

**Artículo de investigación****Specialized platform for the analysis of mathematical models in economics**

Специализированные платформы анализа математических моделей в экономике

Plataformas especializadas para el análisis de modelos matemáticos en economía

Recibido: 22 de agosto del 2019

Aceptado: 17 de septiembre del 2019

Written by:

**Yuri Alexandrovich Kostikov<sup>11</sup>****Alexey Viktorovich Mokryakov<sup>12</sup>**

SPIN-code: 8582-8288

**Alexander Mikhailovich Romanenkov<sup>13</sup>**

SPIN-code: 7586-0934

**Abstract**

Evaluation of the effectiveness of the mathematical model in the economic field is one of the pressing problems at the present time. A number of parameters can often be neglected when simplifying the model; however, it is difficult to assess how adequate the model will remain. In order to get this estimate, it became necessary to determine how the behavior of the model has changed, how the data and their nature have changed. Comparative and statistical analyzes help to find answers to the questions posed. A platform that can evaluate the impact of changes made to the model by some criteria and compare the changed model with its original version using statistical analysis methods was implemented to study this problem.

**Key words:** Platform, mathematical model, economic model, automation, statistical analysis, regression analysis, correlation analysis.

**Аннотация**

Оценка эффективности работы математической модели в экономической области является одной из актуальных проблем в настоящее время. Часто при упрощении модели рядом параметров можно пренебречь, однако достаточно трудно оценить, насколько адекватной она останется. Для того, чтобы получить данную оценку, возникла необходимость определить, насколько изменилось поведение модели, как поменялись данные и их характер. Найти ответы на поставленные вопросы позволяют сравнительный и статистический анализы. Для исследования этой проблемы была реализована платформа, которая может по некоторым признакам оценить влияние внесенных изменений в модель и сравнить с измененную модель с ее исходной версией.

**Ключевые слова:** платформа, математическая модель, экономическая модель, автоматизация, статистический анализ, регрессионный анализ, корреляционный анализ.

<sup>11</sup> Candidate of Physical and Mathematical Sciences, Head of Department 812 Moscow Aviation Institute (National Research University) Moscow, Russia

<sup>12</sup> Candidate of Physical and Mathematical Sciences, Associate Professor Department 813 Moscow Aviation Institute (national research university) Moscow, Russia.

<sup>13</sup> Candidate of Technical Sciences, Associate Professor Department of 812 Moscow Aviation Institute (National Research University). Senior Researcher Federal Research Center Federal Research Center "Computer Science and Management" of the Russian Academy of Sciences Moscow, Russia

## Resumen

La evaluación de la efectividad del modelo matemático en el campo económico es uno de los problemas apremiantes en la actualidad. Una serie de parámetros a menudo se pueden descuidar al simplificar el modelo; sin embargo, es difícil evaluar qué tan adecuado seguirá siendo el modelo. Para obtener esta estimación, se hizo necesario determinar cómo ha cambiado el comportamiento del modelo, cómo han cambiado los datos y su naturaleza. Los análisis comparativos y estadísticos ayudan a encontrar respuestas a las preguntas planteadas. Para implementar este problema, se implementó una plataforma que puede evaluar el impacto de los cambios realizados en el modelo mediante algunos criterios y comparar el modelo modificado con su versión original utilizando métodos de análisis estadístico.

**Palabras clave:** Plataforma, modelo matemático, modelo económico, automatización, análisis estadístico, análisis de regresión, análisis de correlación.

## Introduction

In the modern world, many non-trivial tasks are solved with the help of the latest information technologies. A large number of these tasks require modeling of a specific process for their solution. Creating real conditions for modeling such processes can be very difficult, and often even impossible. Models based on information technology that can simulate almost any process come to the rescue. For example, a physical, mathematical process or imitation of the work of a small enterprise. Such models have several advantages: they are relatively "inexpensive" in terms of development; flexible enough for change; can bring work results closer to real conditions. However, the development of such a model takes a lot of labor and time. Often, it is necessary to apply optimization to already implemented models without violating the principles of its operation. Such optimization is a very complex process; many tests and analyzes are needed to verify that the model has remained effective. In addition to optimization, there is also a process of refinement of the model, in which changes, due to additional parameters, are made to the system under study. This process requires even more time and attention when analyzing the results. These shortcomings determine the general urgent problem in the development and testing of the studied models.

To solve this problem, the task was set: to develop a software package, the purpose of which is to simplify the development of models, as well as automated testing of models to assess their "adequacy". This task must be divided into several subtasks, which will simplify and structure the solution of the main problem (Malysheva, Mokryakov and Nenakhov, 2016 b). The first subtask is the formation of the functionality of the developed software complex. Subsequent subtasks related to the development of the application architecture, user-friendly

interface, data warehouse, etc. will be decided depending on the saturation of the system's functionality. The system's functionality should allow to create projects with different versions of the model under study, provide the user with the ability to edit input parameters, use several methods of analysis of the object under study, and also automatically run and analyze several versions of the model with different input parameters in an automated mode. Other functionality should include the ability to provide the user with several formats for presenting the results of the software package, for example, in graphical representation in the form of histograms, charts, and graphs, or in publicly structured formats XLS, JSON (Bassett, 2015), XML.

Based on the system functionality described above, the second sub-task is to create a universal, dynamic architecture of the software package (Mokryakov, Nenakhov and Nenakhova, 2017). The framework created on the basis of this architecture will play the role of a central axis for the further development of the platform. The architecture must be dynamically expandable to support the use of several options for data analysis and testing. Accordingly, each object in the system must be independent of the others. Moreover, all elements of the system must be replaceable if necessary to expand the functionality. Since the platform implies several different stages of execution, the architecture of this application should have the property of maintaining state, which will allow saving the results of the program at various stages of its execution.

Another important subtask is to determine the mathematical apparatus of the system, which will perform an analysis of the obtained model results (Malysheva, Mokryakov and Nenakhov, 2016 a).

The mathematical apparatus should be quite simple to implement and resistant to changes in the analysis system. However, the most important requirement for a mathematical apparatus is its effectiveness. The result of the work of the mathematical apparatus should demonstrate the result of the implementation of the investigated model. To solve this subproblem, the methods of mathematical statistics are most effective, which allow us to determine the significance, as well as the dependence of the parameters used in the model. In particular, it is necessary to consider regression and correlation analyzes for their implementation in the mathematical apparatus. An equally important subtask is the creation of a convenient user interface that will provide the user with an obvious algorithm of actions for launching testing and analysis of various versions of the model in the platform. The interface should be intuitive, as well as meet the modern requirements of application design.

Since the system provides support for several versions of the objects under study, it is necessary to single out another subtask, which is formulated as determining the interaction of the platform with the data warehouse. The data warehouse should effectively store the results of the application, as well as various meta-information, which is necessary for the correct operation of the platform.

### **Mathematical apparatus**

Based on the described functionality of the developed system and the task, it is necessary to determine the mathematical apparatus, which will be the core of the platform. The choice of the mathematical apparatus is very important for the software complex, since the effectiveness of mathematical methods will determine the overall effectiveness of the entire platform when analyzing the results of its work. Given that the data analysis will be carried out in a separate module of the system, it is necessary to choose a mathematical apparatus that is quite simple to implement and does not require special computing power of a computer. Based on the system's functionality, the analysis of the obtained data will be reduced to the task of determining the degree of dependence between the various variables present in the studied object and comparing the results with the standard to assess the criticality of their deviation. Similar problems are usually solved with the help of mathematical statistics. In modern mathematical statistics, many methods allow to evaluate the degree of connection between two variables.

The multiple regression method is one of the most popular and well-known methods of mathematical statistics. The task of multiple linear regression is to build a linear model of the relationship between a set of continuous predictors and a continuous dependent variable (Draper and Smith, 2007; Nasledov, 2013). In addition to the multiple regression method, there is a correlation analysis that can solve the problems under consideration. The purpose of the correlation analysis is to identify estimates of the strength of the connection between random variables (attributes) that characterize some real process (Draper and Smith, 2007; Nasledov, 2013).

These methods of mathematical statistics are excellent for implementing the task of data analysis in the developed software package. However, the use of these methods separately does not carry much efficiency when testing real problems. The methods discussed above must be combined and an analysis of the results obtained from each of them by the method of comparative analysis be carried out. A comparative analysis, based on data obtained from correlation analysis and multiple regression, will be able to give a more accurate assessment of the degree of dependence of the studied model parameters. Based on these results, a comparative analysis will be able to assess the adequacy of the data obtained by comparing the values with the selected standard. When comparing the values, the degree of interconnection, as well as the nature of the interconnection of the various components, will be taken into account. Information on each component about its significance in the model and how adequate it is with respect to the corresponding standard will be obtained.

### **The main functionality of the platform**

The platform for automated testing and analysis of mathematical and simulation models includes a fairly rich functionality that ensures the efficient operation of the entire software package. The concept of this work was implemented in a separate work (Malysheva, Mokryakov and Nenakhov, 2016 b).

Before starting the application, first of all, the platform configuration file must be configured. In particular, the configuration file is a JSON file. The following entities are defined in the JSON structure:

- A collection of key / value pairs. In different languages, this concept is

implemented as an object, record, structure, dictionary, hash, named list or associative array.

- An ordered list of values. In most languages, this is implemented as an array, vector, list, or sequence.

These entities are universal for many popular programming languages. Therefore, the JSON format is used in many modern information systems as a standard for data exchange.

The developed platform operates with several main entities. The first entity is a project. A project is an object that contains a name, description, and a nested list of versions. We denote the project by  $P(\vec{V})$ , где  $\vec{V} = (v_1, \dots, v_n)$  is the vector of versions inside the project. A project is the top of the hierarchy in a platform system. The project includes an object of study, that is, a model, parameters for launching the model, and tools for analyzing the results of model execution. The user can create, edit and delete projects. According to the logic of the platform, two projects are two different studied models, however, the user is unlimited in creating projects, and can create absolutely any projects.

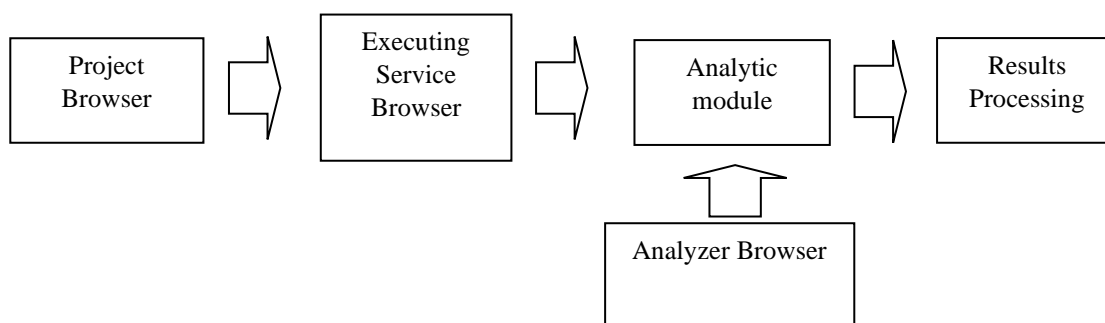
There is one more entity for maintaining versioning and more convenient perception of the object under study in the platform - version. A version is an atomic object of a software complex, which is located below the project in the system hierarchy. We can say that the version is an integral part of the project. Version contains name; description; model as an object of study; list of parameters to run the model; list of analyzers, in order of data processing; the results of the model, which are necessary for further processing. A version can be identified as follows:  $V(M(\vec{J}_m), \vec{A})$ , where  $M$  is the model studied in this version,  $\vec{J}_m$  is the set of model parameters,  $\vec{A}$  is the set of analyzers used in the study.

Another type of entity in the system is the analyzer. The analyzer is a tool for analyzing the data that the platform receives when you run the model with certain parameters. The analyzer contains properties such as name, description, and processing tool. The analyzer is directly related to the version. One version may have several analyzers. Moreover, the order of the analyzers is important, since the data will come

sequentially from one analyzer to another. The analyzer is an independent object in the system and can also be edited, deleted or downloaded. This entity uses an executable file loaded into the system as an analysis tool. Since the analyzer uses an executable file, it does not matter what language the tool was written in. It is only necessary to take into account some requirements when developing a tool. Firstly, the executable must be in exe or jar format. Secondly, the executable file must accept a JSON file, which has a certain structure, as an input parameter. The input file will contain the data that needs to be examined, as well as additional data to form the result of the work. As a result of its work, the program should create a file, the path of which is indicated in the input file, in JSON format with a specific structure, which includes the analysis results and other service information for the platform to work. Thus, we can specify the analyzer as  $A(\vec{J}_a)$ , where  $\vec{J}_a$  are the input parameters for the analyzer.

One of the main entities of the system is the model. A model is the essence of a system that contains an object of study, that is, a mathematical or simulation model. The model is contained in the version in a single copy. This entity, as well as other entities in the platform, has certain properties: name, description and executable file of the model. The model executable file has the exe format, and, like the analyzer executable file, is independent of the development language. In order to make the platform more universal, it is necessary to make a certain standardization of model development for this platform. This standardization is not a limitation, but only a general rule for creating models. The model executable file must accept a file with startup parameters in JSON format as input. This file contains the launch parameters of the model, as well as the meta information that is needed to create the resulting file. The result of the model program is the resulting JSON format file with the absolute path specified in the model input file.

In addition to the basic entities described above, the platform contains several more entities that are necessary for the effective operation of the platform. These entities allow the platform to support all the basic functionality, and also give it such an important property as saving the execution state.



**Figure 1.** The order of study model

As mentioned earlier, the software package is a system that has a modular architecture. This means that the platform consists of several relatively independent modules interconnected by a specially developed protocol. One of these modules is the project explorer. The project browser allows the user to view detailed information about projects and model versions. The module also organizes a convenient graphical interface in the form of a tree hierarchy of projects and versions, as well as a separate form for displaying a particular object.

Detailed information on the version of one of the projects can be viewed, for example, name, description, model parameters and a list of analyzers in the order of data processing. The browser also allows to create new projects and new versions of projects. When creating a new project or version, this module creates the necessary files in the user's file system, or if a database is specified, entries are created in the specified collections.

When creating projects and versions, it is necessary to specify the required parameters for these objects, otherwise the objects will not be created. A project in the system can exist without versions, however, a version of the project cannot exist without a model of the object under study. Also, when creating a version, the user is prompted to enter input parameters for the model.

To simplify the input of parameters for the model, the following scheme was implemented. The parameter name, its template value is indicated, then the range to the left and to the right of the value template is indicated. Value range means the maximum and minimum threshold for changing the template value. Then the value by which the initial value is changed to achieve its minimum and maximum threshold is indicated. There are models that require a very

large number of parameters to work, so the module provides for mass loading of data from different file formats. The module allows to load parameters from files with the JSON, XML, CSV format. The module also provides the ability to edit already entered parameters, as well as delete them.

In addition to the parameters, the executable file of the model that will be examined, as well as its name and description must be selected so that the user can determine what model is being tested in the future. Also, when choosing a model, one more important parameter is indicated, which is called the standard. The reference model is that object with which all subsequent versions of the project are compared. The standard sets the degree of adequacy of the system.

Finally, the project explorer also makes it possible to select from the list of analyzers already existing in the system those that will analyze the object of study. Each analyzer can be selected and see its description, which contains the principle of its operation. If the user understands that this analyzer suits him, then he can add it to the version. So a different number of analyzers can be added. When analyzers are added, a special list is created. The order of the analyzers is very important in this list, since the data from one analyzer to another will be received sequentially in this order. In this regard, the project explorer has a functionality that allows to swap analyzers when editing and creating versions. Editing projects and versions is provided with the same functionality as when creating these entities. The project browser also has the ability to run several versions of projects for testing and analysis.

Similarly, a module called the analyzer browser is also implemented in the platform. The analyzer browser provides functionality for creating, deleting and editing analyzers in the system. The

creation process is the loading of an executable file into the platform, as well as assigning a name and description to it. Upon opening the module, the user will be presented with a list of analyzers already loaded into the platform. This list can be edited. Any analyzer is selected, its name, description and executable file can be changed. Also, analyzers can be deleted. When creating each analyzer, the platform creates special files in the local file system or in the database, if one is specified.

According to the statement of the problem, the platform should investigate a model running with various parameter values. The platform should be able to run several modeling processes with different input data and process the results. An executing service module was created as an implementation of such functionality. This module is capable of running several processes in parallel with different input data. In particular, the module launches several instances of the investigated model equal to the number of combinations of its input parameters. After the completion of each instance of the model, the module in an automated mode collects the results of their work and does their preliminary processing for transmission to the next module. This module uses several paradigms of parallel data processing. Since the formats and structures of the input and output files of the model are standardized, the module is universally suitable for working with any models that are implemented for this platform.

The next module, to which data is transferred from the executing service, is called analytical. This module accepts specially prepared JSON received from the executing service as an input file. The module prepares a list of analyzers that this file should go through. Further, the module sequentially launches several analyzers that were specified in the tested version. The input to the next analyzer is the output from the previous one. Thus, the module transfers data in a chain from one analyzer to another. In the end, a file will be received, which will be pre-processed and transferred to the next module. The main essence of this module is to maintain the execution order of the analyzers and to maintain the integrity of processing. In that module, the main analytics and data processing takes place, based on which it will be possible to assess the adequacy of the model. The development of analyzers for this module is not an easy task, because a fairly powerful mathematical apparatus is used to solve such problems.

After analytics, the prepared data is sent for processing to the results processing module. This module processes and parses the input file. The data from the input file are the sources when conducting a comparative analysis of the version of the project in question with the project standard. In the comparative analysis, the deviations of the dependent variables of the standard and the tested version are compared. Based on the data obtained, it can be concluded that the changes made to the model of this version are critical compared to the standard. The comparative analysis module displays the results in the form of a histogram. This form is the most convenient for visual assessment of the comparison of results. The module also provides the ability to graphically display the results of correlation and regression analyzes. The data are presented in the form of function graphs, which are the dependencies of the model parameters from each other, as well as the dependency functions of the dependent and independent variable. Based on the information displayed, it is possible to assess the degree of dependence of the variables and how much they affect the final result. Assessment can be negative, positive, neutral and critical.

In addition, the module provides the ability to view a full report on the results of the tested model, as well as on the results of each analyzer at a certain stage of testing. The report can be downloaded in a user-friendly format, for example, in XLS, or for further analytics, or for collecting statistics, or for other purposes. As soon as the data appears in the results processing module, it is stored in the local data warehouse or in the database, depending on the system settings. This is done so that the user can see the results of the system at any time and compare them with various standards. Such a scheme of operation makes the system more efficient and user-friendly.

When testing starts and analytics of a certain version of the project occurs, the model goes through several stages of data processing:

- Preparing the model for launch.
- Formation of a file with parameters for launching model instances.
- Parallel launch of multiple instances of the model.
- Collection of the results of model instances and their preliminary processing.
- Preparation and launch of analyzers.
- Processing the results of the analyzers and displaying the final result.

Each stage takes a certain time to complete, especially the stages of launching models and analyzers. There is a chance that something will go wrong at a certain stage and the stage will fail. In this case, it would be convenient for the user to monitor the progress of testing and analytics. To solve this problem, a monitoring system was developed for ongoing tasks. After launching a specific version, this project and its version are included in the monitoring list. After that, on the monitoring tab, it will be possible to select the monitored project and its versions and monitor the progress of the task.

As mentioned earlier, the system is modular, dynamically expandable. The modules in the system are relatively independent of each other. In order to ensure data transfer between these modules, as well as to provide support for the dynamic extensibility of the system, a special protocol for the interaction between the modules was developed. This protocol provides not only reliable data transfer between the modules, but also the preservation of the state of the system at certain stages of its implementation, so that in case of an unexpected shutdown of the platform, during testing and analysis, progress is saved. At the subsequent launch of the platform, the system continued to carry out its tasks from the stages at which it stopped. As atomic data transfer objects, JSON files are used, as well as serialized objects to save the state of the stages. The protocol stores special meta information and intermediate results for each individual stage in serialized files. If at the time of launching the platform there are protocol files in the system that indicate incomplete testing and analysis of some tasks, then the user will be asked to resume the suspended tasks. If the user refuses, the data will be deleted and the launch history will be cleared. In addition to supporting statefulness, the protocol ensures the integrity of data transfer between modules by checking JSON format files for structural integrity. If the structure is broken, the platform will inform the user about this and complete the testing task.

The architecture under consideration allows us to use the system not only for testing and analysis of mathematical and simulation models, but also for tasks of streaming data processing. Such problems can be solved, for example, by increasing the chain of analyzers with which data will be processed sequentially from one executable file to another. Data transfer and their integrity is ensured by a protocol that is capable of storing the state of the system. Another plus of this platform is its simple and open architecture. Other programmers for narrower and special

tasks can supplement this architecture with other ideas.

### **Methods of statistical analysis of model adequacy**

Often in the process of compiling a model there is a need to assess its adequacy and effectiveness. What is adequacy? What is efficiency? How can you evaluate them? Answers to these questions allow advancement in the evaluation of economic and mathematical models.

The adequacy of the model is usually understood as the coincidence of the properties (functions of parameters, characteristics) of the model and the corresponding properties of the modeled object. Accordingly, the assessment of the adequacy of the model is a check of the model's compliance with the real system. It is estimated by the proximity of the calculation results to experimental data.

Before evaluating the mathematical model, it is necessary to conduct a series of checks:

- Does the model use all the essential parameters of those available in the simulated area?
- How many model parameters are not significant for the model?
- Is the functional relationship between the various parameters of the model correctly established?
- Are the ranges of permissible values of the model parameters set correctly?
- How correctly does the model react to changes in its parameters on all ranges of values (especially the absence of illogical and physically impossible states of the model)?

If we consistently evaluate all the points outlined above, we can not only solve the question of the adequacy of the model, but also avoid serious errors in building the model. However, this list of checks is not mandatory and can be supplemented by a number of points, or shortened. The scientific community has not yet proposed rigorous methods for assessing the adequacy of the model.

In the platform under consideration, the analytical module is intended, inter alia, to make a preliminary assessment of the model, as far as possible, based on identifying the degree of dependence of the model parameters, critical parameters, non-essential parameters and presenting a forecast for further model

optimization. One of the main analyzes conducted to evaluate the model is a correlation analysis, namely Pearson correlation. Further, we will consider in more detail what the essence of this analysis is.

The main purpose of correlation analysis is to identify the relationship between two or more studied variables, which is considered as a joint coordinated change in the two studied characteristics. This variability has three main characteristics: shape, direction and strength.

At the moment, there are a number of correlation coefficients. Quite often, the r-Pearson, r-Spearman and  $\tau$ -Kendall correlation criteria are used.

If, due to the correlation coefficient, a relationship between two parameters was revealed, then a causal relationship does not necessarily exist. The connection found only indicates that the variables do not behave quite randomly and possibly influence each other, while it is impossible to establish who is the leader and who is the slave only using correlation.

As already mentioned, Pearson correlation is used in our module. The form of the correlation coefficient is formed in such a way that allows to detect a linear relationship between two variables. Therefore, it is also called the Pearson linear correlation coefficient.

In general terms, the formula for calculating the correlation coefficient is as follows

$$r_{xy} = \frac{\sum(x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 * \sum(y_i - \bar{y})^2}}$$

where  $x_i$  is values taken by variable X,  
 $y_i$  is values taken by variable Y,  
 $\bar{x}$  is X sample mean,  
 $\bar{y}$  is Y sample mean.

The calculation of the correlation coefficient may lead to incorrect conclusions if:

- The actual relationship between the two parameters is non-linear (power, exponential, logarithmic, etc.);
- A sample is used that combines several tests with different sets of parameters;
- The sample contains particularly prominent values (outliers);
- The selection contains easily clustered values.

In this paper, we calculate the correlation coefficient for each of the pairs of parameters for the further distribution of pairs of values ("parameter X - parameter Y": "value of the correlation coefficient") by categories:

- Positive (positive correlation);
- Negative (negative correlation);
- Neutral (zero correlation);
- Critical (coefficient value  $\geq |0.7|$ ).

After calculating the correlation coefficients, another analysis is performed - regression analysis, or rather, the multiple regression equation is calculated.

In our platform, we can solve two problems of regression analysis: find the form of dependence between the parameters and determine the regression function.

The nature and form of the relationship between the variables can form the following types of regression:

- Positive linear regression (expressed in uniform growth of function);
- Positive uniformly accelerating regression;
- Positive equally slowly increasing regression;
- Negative linear regression (expressed in a uniform drop in function);
- Negative equally accelerating decreasing regression;
- Negative equally slowly decreasing regression.

It is worth noting that these forms of regression are rarely found in these problems without mixing with each other. Then we will talk about combined forms of regression.

Speaking about the calculation of the regression function, we must evaluate the effects on the dependent parameter of other important parameters of the model, provided that the other parameters of the model are constant, as well as there are no random effects on the parameter under study. *The regression function* is defined as a mathematical equation of one type or another.

The following linear regression equation is in sufficient demand:

$$Y = \sum_{i=1}^n a_i x_i + b_0 + e,$$



where  $a_i$  are regression (angular) or B-coefficients,  $b_0$  is a free term (if used),  $e$  is a term containing an error.

Thanks to this linear regression equation, mathematical models in economics, medicine, and a number of other fields are well described.

Under the condition that the parameter depends on many parameters, we go into multidimensional space. In the general case, the multiple regression function will have the following form:

$$Y = a + b_1 * X_1 + b_2 * X_2 + \dots + b_p * X_p.$$

Regression coefficients (or B-coefficients) represent the independent contributions of each independent variable to the prediction of the dependent variable.

The analytical module allows to find the coefficients of the multiple regression function.

When we want to use multiple regression, that is, the temptation to maximize the number of influencing parameters in the formula, we hope that in this way we take into account all significant parameters. However, we forget that a large number of non-essential parameters will still affect the form of regression, which will increase its error and reduce the possibility of using regression to predict the behavior of the model. In addition, the regression should be built on a large sample to reduce possible errors. Many authors believe that analyzing any parameter for each of the other parameters requires at least 10, and preferably 20, measurements. In this case, we can talk about the potential reliability of the regression lines.

To find multiple regression coefficients, data approximation methods are used, which naturally increases the error. The least squares method will help estimate the magnitude of such an error. The Least Squares method is based on the following:

- Interference and the parameters themselves have a different nature of occurrence, and therefore are independent;
- The mathematical expectation of an error should be equal to zero;
- The sample variance of the error is minimal.

The quality of the regression equation found can be estimated by the size of the error between the

empirical data and the parameter values according to the regression formula in certain sets of parameter values. Provided that the error is below a certain level, we believe that we have solved the problem of regression analysis. Otherwise, with a significant error, it is necessary to change the regression equation and repeat the parameter calculations. If there are several indicators in the model, then it is necessary to solve the problem of regression analysis for each of them separately.

In addition to the coefficients of the equation, the standard deviation is also found.

The standard deviation, or standard deviation, is a measure of the dispersion of a random variable relative to its mathematical expectation. The deviation is defined as the square root of the variance of a random variable and is calculated by the formula:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2},$$

where  $\sigma^2$  is variance;  $x_i$  – is the  $i$ -th element of the sample;  $n$  is the sample size;  $\bar{x}$  is average sample:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i.$$

The applied value of the standard deviation is to identify the degree of variability of the value of the studied value.

At the exit from the analytic module, a JSON file is generated, in which correlation coefficients, regression equation coefficients and standard deviation are entered.

Based on the obtained values, a conclusion is drawn on the importance of specific parameters for the model under study. With a large number of parameters insignificant for the model, the studied model can be considered at least overcomplicated, and possibly inadequate.

### Testing the platform on real data

Let's consider a concrete example in more detail. As an example of working with the System, we considered an econometric modeling of the cost of apartments in Moscow (Zamoskvorechye district). The data are presented by the Miel real estate agency and the database for rental and sale

of residential, commercial and suburban real estate “Realty.dmir.ru” (<http://Realty.dmir.ru>). As already described earlier in this work, this platform makes it possible to evaluate not only a clearly defined mathematical model, but also a model with data having a correlation dependence instead of a direct mathematical dependence.

To conduct an econometric modeling of the apartment market based on proposals for the sale of apartments, a sample was constructed containing 75 observations:

Evaluation of the price of 1 m<sup>2</sup> of an apartment is made by the following factors:

- Total area of the apartment
- Type of house (monolithic / panel)
- The presence of a balcony
- Kitchen area
- Type of housing (new / secondary housing)

In addition to the general report, the user can see the scattering diagrams of interest by selecting the abscissa (independent variable) and ordinate (dependent variable). The corresponding correlation coefficient can be seen in the report by the name displayed in the legend of the scattering diagram.

**Table 1.** Apartment data

Observation number	The cost of 1 m <sup>2</sup> of apartment	Apartment Area	House type	Balcony	Kitchen area	Type of housing
1	360 000	80	0	0	25,00	0
2	388 015	110	0	1	23,00	0
3	328 393	127	0	0	30,00	0
4	319 000	135	0	1	20,00	0
5	343 600	76	0	0	16,00	0
6	360 000	75	0	1	16,00	0
7	315 499	107	0	0	12,00	0
8	470 000	62	0	0	16,00	0
9	305 006	137	0	0	20,00	0
10	338 398	72	0	1	20,00	0
11	309 632	147	1	0	50,00	0
12	396 660	45	1	1	11,30	0
13	300 400	120	0	1	14,00	0
14	390 400	70	0	1	14,00	0
15	257 151	154	0	1	25,00	0
...	...	...	...	...	...	...
62	364 602	93	0	1	14,00	1
63	485 400	75	0	1	18,00	0
64	221 400	180	0	1	30,00	1
65	208 600	49	1	0	10,00	1
66	307 850	75	1	1	13,00	1
67	263 600	55	1	0	6,50	1
68	307 260	51	0	1	10,00	0
69	264 600	108	0	0	15,00	0
70	255 430	46	1	1	12,00	1
71	294 290	53	1	0	15,00	0
72	327 800	61	0	0	9,00	1
73	333 600	74	0	0	15,00	1

74	200 200	90	1	1	9,00	0
75	495 640	78	0	1	30,00	0

The results of the correlation analysis for this version of the model are presented below in the table:

**Table 2.** The results of correlation analysis

	<i>The cost of 1 sq.m. apartments</i>	<i>Area of apartment</i>	<i>Type of house</i>	<i>Balcony</i>	<i>Area of kitchen</i>	<i>Type of housing</i>
<i>The cost of 1 sq.m. apartments</i>	1,000	-0,146	-0,631	0,119	0,337	-0,528
<i>Area of apartment</i>	-0,146	1,000	-0,364	-0,034	0,505	-0,073
<i>Type of house</i>	-0,631	-0,364	1,000	0,505	-0,338	0,468
<i>Balcony</i>	0,119	-0,034	-0,066	1,000	-0,001	0,052
<i>Area of kitchen</i>	0,337	0,505	-0,338	-0,001	1,000	-0,421
<i>Type of housing</i>	-0,528	-0,073	0,468	0,052	-0,421	1,000

**Table 3.** The results of the regression análisis

<b>Step 1</b>	Coefficient
Constant	399882,01
Total area	-1231,89
House type	-95966,92
The balcony	9115,73
Kitchen area	3369,28
Type of housing	-16243,08

<b>Step 2</b>	Coefficient
Constant	406194,23
total area	-1246,79
House type	-97336,47
Kitchen area	3406,99
Type of housing	-14996,45

The results of the correlation analysis show that the correlation between the cost of 1 m<sup>2</sup> and the “Balcony” factor is weak (the presence of a balcony slightly affects the cost of 1 m<sup>2</sup>). Let's try not to take this parameter into account in the new version of the model in order to get new results with greater accuracy.

We give an economic interpretation of the coefficients of the model:

- The coefficient under the factor “Apartment Area” is statistically significant. The living area of the apartment significantly affects its price.

- The coefficient under the fictitious variable “House Type” shows that, *ceteris paribus*, the price of 1 m<sup>2</sup> in a monolithic house is 97,336.47 rubles. higher than in a panel house.
- The coefficient for the “Kitchen Area” factor is also statistically significant. With an increase in the area of the kitchen by 1 m<sup>2</sup>, the price of an apartment, *ceteris paribus*, increases by an average of 3406.99 rubles.

Next, consider the second example. In 2001, Jim O’Neill, Goldman Sachs Bank analyst, proposed the term BRIC (Brazil (BRA), Russia (RUS), India (IND), China (CHN)). The term refers to a number of countries, due to which, in the opinion of the analyst, the global economy will mainly be ensured. Since 2011, the South African Republic

(ZAF) has also been included in this club, and this group of states has become known as BRICS. Also in recent years, they say that Argentina is close in its parameters to the BRICS countries. It is worth noting that the BRICS countries account for about a quarter of the Earth’s territory, almost half of the world population and a quarter of world GDP.

Countries unite in this structure, usually in terms of economic growth and a number of other macro-indicators. Let’s try to analyze the proximity of countries in terms of gross domestic product per capita. In addition to the BRICS countries, we also take data for the USA (USA), Canada (CAN), Argentina (ARG) and Kazakhstan (KAZ). Data for analysis were taken from the World Bank website (Databank of World Bank).

**Table 4.** GDP per capita by country in \$

Year	ARG	BRA	CAN	CHN	IND	KAZ	RUS	USA	ZAF
2000	3749,8	24190,2	1229,0	443,3	7708,1	1771,6	3032,4	959,4	36334,9
2001	3156,8	23738,2	1490,9	451,6	7208,4	2100,4	2668,4	1053,1	37133,2
2002	2829,3	24169,3	1658,0	471,0	2593,4	2377,5	2502,3	1148,5	38023,2
2003	3070,9	28200,7	2068,1	546,7	3349,8	2975,1	3751,3	1288,6	39496,5
2004	3637,5	32034,3	2874,3	627,8	4277,7	4102,4	4833,7	1508,7	41712,8
2005	4790,4	36266,2	3771,3	714,9	5109,9	5323,5	5381,5	1753,4	44114,7
2006	5886,5	40385,9	5291,6	806,8	5919,0	6920,2	5602,0	2099,2	46298,7
...	...	...	...	...	...	...	...	...	...
2012	12370,0	52542,3	12386,7	1443,9	13082,7	15434,6	7501,4	6316,9	51603,5
2013	12300,3	52504,7	13890,6	1449,6	13080,3	16007,1	6829,0	7050,6	53106,9
2014	12112,6	50835,5	12807,3	1573,9	12334,8	14100,7	6428,3	7651,4	55033,0
2015	8814,0	43495,1	10510,8	1605,6	13789,1	9313,8	5733,1	8033,4	56803,5
2016	8712,9	42279,9	7714,8	1729,3	12790,2	8745,4	5262,1	8078,8	57904,2
2017	9880,9	45069,9	9030,3	1981,5	14591,9	10750,6	6120,5	8759,0	59927,9
2018	8920,8	46210,5	9331,0	2015,6	11652,6	11288,9	6339,6	9770,8	62641,0

After a correlation analysis, we got the following table:

**Table 5.** Correlation of GDP per capita between countries

	ARG	BRA	CAN	CHN	IND	KAZ	RUS	USA	ZAF
A									
R		0,857	0,768	0,896	0,889	0,855	0,799	0,828	0,681
G									
B									
R	0,857		<b>0,95</b>	0,766	0,809	<b>0,968</b>	<b>0,968</b>	0,738	0,905
A									
C									
A	0,768	<b>0,95</b>		0,74	0,801	<b>0,956</b>	<b>0,976</b>	0,787	<b>0,957</b>
N									
C									
H	0,896	0,766	0,74		<b>0,979</b>	0,818	0,753	<b>0,964</b>	0,635
N									
I									
N	0,889	0,809	0,801	<b>0,979</b>		0,829	0,786	<b>0,977</b>	0,729
D									
K									
A	0,855	<b>0,968</b>	<b>0,956</b>	0,818	0,829		<b>0,98</b>	0,797	0,874
Z									
R									
U	0,799	<b>0,968</b>	<b>0,976</b>	0,753	0,786	<b>0,98</b>		0,752	0,906
S									
U	0,828	0,738	0,787	<b>0,964</b>	<b>0,977</b>	0,797	0,752		0,693
A									
Z									
A	0,681	0,905	<b>0,957</b>	0,635	0,729	0,874	0,906	0,693	
F									

Based on the obtained correlation coefficients, the platform makes assumptions about the division of the countries in question into 2 clusters according to the indicator in question. The first cluster includes Brazil, Canada, Kazakhstan, Russia. In the second, India, China and the United States. Within each of the clusters, the pairwise correlation coefficient does not fall below 0.95. As can be seen from the studied indicator, the BRICS countries are not very similar, which, however, does not give us the opportunity to draw serious conclusions.

### Conclusion

The task was set: to develop a system for automated testing and analysis of economic, mathematical and simulation models, which is a universal platform that helps to quickly and accurately determine the assessment of the adequacy of the developed model. The platform should be flexible enough, scalable, relatively easy to use and effective for the tasks that the user will set. The system must be reliable and at the same time support the ability to store data about previous system launches. The result of the work of such a platform should be analytical results that will make it easy to determine the assessment of the adequacy of the studied object. As a result of solving the problem, a software package was created that allows to use modeling

applications developed in any programming language as objects of study. This system provides the user with the ability to enter parameters to start the simulation process, and also performs parallel tasks performed by the user. The platform also gives the user complete freedom to use, as an analytics tool, not counting the ones already available in the system, personally developed modules. Executable objects of models and analyzers should only satisfy fairly simple system requirements. The platform has a reliable and variable data storage system. A specially developed protocol allows to save the state of the system during its operation, which makes it possible to resume the task if it was not completed when the platform window was closed.

### References

- Bassett L. (2015). Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON. O'Reillym.
- Databank of World Bank. Available at: <https://databank.worldbank.org>.
- Draper N. and Smith G. (2007). Applied regression analysis. Moscow: Williams.
- Malysheva M. S., Mokryakov A. V. and Nenakhov E. V. (2016 a). Methods of statistical analysis of the significance of the changes made to the mathematical model. In: XLIII

---

International Youth Scientific Conference "Gagarin Readings - 2016". Moscow: MAI (NRU), 1, 225-226.

Malysheva M. S., Mokryakov A. V. and Nenakhov E. V. (2016 b). The concept of the platform for the analysis of mathematical and simulation models. In: XLII International Youth Scientific Conference "Gagarin Readings - 2016". Moscow: MAI (NRU), 1, 224-225.

Mokryakov A. V., Nenakhov E. V. and Nenakhova M. S. (2017). Platform for the automated analysis of mathematical and simulation models. In: XLIII International Youth Scientific Conference "Gagarin Readings - 2017". Moscow: MAI (NRU), 781-782.

Nasledov A. (2013). IBM SPSS Statistics 20 and AMOS: professional statistical data analysis. St. Petersburg: Peter.