used to determine the relationship in cases where it is not possible to use the Pearson correlation coefficient. Ranks are used when calculating the Spearman correlation coefficient. The advantage of the Spearman coefficient is that it is more sensitive to the relationship. The limitation of the Spearman coefficient is that each sample must contain at least 5 values. Algorithm for calculating the Spearman correlation:

1) record both samples in the table (Table 2);
2) assigning a rank to each sample value (relative to the elements of this sample);
3) calculating the difference in ranks for each pair of values;
4) calculating the square of each difference in ranks;
5) calculating the sum of the ranks of each sample and the sum of the squares of the differences;
6) calculating the Spearman correlation coefficient according to formula [18]:

$$r_s = 1 - \frac{6 \cdot \sum (R(x)_i - R(y)_i)^2}{n \cdot (n^2 - 1)}. \quad (2)$$

Based on the data in Table 2, the value rs = 0.8214 was obtained. The algorithm for the Spearman correlation coefficient calculating is shown in Figure 2.

To perform the calculations, the corrValue (numerical value) and Spearman (calculation of the correlation coefficient itself and drawing a conclusion) classes were designed and implemented.

The corrValue class describes the numerical value used in the calculations and has the following fields: value - numerical value, rank - rank, num - ordinal number of the value in the list. Class methods: init - constructor, setRank() - sets the rank, getRank() - returns the rank, setValue() - sets the value, getValue() - returns the value, setNum() - sets the ordinal number, getNum() - returns the ordinal number.

Table 2: Calculation of spearman correlation coefficient.

Num	*x*	*y*	*R(x)*	***R(y)***	$(R(x)-R(y))^2$
1	3	5	2	3	1
2	4	2	3	1	4
3	5	8	5	4	1
4	7	9	7	6	1
5	2	4	1	2	1
6	6	9	6	7	1
7	4	8	3	4	1
Sum	31	45	27	27	10

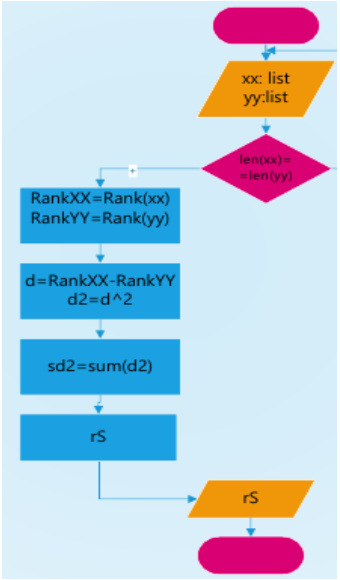

Figure 2: Schematic of the algorithm for calculating Spearman's correlation.

The Spearman class performs calculations and draws a conclusion about the existence and nature of the relationship. Class fields: xx - first sample, uu - second sample, N - number of values in each sample, RankXX - list of objects of class corrValue (first sample), RankYY - list of objects of class corrValue (second sample), RankXXX - list of objects of class corrValue (first sample, used to find rank), RankYYY - list of objects of class corrValue (second sample, used to find rank), correl - value of correlation

coefficient, Conclusion - conclusion about existence of connection.

The Spearman class has the following methods: init() - constructor, correlation () - calculation and return of Spearman's correlation coefficient, concl() - formulation and return of conclusion about existence of connection between samples and nature of this connection. The generated PDF report contains the value of the correlation coefficient, a conclusion about the nature of the relationship between the samples, and a plot of the distribution of the samples being compared. For the Spearman class, performance testing results revealed an algorithm complexity of $O(n^2)$. The time to process one pair of values increased proportionally to the sample size.

The τ-Kendall coefficient is used as an alternative to determine the relationship in the same cases as Spearman's rank correlation coefficient, but has a different calculation algorithm. Let's consider this algorithm in more detail. Let there be two samples of the same size. These values are written in the table (Table 3).

Table 3: Initial data for calculation the Kendall correlation coefficient.

Num	*x*	*y*
1	7	5
2	1	4
3	9	1
4	3	9
5	5	8
6	7	3
7	6	6
8	2	11
9	10	12

The algorithm for calculating the τ-Kedall coefficient is as follows:

1) sort the records by one of the characteristics (in our example - X) in ascending order;
2) calculate the ranks of the values for each column separately and write them next to each other (Table 4);
3) calculate the number of matches: for each record, count how many times its rank in Y is less than the ranks of the records below it and enter this number in column P;
4) calculate the number of inversions: for each record, count how many times its rank in Y is greater than the rank of the records below it;
5) calculate the τ-Kendall coefficient using formula [19]:

$$\tau = \frac{\sum P - \sum Q}{n \cdot (n-1)/2}\ , \qquad (3)$$

where ΣP is the number of matches; ΣQ is the number of inversions; n is the total number of records.

The scheme of algorithm is shown in Figure 3.

To implement the developed algorithm, the Record and Kendall classes were created. The Record class contains all the data about one row of the table of values and has the following fields: num - ordinal number of the record, x - value of the first sample, y - value of the second sample, rank1 - rank of the value of the first sample, rank2 - rank of the value of the second sample. The Record class has the following methods: setRank1() - sets the rank of the element of the first sample, setRank2() - sets the rank of the element of the first sample, setX() - sets the value of the element of the first sample, setY() - sets the value of the element of the second sample.

Table 4: Calculation of Kendall correlation coefficient.

Num	*x*	*R(x)*	*y*	$\boldsymbol{R(y)}$	Coincidence (P)	Inversion (Q)
1	1	1	4	3	6	2
2	2	2	11	8	1	6
3	3	3	9	7	1	5
4	5	4	8	6	1	4
5	6	5	6	5	1	3
6	7	6	3	2	2	1
7	7	7	5	4	1	1
8	9	8	1	1	1	0
9	10	9	12	9	0	0

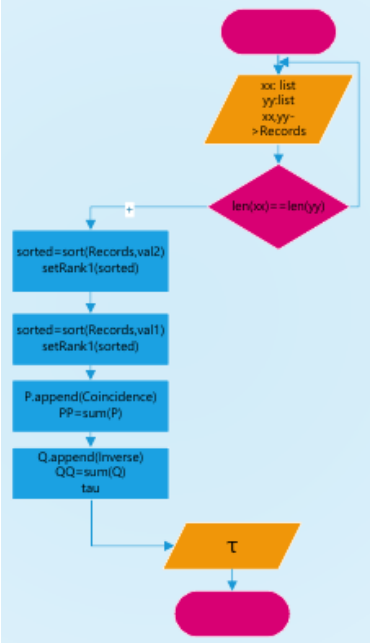

Figure 3: Schematic of the algorithm for calculating the Kendall correlation coefficient.

The Kendall class contains all the samples, initial, intermediate and final data from calculating the Kendall correlation coefficient. The Kendall class has the following fields: x1 is the value of the first sample, y1 is the value of the second sample, Records is a list of Record class objects, N is the number of values in each sample, sorted is a sorted list of Record class objects, P is a list of matches, Q is a list of inversions, PP is the number of matches, QQ is the number of inversions, tau is the value of the Kendall correlation coefficient. The Kendall class has the following methods: init() is a constructor, correlation() is the calculation and return of the correlation coefficient, concl() is the formulation and return of the conclusion. The generated PDF report contains the value of the correlation coefficient, a conclusion about the nature of the relationship between the samples, and a plot of the distribution of the samples being compared. The results of testing the algorithm for calculating Kendall's correlation showed that the algorithm had a complexity of $O(n^2)$. The time to process one pair of values increased proportionally to the increase in sample size.

The developed Correlation library contains the following classes (Fig. 4).

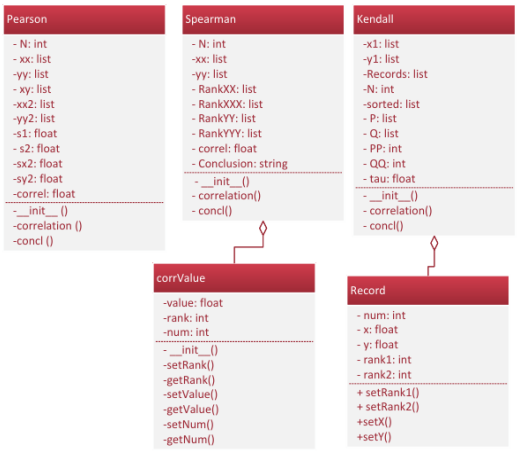

Figure 4: Class diagram of the correlation software library.

The created software library contains 3 main classes that calculate Pearson, Spearman, and Kendall correlation values, as well as additional classes used in the Spearman and Kendall classes.

The created software libraries were integrated into the application, which has a graphical user interface implemented in two versions: one based on the PyQt 5 library and the other based on the Custom Tkinter library. Their use is because the standard Tkinter library lacks tools for creating menus. The interface of the application based on the PyQT 5 library is shown in Figure 5.

The graphical interface of the Correlation Checker application contains a menu consisting of 3 sections: Input data, Data, and Save results. The developed Correlation Checker 1.0 application has the following options for entering initial data: from a text file, from a SQLite3 database, from a table on a form. After entering the initial data, they can be seen in the form of a table. To do this, you need to select the Data menu option. Using the Save results menu, you can generate a report on the correlation study in the form of a PDF document. The main application window contains radio buttons with which you can select the type of correlation: Pearson, Spearman, or Kendall. The Correlation Checker 2.0 application (Fig. 6) was also created based on the freely distributed Custom Tkinter library. The main menu options are the same as in the previous version.

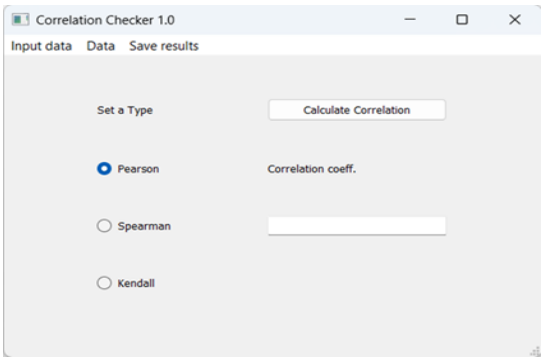

Figure 5: Application interface based on the PyQT 5 library.

The use of the Custom Tkinter library is because the PyQT 5 library is paid, and the Custom Tkinter library is free, although it has fewer widgets. The main window of the application also contains a field for entering initial data (Input data arrays). Further, the entered arrays are displayed in the DataArray1 and DataArray2 fields. This version of the application calculates all 3 types of correlation: Pearson, Spearman, and Kendall. Using the Report menu, you can generate a correlation report in the form of a PDF document. Therefore, applications based on the PyQT 5 and Custom Tkinter libraries have the same functionality and can be used to study the relationship between two data sets.

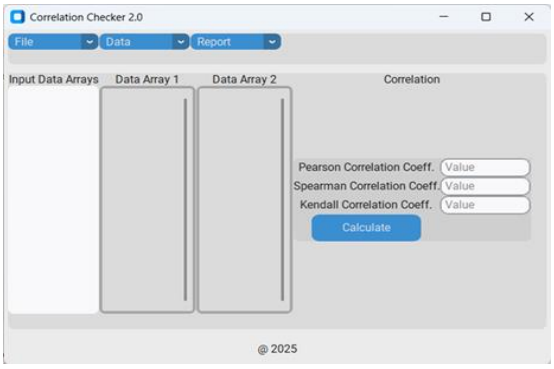

Figure 6: Interface of the application based on the Custom Tkinter library.

Since both versions of the Correlation Checker application use the same libraries, the results they calculate will be the same. Therefore, we used MS Excel spreadsheets for verification. This software tool contains built-in functions for calculating Pearson, Spearman, and Kendall correlation coefficients. For the same input data (Tables 1, 2, 4), MS Excel produced the following values: Pearson=0.640411, Spearman=0.643, and Kendall=0.64286. The program we developed with the same input parameters (Tables 1, 2, 4) displayed the following values: Pearson=0.64041, Spearman=0.64321, and Kendall=0.64286. Therefore, based on the verification of the results obtained, we can conclude that the developed program calculates the correct result.

5 CONCLUSIONS

The developed Correlation Checker application is based on the Correlation library we developed, the classes and methods of which provide the calculation of the Pearson, Spearman, and Kendall correlation coefficients. This application is intended for data analysis, as well as for calculating the specified correlation coefficients during the study of the academic discipline “Fundamentals of Scientific Research”. The developed application has the following advantages: open source; ease of modernization and development (the application is written in Python); modular architecture of the application; use of only built-in Python data types in the Correlation library. Thanks to this, the application can be upgraded in the future, even if support for libraries such as PyQT and Custom Tkinter is discontinued. It will only be necessary to rewrite the interface using other libraries. The generated PDF report does not contain instructions for interpretation, but a conclusion about the existence of a relationship between the two samples.

REFERENCES

[1] R. Taylor, “Interpretation of the correlation coefficient: a basic review,” Journal of Diagnostic Medical Sonography, vol. 6, no. 1, pp. 35-39, 1990.

[2] B. Ratner, “The correlation coefficient: Its values range between +1/-1, or do they?” Journal of Targeting, Measurement and Analysis for Marketing, vol. 17, no. 2, pp. 139-142, 2009.

[3] D. Chicco and G. Jurman, “The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification,” BioData Mining, vol. 16, no. 1, p. 4, 2023.

[4] J. Jiang, X. Zhang, and Z. Yuan, “Feature selection for classification with Spearman’s rank correlation coefficient-based self-information in divergence-based fuzzy rough sets,” Expert Systems with Applications, vol. 249, p. 123633, 2024.

[5] S. Chatterjee, “A new coefficient of correlation,” Journal of the American Statistical Association, vol. 116, no. 536, pp. 2009-2022, 2021.

[6] X. Su, S. Shang, Z. Xu, H. Qian, and X. Pan, "Assessment of Dependent Performance Shaping Factors in SPAR-H Based on Pearson Correlation Coefficient," CMES-Computer Modeling in Engineering & Sciences, vol. 140, no. 2, pp. 23-26, 2024.

[7] H. Yu and A. D. Hutson, "A robust Spearman correlation coefficient permutation test," Communications in Statistics - Theory and Methods, vol. 53, no. 6, pp. 2141-2153, 2024.

[8] F. Han and Z. Huang, "Azadkia-Chatterjee's correlation coefficient adapts to manifold data," The Annals of Applied Probability, vol. 34, no. 6, pp. 5172-5210, 2024.

[9] C. A. Mertler, R. A. Vannatta, and K. N. LaVenia, Advanced and Multivariate Statistical Methods: Practical Application and Interpretation, Abingdon-on-Thames: Routledge, 2021.

[10] V. Mykhalchuk, "Assessment of the objectiveness of cognitive prejudice mitigation in a neuroeducational strategy with the highlighting of vulnerable indicators," Measuring and Computing Devices in Technological Processes, vol. 1, pp. 52-58, 2025.

[11] H. Kang, "Sample size determination and power analysis using the G*Power software," Journal of Educational Evaluation for Health Professions, vol. 18, pp. 52-58, 2021.

[12] R. Bivand, G. Millo, and G. Piras, "A review of software for spatial econometrics in R," Mathematics, vol. 9, no. 11, p. 1276, 2021.

[13] L. Qi, W. Lin, X. Zhang, W. Dou, X. Xu, and J. Chen, "A correlation graph based approach for personalized and compatible web APIs recommendation in mobile app development," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 6, pp. 5444-5457, 2022.

[14] S. Mguil-Touchal, F. Morestin, and M. Brunei, "Various experimental applications of digital image correlation method," in WIT Transactions on Modelling and Simulation, vol. 17, pp. 1227-1236, 2024.

[15] S. D. Rahmawati, A. F. Ihsan, D. Darmadi, F. Priadi, and U. W. R. Siagian, "Web-Based Application for Calculation of Physical Properties of Hydrocarbon Fluids Using Compositional Approach," Journal IATMI, vol. 17, pp. 1-12, 2022.

[16] V. M. Bazurin, O. I. Pursky, and O. S. Chashechnikova, "Application for Statistical Processing of Pedagogical Experiment Results: A Component-Based Approach," in 2025 International Conference on Inventive Computation Technologies (ICICT), Apr. 2025, [Online]. Available: https://doi.org/10.1109/ICICT64420.2025.11004909.

[17] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in Noise Reduction in Speech Processing, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1-4.

[18] J. C. De Winter, S. D. Gosling, and J. Potter, "Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data," Psychological Methods, vol. 21, no. 3, p. 273, 2016.

[19] H. Abdi, "The Kendall rank correlation coefficient," in Encyclopedia of Measurement and Statistics, vol. 2, 2007, pp. 508-510.