

Міністерство освіти і науки України
Департамент освіти і науки Миколаївської облдержадміністрації
Миколаївське територіальне відділення МАН України

Відділення: комп'ютерних наук
Секція: технології програмування

Побудова трансформованого еліпсу передбачення в системі Scilab

Роботу виконала:
Приходько Олена Сергіївна,
учень 11 класу Миколаївської
гімназії №2 Миколаївської
міської ради

Науковий керівник:
Комаров Михайло Юрійович,
Учитель інформатики
Миколаївської гімназії №2

Миколаївське територіальне відділення МАН УКРАЇНИ

Анотація

Приходько Олена Сергіївна,
учениця 11-Б класу
Миколаївської гімназії №2,
Миколаївської міської ради,
Миколаївської області

Науковий керівник: Комаров Михайло Юрійович, учитель інформатики
Миколаївської гімназії №2

Побудова трансформованого еліпсу передбачення в системі Scilab

У роботі представлена розробка програми для побудови трансформованого еліпсу передбачення.

Було проаналізовано існуючі методи та математичні моделі для побудови трансформованого еліпсу передбачення, удосконалено математичну модель для побудови трансформованого еліпсу передбачення на основі застосування перетворення Джонсона, розроблено програму для побудови трансформованого еліпсу передбачення для системи Scilab.

Використання трансформованого еліпсу передбачення в порівнянні зі звичайним еліпсом передбачення дає змогу покращити моделювання області розподілу двовимірних негаусівських даних, що зокрема дозволяє підвищити достовірність визначення викидів у таких даних.

Ключові слова: Еліпс передбачення, перетворення Джонсона, двовимірні негаусівські дані, визначення викидів, Scilab, Sci - мова.

ЗМІСТ

ВСТУП	4
1 МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ ПОБУДОВИ ТРАНСФОРМОВАНОГО ЕЛІПСУ ПЕРЕДБАЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ	7
1.1 Існуючі методи та математичні моделі для побудови трансформованого еліпсу передбачення	7
1.2 Удосконалена математична модель для побудови трансформованого еліпсу передбачення на основі перетворення Джонсона	9
1.3 Засоби реалізації програми для побудови трансформованого еліпсу передбачення	12
1.4 Постановка задачі	14
2 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ ПОБУДОВИ ТРАНСФОРМОВАНОГО ЕЛІПСУ ПЕРЕДБАЧЕННЯ	17
2.1 Розробка ескізного проекту програми для побудови трансформованого еліпсу передбачення	17
2.2 Розробка технічного проекту програми для побудови трансформованого еліпсу передбачення	19
2.3 Розробка робочого проекту програми для побудови трансформованого еліпсу передбачення	24
ВИСНОВКИ	31
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	32
ДОДАТОК А – Текст програми	33

ВСТУП

Актуальність теми. При статистичному аналізі двовимірних даних використовується, як правило, еліпс [1, 2]. Завдяки еліпсу передбачення можна, наприклад, виявити двовимірні викиди (bivariate outliers) в даних, тому він часто застосовується для цього, зокрема в інформаційних технологіях. Статистичні методи, що базуються на еліпсі передбачення, використовують припущення, що дані апроксимуються двовимірним розподілом Гаусу. Але таке припущення може бути застосованим тільки в поодиноких випадках. Це призводить до необхідності побудови трансформованого еліпсу передбачення для конкретного набору даних, розподіл якого суттєво відрізняється від нормального [3]. Саме такими даними є, наприклад, трудомісткості розробки програмних проектів в залежності від кількості функціональних точок [4].

У роботі [3] був запропонований метод побудови трансформованого еліпсу передбачення для двовимірних негаусівських даних. Цей метод складається з наступних етапів. Спочатку двовимірні негаусівські дані нормалізують за допомогою бієктивного двовимірного нормалізуючого перетворення. Після цього отримують рівняння еліпсу передбачення для нормалізованих даних. На останньому етапі будують трансформований еліпс передбачення для двовимірних початкових даних на основі рівняння еліпсу передбачення для нормалізованих даних та зворотного перетворення. Але в роботі [3] не була наведена математична модель для побудови трансформованого еліпсу передбачення на останньому етапі, що призводить до необхідності подальшого її розвитку.

Крім того, існуюче програмне забезпечення (ПЗ), яке застосовується для обробки даних, наприклад, MatLab, Scilab, Excel, не має у своєму складі відповідних функцій, які дозволяли робити побудову трансформованого

еліпсу передбачення для двовимірних даних без застосування припущення про їх нормальність.

Саме тому, виникає потреба в удосконаленні відповідної математичної моделі та у розробці оригінальної програми для побудови трансформованого еліпсу передбачення для двовимірних даних.

Отже, удосконалення математичної моделі для побудови трансформованого еліпсу передбачення є актуальним та необхідним, оскільки дозволяє покращити моделювання області розподілу двовимірних негаусівських даних, за рахунок чого, наприклад, підвищити достовірність визначення наявності викидів у таких даних.

Мета і завдання дослідження. Метою наукової роботи є удосконалення математичної моделі для побудови трансформованого еліпсу передбачення на основі перетворення Джонсона і розробка програми для її реалізації, що дозволить покращити моделювання області розподілу двовимірних негаусівських даних. Для досягнення поставленої мети в роботі потрібно вирішити такі завдання.

- 1) Виконати аналіз існуючих математичних моделей для побудови трансформованого еліпсу передбачення для двовимірних даних.
- 2) Зробити постановку задачі на розробку програми для побудови трансформованого еліпсу передбачення для двовимірних даних.
- 3) Розробити ескізний, технічний та робочий проекти програми для побудови трансформованого еліпсу передбачення для двовимірних даних.

Об'єктом дослідження є процес побудови трансформованого еліпсу передбачення. *Предметом дослідження* є математична модель для побудови трансформованого еліпсу передбачення.

Методи дослідження, що використовуються: методи аналітичної геометрії, теорії ймовірності та математичної статистики.

Наукова новизна отриманих результатів полягає у наступному. Автором отриманий такий новий науковий результат: удосконалено математичну модель для побудови трансформованого еліпсу передбачення на основі застосування перетворення Джонсона, яка покращує моделювання області розподілу двовимірних негаусівських даних.

Практичне значення одержаних результатів полягає у розробці програми для побудови трансформованого еліпсу передбачення в системі Scilab.

1 МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ ПОБУДОВИ ТРАНСФОРМОВАНОГО ЕЛІПСУ ПЕРЕДБАЧЕННЯ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Існуючі методи та математичні моделі для побудови трансформованого еліпсу передбачення

При статистичному аналізі двовимірних даних використовується, як правило, еліпс [1, 2]. Завдяки еліпсу передбачення можна, наприклад, виявити двовимірні викиди (bivariate outliers) в даних, тому він часто застосовується для цього, зокрема в інформаційних технологіях. Статистичні методи, що базуються на еліпсі передбачення, використовують припущення, що дані апроксимуються двовимірним розподілом Гаусу. Але таке припущення може бути застосованим тільки в окремих випадках. Це призводить до необхідності побудови трансформованого еліпсу передбачення для конкретного набору даних, розподіл якого суттєво відрізняється від нормального [3]. Саме такими є, наприклад, трудомісткості розробки програмних проектів в залежності від кількості функціональних точок [4].

В роботі [3] був запропонований метод побудови трансформованого еліпсу передбачення для двовимірних негаусівських даних. Цей метод складається з наступних етапів. Спочатку двовимірні негаусівські дані нормалізують за допомогою бієктивного двовимірного нормалізуючого перетворення. Після цього отримують рівняння еліпсу передбачення для нормалізованих даних. На останньому етапі будують рівняння трансформованого еліпсу передбачення для двовимірних початкових даних на основі рівняння еліпсу передбачення для нормалізованих даних та зворотного перетворення. Але в [3] не була наведена математична модель для

побудови трансформованого еліпсу передбачення на останньому етапі, що призводить до необхідності подальшого її розвитку.

Як вказано вище, метод побудови трансформованого еліпсу передбачення [3] складається з трьох етапів.

На першому етапі вектор двовимірних негаусівських даних $\mathbf{X} = \{X_1, X_2\}^T$ нормалізують у двовимірний гаусівський вектор $\mathbf{Z} = \{Z_1, Z_2\}^T$ на основі перетворення

$$\mathbf{Z} = \psi(\mathbf{X}). \quad (1.1)$$

Перетворення (1.1) має зворотне перетворення

$$\mathbf{X} = \psi^{-1}(\mathbf{Z}). \quad (1.2)$$

На другому етапі будують рівняння еліпсу передбачення для нормалізованих даних

$$(\mathbf{Z} - \mathbf{m}_Z)^T \mathbf{S}^{-1} (\mathbf{Z} - \mathbf{m}_Z) = \frac{2(N^2 - 1)}{N(N - 2)} F_{2, N-2, \alpha}, \quad (1.3)$$

де $F_{2, N-2, \alpha}$ – квантиль F -розподілу; N – об'єм вибірки; \mathbf{m}_Z – вектор середніх значень, $\mathbf{m}_Z = (m_{Z_1}, m_{Z_2})^T$; α – рівень значимості; \mathbf{S} – матриця коваріації

$$\mathbf{S} = \begin{pmatrix} S_{Z_1}^2 & S_{Z_1 Z_2} \\ S_{Z_1 Z_2} & S_{Z_2}^2 \end{pmatrix}.$$

Оцінювання коваріаційної матриці здійснюється за формулою

$$S_N = \frac{1}{N} \sum_{i=1}^N (\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^T.$$

Ліву частину рівняння (1.3) можна представити у вигляді

$$\frac{S_{Z_1}^2 S_{Z_2}^2}{S_{Z_1}^2 S_{Z_2}^2 - S_{Z_1 Z_2}^2} \left[\frac{(Z_1 - m_{Z_1})^2}{S_{Z_1}^2} + \frac{(Z_2 - m_{Z_2})^2}{S_{Z_2}^2} - \frac{2S_{Z_1 Z_2} (Z_1 - m_{Z_1})(Z_2 - m_{Z_2})}{S_{Z_1}^2 S_{Z_2}^2} \right] =$$

$$= \frac{2(N^2 - 1)}{N(N - 2)} F_{2, N-2, \alpha} \quad (1.4)$$

На третьому етапі рівняння трансформованого еліпсу передбачення для двовимірних негаусівських даних будується на основі рівняння еліпсу передбачення (1.3) для нормалізованих даних та перетворення (1.1). З урахуванням (1.4) рівняння еліпсу передбачення має вигляд

$$\begin{aligned} \frac{[\psi_1(X_1) - m_{Z_1}]^2}{S_{Z_1}^2} + \frac{[\psi_2(X_2) - m_{Z_2}]^2}{S_{Z_2}^2} - \frac{2S_{Z_1 Z_2} [\psi_1(X_1) - m_{Z_1}] [\psi_2(X_2) - m_{Z_2}]}{S_{Z_1}^2 S_{Z_2}^2} = \\ = \frac{2(N^2 - 1)(S_{Z_1}^2 S_{Z_2}^2 - S_{Z_1 Z_2}^2)}{N(N - 2) S_{Z_1}^2 S_{Z_2}^2} F_{2, N-2, \alpha} \end{aligned} \quad (1.5)$$

На останньому етапі будують рівняння трансформованого еліпсу передбачення для двовимірних початкових даних на основі рівняння еліпсу передбачення для нормалізованих даних (1.3) або (1.5) та зворотного перетворення (1.2).

У якості нормалізуючого перетворення (1.1) в роботі [3] обрано двовимірне перетворення Джонсона, яке має вигляд

$$\mathbf{Z} = \boldsymbol{\gamma} + \boldsymbol{\eta} \mathbf{h}[\boldsymbol{\lambda}^{-1}(\mathbf{X} - \boldsymbol{\phi})] \sim N_m(\mathbf{0}_m, \mathbf{S}), \quad (1.6)$$

де $\boldsymbol{\gamma}$, $\boldsymbol{\eta}$, $\boldsymbol{\phi}$ та $\boldsymbol{\lambda}$ – параметри нормалізуючого перетворення Джонсона;

$$\boldsymbol{\gamma} = (\gamma_1, \gamma_2)^T; \quad \boldsymbol{\eta} = \text{diag}(\eta_1, \eta_2); \quad \boldsymbol{\phi} = (\phi_1, \phi_2)^T; \quad \boldsymbol{\lambda} = \text{diag}(\lambda_1, \lambda_2);$$

$$\mathbf{h}[(y_1, y_2)] = \{h_1(y_1), h_2(y_2)\}^T; \quad h_i(\cdot) \text{ – одна з функцій перетворення}$$

$$h = \begin{cases} \ln(y), & \text{for } S_L \text{ (log normal) family;} \\ \ln[y/(1 - y)], & \text{for } S_B \text{ (bounded) family;} \\ \text{Arsh}(y), & \text{for } S_U \text{ (unbounded) family;} \\ y & \text{for } S_N \text{ (normal) family.} \end{cases}$$

$$\text{Тут } y = (x - \phi)/\lambda; \quad \text{Arsh}(y) = \ln(y + \sqrt{y^2 + 1}).$$

1.2 Удосконалена математична модель для побудови трансформованого еліпсу передбачення на основі перетворення Джонсона

У разі застосування двовимірного перетворення Джонсона (1.6) для нормалізації негаусівських даних рівняння (1.4) можна записати як

$$Z_1^2 + Z_2^2 - 2S_{Z_1Z_2}Z_1Z_2 = \frac{2(N^2 - 1)(1 - S_{Z_1Z_2}^2)}{N(N - 2)}F_{2,N-2,\alpha} \quad (1.7)$$

Якщо в (1.7) змінну Z_1 розглядати як таку, значення якої задається, то тоді ми маємо алгебраїчне квадратичне рівняння відносно змінної Z_2 . Його рішення можна записати як

$$Z_{2,1,2} = S_{Z_1Z_2}Z_1 \pm \left[S_{Z_1Z_2}^2Z_1^2 + \frac{2(N^2 - 1)(1 - S_{Z_1Z_2}^2)}{N(N - 2)}F_{2,N-2,\alpha} - Z_1^2 \right]^{0,5} \quad (1.8)$$

Використовуючи до (1.8) зворотне перетворення (1.2) отримуємо математичну модель для побудови трансформованого еліпсу передбачення на основі перетворення Джонсона

$$X_{2,1,2} = \psi_2^{-1} \left\{ S_{Z_1Z_2}Z_1 \pm \left[S_{Z_1Z_2}^2Z_1^2 + \frac{2(N^2 - 1)(1 - S_{Z_1Z_2}^2)}{N(N - 2)}F_{2,N-2,\alpha} - Z_1^2 \right]^{0,5} \right\} \quad (1.9)$$

Зазначимо, що у (1.9) зміна значення Z_1 здійснюється у межах наступних двох значень:

$$Z_{1,1,2} = \pm \left[\frac{2(N^2 - 1)}{N(N - 2)}F_{2,N-2,\alpha} \right]^{0,5} \quad (1.10)$$

Двом значенням Z_1 з (1.10) відповідають два значенням X_1 , які знаходяться за наступною формулою:

$$X_{1,2} = \Psi_1^{-1} \left\{ \pm \left[\frac{2(N^2 - 1)}{N(N - 2)} F_{2, N-2, \alpha} \right]^{0,5} \right\}. \quad (1.11)$$

За допомогою (1.9)-(1.11) можна побудувати трансформований еліпс передбачення наступним чином.

Крок 1. Задаємо сім'ю та відповідні параметри перетворення Джонсона.

Крок 2. Задаємо рівень значущості. Зазвичай обирають значення 0,05.

Крок 3. Для заданого рівня значущості знаходимо квантиль F -розподілу $F_{2, N-2, \alpha}$.

Крок 4. За (1.10) знаходимо межі зміни значень Z_1 .

Крок 5. За (1.11) знаходимо межі зміни значень X_1 .

Крок 6. У визначених межах зміни значень Z_1 знаходимо за (1.9) значення X_2 .

У якості прикладу наведемо результати побудови трансформованого еліпсу передбачення для двовимірних негаусівських даних – трудомісткості виконання (у годинах) 133 програмних проєктів в залежності від розміру проєкту (у функціональних точках) з [4]. Було обрано нормалізуюче перетворення Джонсона сім'ї S_B . Трансформований еліпс передбачення (червона лінія) для цих даних для рівня значимості 0,05 наведений на рис.1.1.

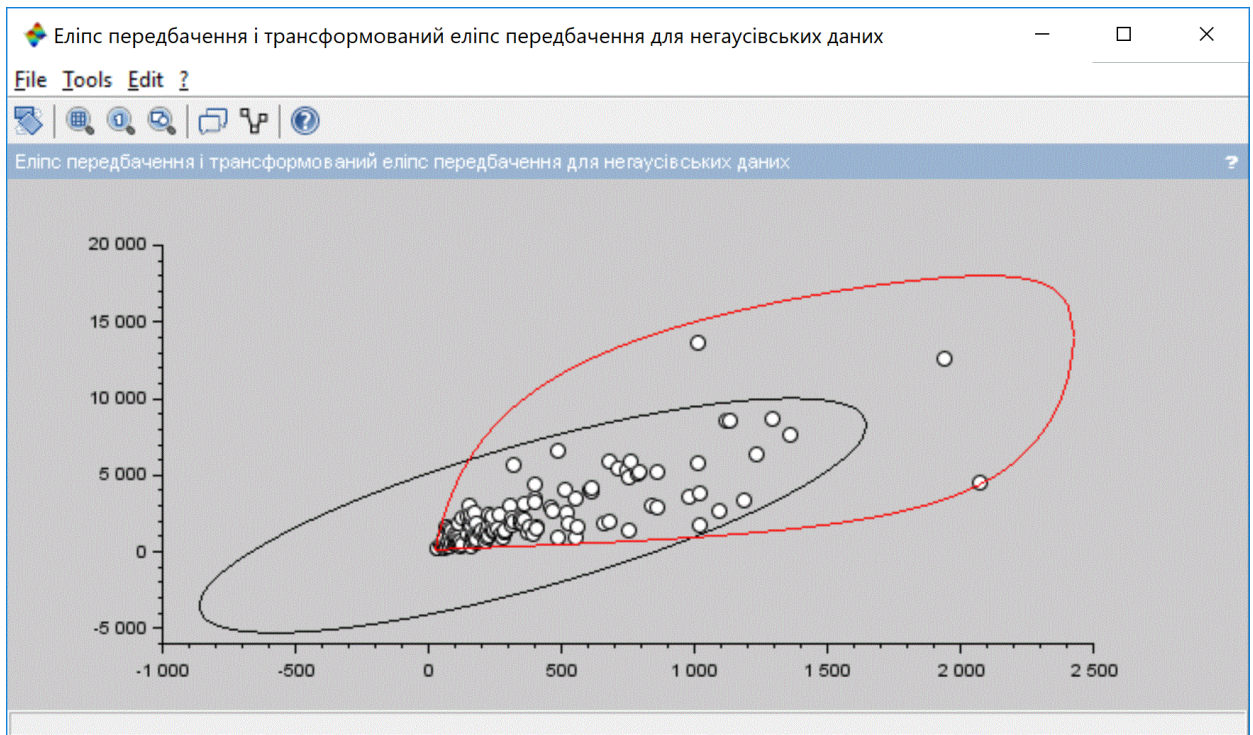


Рисунок 1.1 – Трансформований еліпс передбачення для двовимірних негаусівських даних

На рис.1.1 по осі абсцис відкладено кількість функціональних точок, а по осі ординат – трудомісткість виконання (у годинах) програмних проектів. На рис.1.1 також наведений еліпс передбачення (чорна лінія) для рівня значимості 0,005. Як можна побачити з рис.1.1, у порівнянні з еліпсом передбачення трансформований еліпс передбачення покращує моделювання області розподілу двовимірних негаусівських даних та дає викидів. Еліпс передбачення призводить до ідентифікації трьох викидів. Крім того, на відміну від еліпсу передбачення трансформований еліпс передбачення не має від'ємних абсцис та ординат. Отримані результати співпали з результатами, що наведені у [3, 5], що свідчить про адекватність запропонованої математичної моделі для побудови трансформованого еліпсу передбачення на основі перетворення Джонсона.

1.3 Засоби реалізації програми для побудови трансформованого еліпсу передбачення

Зазвичай сучасні програмні системи з обробки інформації є орієнтованими на кінцевого користувача, тому розробляються з зручним графічним інтерфейсом та отриманням (візуалізацією) результату обробки даних. Для реалізації програми для побудови трансформованого еліпсу передбачення для двомірних негаусівських даних необхідно розглянути проектні рішення з вибору мови програмування.

Тобто це повинна бути програма, яка буде надавати можливість вводити вхідні дані, а також в якій будуть реалізовані алгоритми обробки цих даних за допомогою обраних методів. Тому для розробки програмного забезпечення необхідно визначитись з мовою програмування, якою можливо реалізувати цю програму.

Для вирішення цього можна використовувати любі мови програмування, але оскільки програма виконує складні математичні обчислення, бажано щоб мова програмування та середовище розробки дозволяло легко працювати з ними. Цим умовам відповідає пакет прикладних програм для вирішення задач моделювання і обчислень MATLAB, з вбудованою однойменною мовою програмування та Scilab, з вбудованою Sci-мовою.

Мова MATLAB (<https://www.mathworks.com/>) є високорівневою інтерпретованою мовою програмування. Включає засновані на матрицях структури даних, широкий спектр функцій, інтегроване середовище розробки, об'єктно-орієнтовані можливості і інтерфейси до програм, написаних на інших мовах програмування. Переваги – багата вбудована

функціональність, працює на більшості ОС. Недоліки цього рішення – воно платне та потребує багато обчислювальних ресурсів.

Scilab (<http://www.scilab.org>) є програмним пакетом для наукових обчислень, що вільно розповсюджується. Він є певною альтернативою комерційним пакетам моделювання MATLAB/Simulink та MATRIXx/SystemBuild [6-8].

Scilab має схожу з MATLAB мову програмування – Sci-мову [6-8]. Sci-мова є високорівневою структурованою мовою програмування. Дозволяє працювати з елементарними і великим числом спеціальних функцій, має потужні засоби роботи з матрицями, поліномами, також проводити чисельні обчислення і рішення задач лінійної алгебри, оптимізації і моделювання, потужні статистичні функції, а також засіб для побудови та роботи з графіками.

Переваги – безкоштовне для комерційного використання, достатній вбудований функціонал, не потребує багато обчислювальних ресурсів, працює на більшості ОС. Для розробки програми для побудови трансформованого еліпсу передбачення для двомірних негаусівських даних недоліків не виявлено.

Отже будемо використовувати середовище Scilab для розробки програми для побудови трансформованого еліпсу передбачення для двомірних негаусівських даних.

Так як Scilab вже містить вбудоване середовище програмування та побудови інтерфейсів, то для складання програми за допомогою обраної мови не має потреби використовувати різноманітні середовища розробки, які дозволяють значно зменшити витрати часу програміста щодо проектування програмної системи, проектування інтерфейсу користувача, написання програмного коду програми.

Для реалізації віконного інтерфейсу користувача та математичних алгоритмів обробки інформації добре підійде ця мова програмування, тому що вона добре документована та широко розповсюджена для наукових обчислень, а також має дуже схожу з MATLAB мову програмування. Це є також важливим моментом, оскільки, як правило, розроблені програмне забезпечення необхідно підтримувати з часом, а іноді розробника ПЗ в момент необхідності внесення змін до програми знайти неможливо, тому виникає необхідність пошуку іншого програміста для доробки програми. Звичайно такого спеціаліста важче знайти, якщо ПЗ було розроблено маловідомими засобами.

Виходячи з цього можна зробити висновок, що ця Sci-мова програмування добре підійде для розробки програми для побудови трансформованого еліпсу передбачення для двомірних негаусівських даних.

1.3 Постановка задачі

Виходячи з аналізу існуючих методів та моделей для побудови трансформованого еліпсу передбачення для двомірних даних сформулюємо постановку задачі. Потрібно створити проект програми для побудови трансформованого еліпсу передбачення, яка би задовольняла таким вимогам.

Вимоги до програми.

Вимоги функціональних характеристик.

Вимоги до складу функцій, що виконуються.

Програма має виконувати такі функції:

- 1) Вводити з текстового файлу емпіричні дані.

2) Здійснювати вибір нормалізуючого перетворення Джонсона певної сім'ї.

3) Виконувати оцінювання параметрів нормалізуючого перетворення Джонсона.

4) Виконувати нормалізацію двомірних негаусівських даних;

5) Визначати значення координат точок еліпсу передбачення для двомірних нормалізованих даних та трансформованого еліпсу передбачення для двомірних негаусівських даних.

6) Виводити на екран значення координат точок еліпсу передбачення та нормалізованих двомірних негаусівських даних.

7) Виводити на екран значення координат точок трансформованого еліпсу передбачення та двомірних негаусівських даних.

8) Виводити у текстовий файл значення координат точок еліпсу передбачення та трансформованого еліпсу передбачення.

Вимоги до організації даних

Вхідні дані:

- значення двомірних негаусівських даних (емпіричні дані) вводяться з текстового файлу;

- початкові значення параметрів нормалізуючого перетворення Джонсона вводяться в інтерфейсне вікно;

- сім'я нормалізуючого перетворення Джонсона обирається шляхом переключення радіокнопок.

Вихідні дані:

- значення параметрів нормалізуючого перетворення Джонсона виводяться в інтерфейсне вікно;

- нормалізовані емпіричні дані виводяться в інтерфейсне вікно;

- значення координат точок еліпсу передбачення для нормалізованих двомірних негаусівських даних виводяться в інтерфейсне вікно та текстовий файл;

- значення координат точок трансформованого еліпсу передбачення для двомірних негаусівських даних виводяться в інтерфейсне вікно та текстовий файл.

Вимоги до складу й параметрів технічних засобів.

Програма має експлуатуватись на обладнанні з параметрами не нижчими за наступні: процесор Pentium IV (або еквівалентний) з інструкціями SSE2, оперативна пам'ять 2 Гб (1 Гб мінімум), простір на жорсткому диску 600 Мб мінімум.

Вимоги до інформаційної та програмної сумісності.

Для запуску та роботи з програмним забезпеченням необхідна операційна система Windows Vista/7/8/10 (32 або 64 біти), система моделювання Scilab 6.0.0., компілятор C (Visual Studio 2010 або Visual Express 2010) для компіляції зовнішніх модулів C або C++ та використання моделі в Xcos.

2 РОЗРОБКА ПРОЕКТУ ПРОГРАМИ ДЛЯ ПОБУДОВИ ТРАНСФОРМОВАНОГО ЕЛІПСУ ПЕРЕДБАЧЕННЯ

2.1 Розробка ескізного проекту програми для побудови трансформованого еліпсу передбачення

В ескізному проекті було розроблено зовнішню специфікацію програми для побудови трансформованого еліпсу передбачення та стани цієї програми при її виконанні.

Зовнішня специфікація програмного забезпечення.

Назва програмного забезпечення: tr_pred_el.

Функції, які виконує програма:

- 1) Вводити з текстового файлу емпіричні дані.
- 2) Здійснювати вибір нормалізуючого перетворення Джонсона певної сім'ї.
- 3) Виконувати оцінювання параметрів нормалізуючого перетворення Джонсона.
- 4) Виконувати нормалізацію двомірних негаусівських даних;
- 5) Визначати значення координат точок еліпсу передбачення для двомірних нормалізованих даних та трансформованого еліпсу передбачення для двомірних негаусівських даних.
- 6) Виводити на екран значення координат точок еліпсу передбачення та нормалізованих двомірних негаусівських даних.

7) Виводити на екран значення координат точок трансформованого еліпсу передбачення та двомірних негаусівських даних.

8) Виводити у текстовий файл значення координат точок еліпсу передбачення та трансформованого еліпсу передбачення.

Вхідні дані:

- значення двомірних негаусівських даних (емпіричні дані) вводяться з текстового файлу;

- початкові значення параметрів нормалізуючого перетворення Джонсона вводяться в інтерфейсне вікно;

- сім'я нормалізуючого перетворення Джонсона обирається шляхом переключення радіокнопок.

Вихідні дані:

- значення параметрів нормалізуючого перетворення Джонсона виводяться в інтерфейсне вікно;

- нормалізовані емпіричні дані виводяться в інтерфейсне вікно;

- значення координат точок еліпсу передбачення для нормалізованих двомірних негаусівських даних виводяться в інтерфейсне вікно та текстовий файл;

- значення координат точок трансформованого еліпсу передбачення для двомірних негаусівських даних виводяться в інтерфейсне вікно та текстовий файл.

Зовнішні ефекти: поява графічного вікна для вводу даних, управління виводом даних, оцінювання параметрів нормалізуючого перетворення Джонсона, нормалізації негаусівських даних, побудови еліпсу передбачення для нормалізованих двомірних негаусівських даних, побудови трансформованого еліпсу передбачення для двомірних негаусівських даних, виходу з програми (закриття всіх графічних вікон).

Для проектування інтерфейсу програми для побудови трансформованого еліпсу передбачення для двовірних даних представимо стани програми при виконанні у вигляді відповідної діаграми станів. Діаграма станів програми для побудови трансформованого еліпсу передбачення для двовірних даних наведена на рис. 2.1.

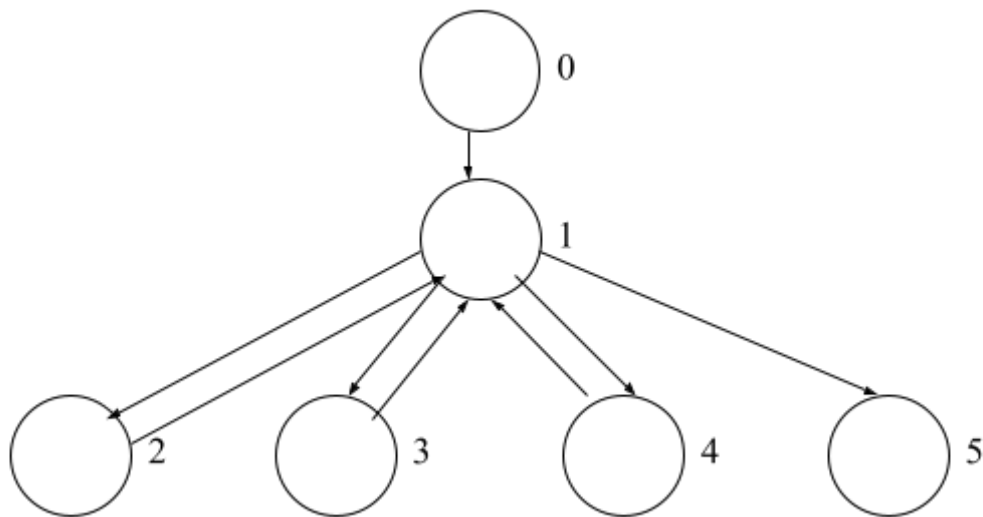


Рисунок 2.1 – Діаграма станів програми для побудови трансформованого еліпсу передбачення для двовірних негаусівських даних

На діаграмі станів програмного забезпечення введені такі позначення станів: 0 – початковий стан, 1 – графічне вікно головного модуля, 2 – графічне вікно візуалізації введених двовірних негаусівських даних (емпіричних даних), 3 – графічне вікно із результатами знаходження трансформованого еліпсу передбачення для двовірних негаусівських даних, 4 – графічні вікна з результатами побудови еліпсу передбачення для нормалізованих двовірних негаусівських даних та трансформованого еліпсу передбачення для двовірних негаусівських даних, 5 – кінцевий стан.

2.2 Розробка технічного проекту програми для побудови трансформованого еліпсу передбачення

В технічному проєкті було спроектовано програму для побудови трансформованого еліпсу передбачення на основі покрокової деталізації зверху вниз, розроблено специфікації модулів програми, побудовано алгоритми модулів програми.

Програма складається з таких модулів: головного модуля (`tr_pred_el`), модуля для введення двомірних негаусівських даних (емпіричних даних) з файлу та їх візуалізації (`DataInput`), модуля для побудови трансформованого еліпсу передбачення для двомірних негаусівських даних (`Pred_el`), модуля для виводу графіків еліпсу передбачення для нормалізованих двомірних негаусівських даних та трансформованого еліпсу передбачення для двомірних негаусівських даних (`Outputy`), модуля для закриття графічних вікон (`_Close`). Схема взаємодії модулів програми наведена на рис. 2.2.

Специфікація головного модуля.

Назва модуля: `tr_pred_el`.

Функції, які виконує програма:

- 1) Відкриття графічного вікна головного модуля.
- 2) Введення імені файлу з емпіричними даними.
- 3) Виклик функції для введення емпіричних даних з файлу та їх візуалізації у графічному вікні.
- 4) Вибір нормалізуючого перетворення Джонсона певної сім'ї.
- 5) Виклик функції для побудови трансформованого еліпсу передбачення для двомірних негаусівських даних.
- 6) Виклик функції для виводу графіків еліпсу передбачення для нормалізованих двомірних негаусівських даних та трансформованого еліпсу передбачення.

7) Виклик функції для закриття графічних вікон.

Вхідні дані:

- значення двомірних негаусівських даних (емпіричні дані);
- сім'я нормалізуючого перетворення Джонсона;
- значення параметрів нормалізуючого перетворення Джонсона.

Вихідні дані:

- значення параметрів нормалізуючого перетворення Джонсона.
- нормалізовані емпіричні дані;
- значення координат точок еліпсу передбачення.

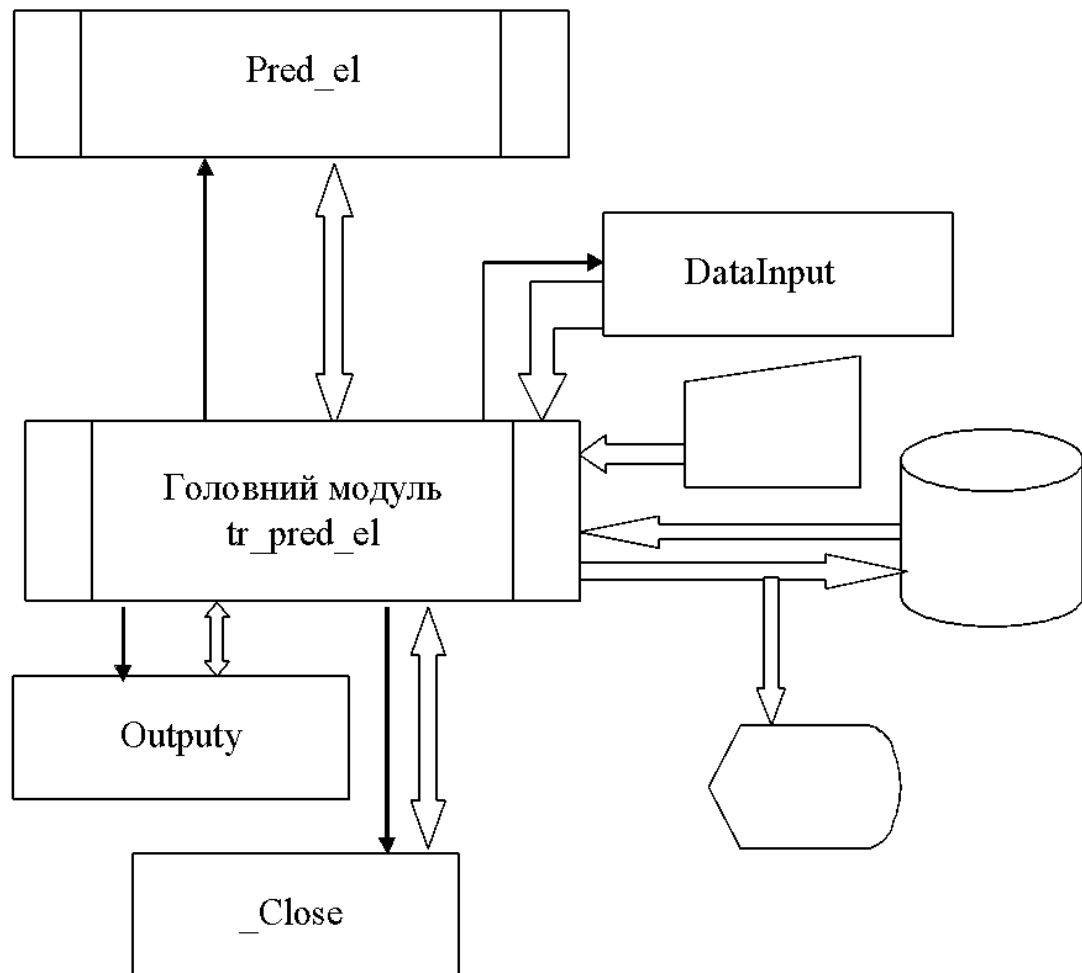


Рисунок 2.2 – Схема взаємодії модулів програми для побудови трансформованого еліпсу передбачення

Зовнішні ефекти: поява графічного вікна для вводу даних, управління виводом даних, оцінювання параметрів нормалізуючого перетворення Джонсона, нормалізації двовірних негаусівських даних, побудова

трансформованого еліпсу передбачення для двомірних негаусівських даних, вихід з програми (закриття всіх графічних вікон).

Специфікація модуля для введення двомірних негаусівських даних з файлу та їх візуалізації.

Назва модуля: DataInput.

Функції, які виконує модуль:

1) Введення двомірних негаусівських даних (значень випадкових величини) з файлу.

2) Формування одномірних масивів значень випадкових величини (двомірних негаусівських даних).

3) Відкриття графічного вікна для візуалізації введених двомірних негаусівських даних.

4) Візуалізацію введених двомірних негаусівських даних.

Вхідні дані:

- значення двомірних негаусівських даних (емпіричні дані).

Вихідні дані:

- одномірні масиви значень двомірних негаусівських даних (емпіричні дані);

- кількість значень двомірних негаусівських даних;

- ідентифікатор графічного вікна.

Зовнішні ефекти: Поява графічного вікна для візуалізації введених двомірних негаусівських даних.

Специфікація модуля для побудови трансформованого еліпсу передбачення для двомірних негаусівських даних.

Назва модуля: pred_el.

Функції, які виконує модуль:

1) Нормалізація двомірних негаусівських даних.

2) Визначення координат точок трансформованого еліпсу передбачення для двомірних негаусівських даних.

Вхідні дані:

- одномірні масиви значень двомірних негаусівських даних (емпіричні дані);
- кількість значень двомірних негаусівських даних.

Вихідні дані:

- одномірні масиви координат точок еліпсу передбачення для нормалізованих двомірних негаусівських даних;
- одномірні масиви координат точок трансформованого еліпсу передбачення для двомірних негаусівських даних;
- ідентифікатори графічних вікон, в яких здійснювався вивід координат точок еліпсу передбачення для нормалізованих двомірних негаусівських даних та трансформованого еліпсу передбачення для двомірних негаусівських даних.

Зовнішні ефекти: Поява графічних вікон для виводу координат точок еліпсу передбачення для нормалізованих двомірних негаусівських даних та трансформованого еліпсу передбачення для двомірних негаусівських даних.

Специфікація модуля для закриття графічних вікон.

Назва модуля: _Close.

Функції, які виконує модуль:

Закриття графічних вікон.

Вхідні дані:

- ідентифікатори графічних вікон.

Вихідні дані: відсутні.

Зовнішні ефекти: закриття графічних вікон.

В подальшому були розроблені алгоритми зазначених модулів. При подальшій покроковій деталізації зверху вниз алгоритми зазначених модулів можна записати scі-мовою.

2.3 Розробка робочого проекту програми для побудови трансформованого еліпсу передбачення

В робочому проєкті було розроблено програму для побудови трансформованого еліпсу передбачення, здійснено тестування програми, здійснено випробування програми.

Спочатку scі-мовою системи моделювання Scilab 6.0.0 було виконано кодування алгоритмів, що були розроблені на стадії технічного проєкту. Текст програми наведений в додатку А. Приклад кодування модуля Pred_el, що здійснює побудову трансформованого еліпсу передбачення для двомірних негаусівських даних, scі-мовою в системі моделювання Scilab 6.0.0 наведено нижче.

```
//Функція побудови трансформованого еліпсу передбачення
//для двомірних негаусівських даних.
function [y_pezy_min, y_pezy_max, xi_z, pezy_min, pezy_max, zxi, zx,
zy]=pred_el(x, y, n)

//Считываем значение переменных из текстовых полей и
//преобразовываем их к числовому типу.
ax=eval(get(edit_ax, 'string'));
bx=eval(get(edit_bx, 'string'));
cx=eval(get(edit_cx, 'string'));
dx=eval(get(edit_dx, 'string'));
ay=eval(get(edit_ay, 'string'));
by=eval(get(edit_by, 'string'));
cy=eval(get(edit_cy, 'string'));
dy=eval(get(edit_dy, 'string'));
```

```

//Проверяем значение флажка, если флажок выключен,
if (get (radio_bikv_x, 'value')==0) & (get (radio_bikv_y, 'value')==0)
    zxA=0;
    zyA=0;
    for i=1:n
        zx(i)=ax+bx*log((x(i)-cx)/(dx-x(i)+cx));
        zy(i)=ay+by*log((y(i)-cy)/(dy-y(i)+cy));
        zxA=zxA+zx(i);
        zyA=zyA+zy(i);
    end;
    zxA=zxA/n;
    zyA=zyA/n;
    S2zx=0;
    S2zy=0;
    szxzy=0;
    for i=1:n
        S2zx=S2zx+(zx(i)-zxA)^2;
        S2zy=S2zy+(zy(i)-zyA)^2;
        szxzy=szxzy+(zx(i)-zxA)*(zy(i)-zyA);
    end;
    S2zx=S2zx/n;
    S2zy=S2zy/n;
    szxzy=szxzy/n;

    Dfn=2;
    Dfd=n-2;
    Q=0.005;
    P=1-Q;
    [F]=cdf('F', Dfn, Dfd, P, Q);
    test_st=2*(n*n-1)*F/n/(n-2);

    zxmin=zxA-(S2zx*test_st)^0.5;
    zxmax=zxA+(S2zx*test_st)^0.5;
    step=(zxmax-zxmin)/(n-1);
    //    step=(zxmax-zxmin)/n;
    cS=(S2zx*S2zy-szxzy^2)/S2zx/S2zy;
    a=1/S2zy;
    for i=1:n
        zxi(i)=zxmin+step*(i-1);
    //    zxi(i)=zxmin+step*i;
        c=cS*test_st-(zxi(i)-zxA)^2/S2zx;
        b=2*szxzy*(zxi(i)-zxA)/S2zx/S2zy;
        dis=b*b+4*a*c;
        sqdis=dis^0.5;
        pezy_min(i)=zyA+(b-sqdis)/2/a;
        pezy_max(i)=zyA+(b+sqdis)/2/a;

    end;

    for i=1:n
        xi_z(i)=cx+dx/(1+exp(-(zxi(i)-ax)/bx));
        y_pezy_min(i)=cy+dy/(1+exp(-(pezy_min(i)-ay)/by));
        y_pezy_max(i)=cy+dy/(1+exp(-(pezy_max(i)-ay)/by));
    end;

```

```

end;

if dis>0
    set(textresult,'string',...
        sprintf("Еліпс передбачення знайдено\Det=%1.4f",dis));
else
    set(textresult,'string',...
        sprintf("Еліпс передбачення не знайдено\Det=%1.4f",dis));

end;

else

    set(textresult,'string','Еліпс передбачення не знайдено');

end;

endfunction

```

Далі в середовищі Scilab 6.0.0 було створено sce-файл розміром 10 КБ. Зазначимо, що в програмі використовувався динамічний спосіб створення інтерфейсних компонентів. Він полягає у тому, що при виконанні програми можуть створюватися і знищуватися ті чи інші елементи управління (командні кнопки, радіокнопки та таке інше), а їх властивостям присвоюються відповідні значення. В програмі були використані такі інтерфейсні компоненти: вікно редагування, командні кнопки та радіокнопки. Графічне вікно програми із зазначеними інтерфейсними компонентами наведено на рис.2.3.

Було проведено тестування окремих модулів. У якості прикладу наведемо результати тестування для модуля побудови еліпсу передбачення для двомірних даних – трудомісткості виконання (у годинах) 133 програмних проектів в залежності від розміру проекту (у функціональних точках) з [10]. Було обрано нормалізуюче перетворення Джонсона сім'ї S_B .

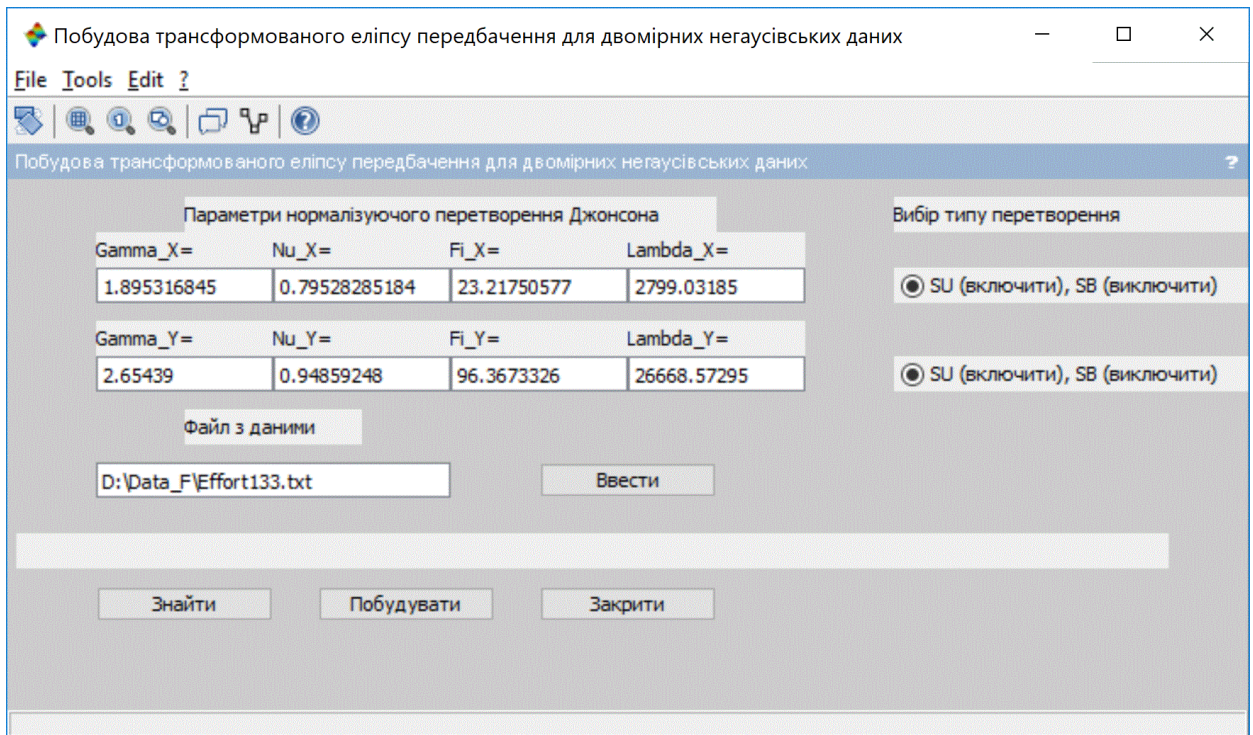


Рисунок 2.3. – Графічне вікно програмного забезпечення з інтерфейсними компонентами

Результати візуалізації трудомісткості виконання (у годинах) 133 програмних проектів в залежності від розміру проекту (у функціональних точках) – двомірні негаусівські дані (емпіричні дані) наведені на рис.2.4.

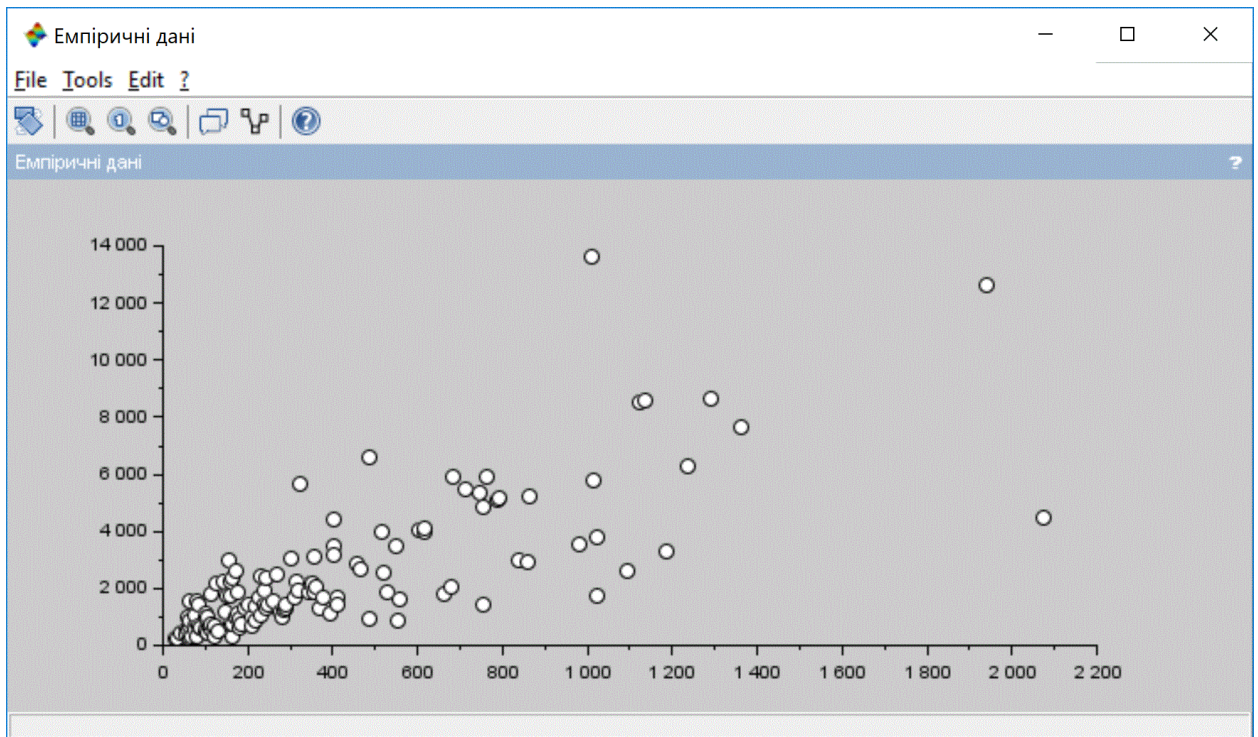


Рисунок 2.4. – Результати візуалізації трудомісткості виконання (у годинах) 133 програмних проектів в залежності від розміру проекту (у функціональних точках)

Було проведено тестування помилкового вибору сім'ї перетворення Джонсона. Результати роботи програми для цього випадку наведено на рис.2.5: видається повідомлення «Еліпс передбачення не знайдено».

Після правильного вибору сім'ї перетворення Джонсона: видається повідомлення «Еліпс передбачення не знайдено». Результати роботи програми для цього випадку наведено на рис.2.6.

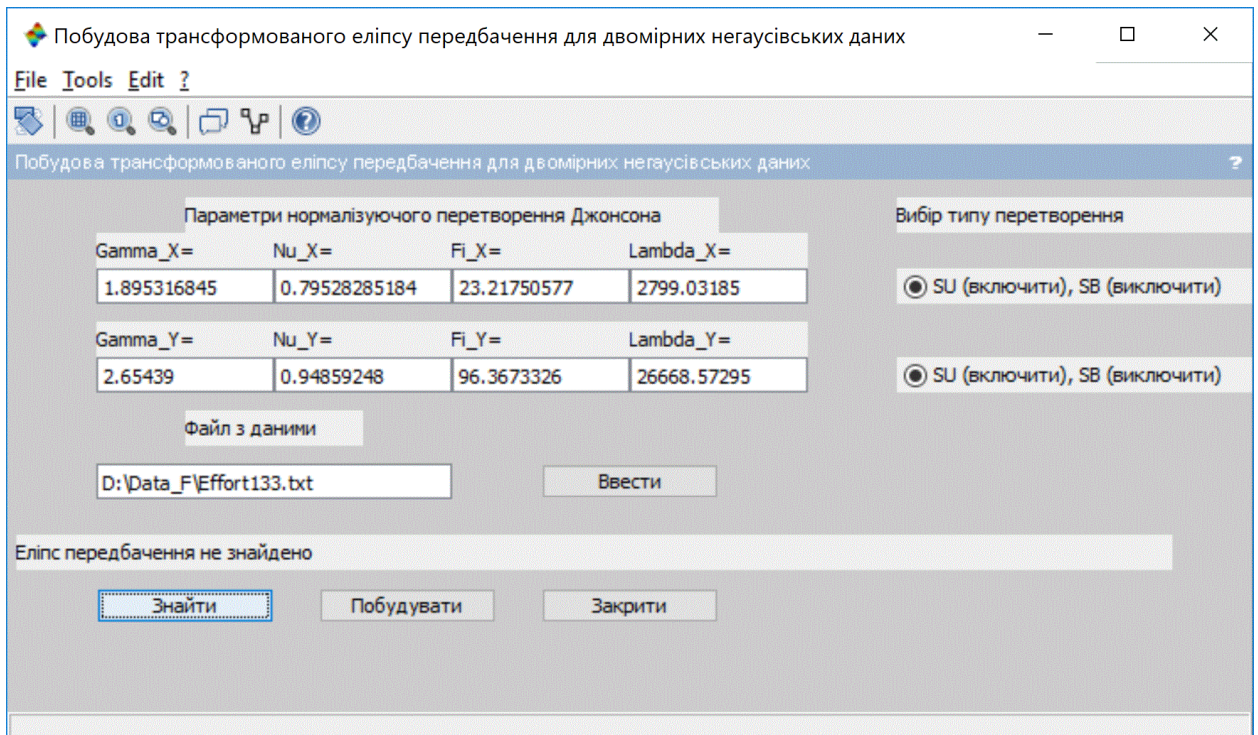


Рисунок 2.5. – Результати роботи програми у разі помилкового вибору сім'ї перетворення Джонсона

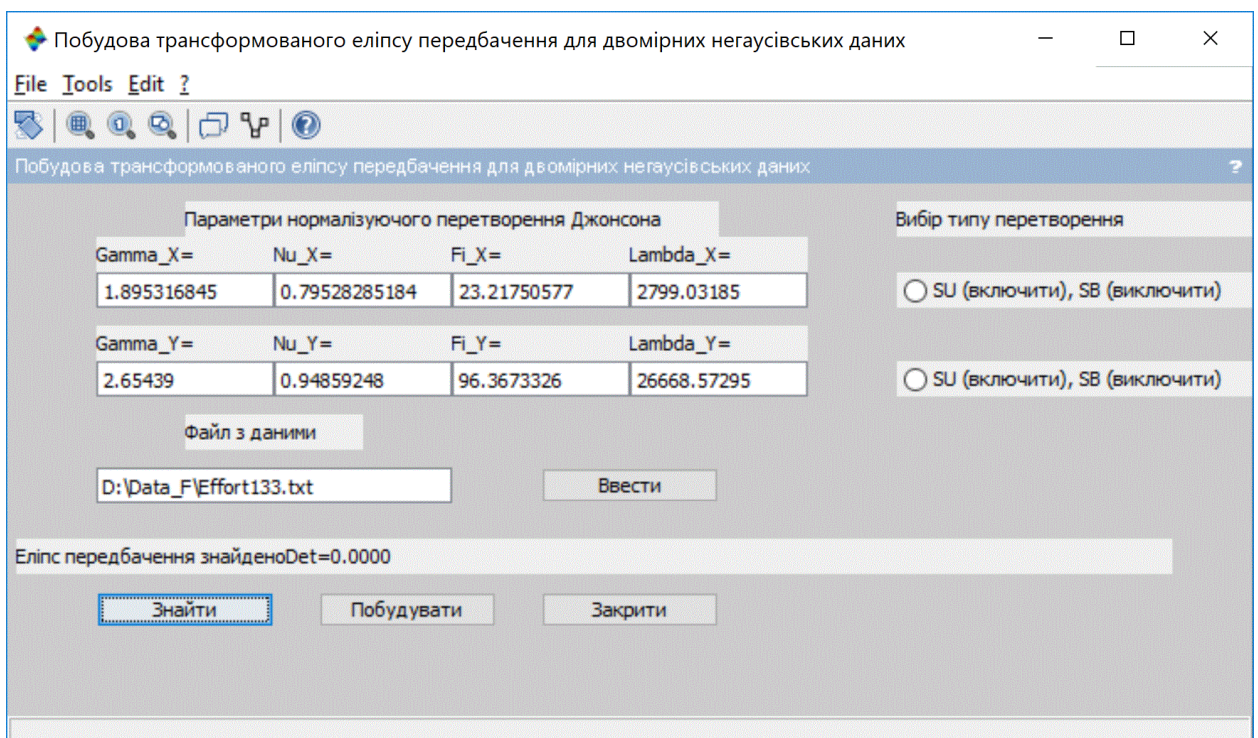


Рисунок 2.6. – Результати роботи програми у разі правильного вибору сім'ї перетворення Джонсона

Результати побудови еліпсу передбачення для нормалізованих двовірних негаусівських даних наведені на рис.2.7., результати побудови трансформованого еліпсу передбачення для двовірних негаусівських даних – на рис.2.8.

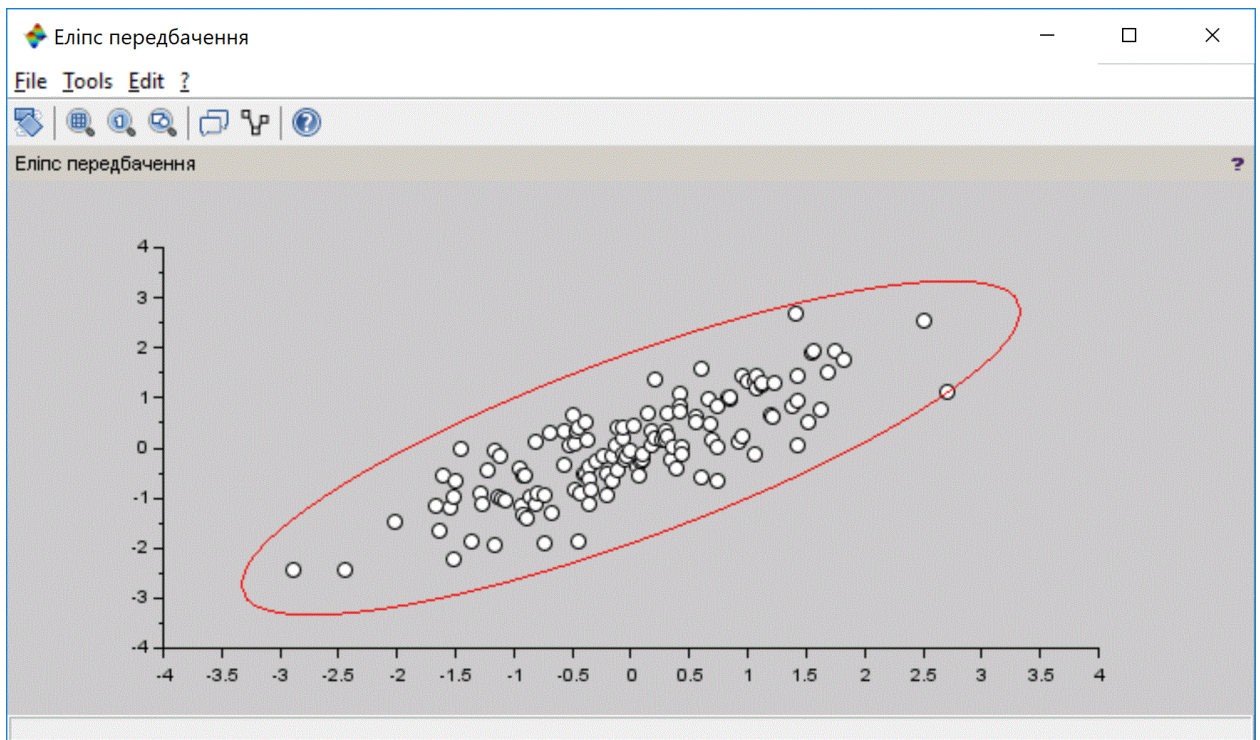


Рисунок 2.7. – Результати побудови еліпсу передбачення для нормалізованих двовимірних даних з трудомісткості виконання 133 програмних проектів в залежності від розміру проекту

На рис.2.7 також наведені нормалізовані емпіричні дані з трудомісткості виконання 133 програмних проектів в залежності від розміру проекту.

На рис.2.8 також наведені емпіричні дані з трудомісткості виконання (у годинах) 133 програмних проектів в залежності від розміру проекту (у функціональних точках).

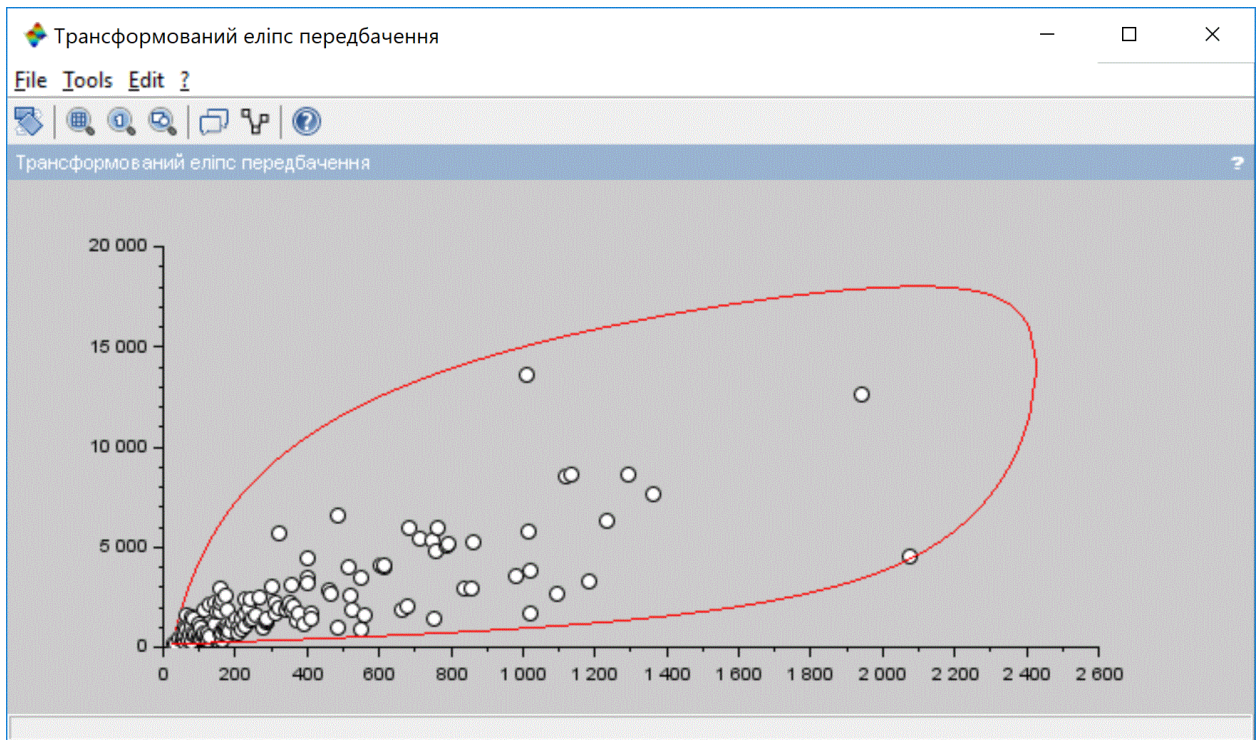


Рисунок 2.8. – Результати побудови трансформованого еліпсу передбачення для двовимірних даних з трудомісткості виконання (у годинах) 133 програмних проектів в залежності від розміру проекту (у функціональних точках)

Отримані результати побудови трансформованого еліпсу передбачення для двовимірних негаусівських даних з трудомісткості виконання 133 програмних проектів в залежності від розміру проекту практично співпали з результатами, що наведені в роботах [3, 5]. Це свідчить про працездатність розробленої програми для побудови трансформованого еліпсу передбачення для двовимірних негаусівських даних.

ВИСНОВКИ

У науковій роботі удосконалено математичну модель для побудови трансформованого еліпсу передбачення на основі застосування перетворення Джонсона, яка покращує моделювання області розподілу двовимірних негаусівських даних.

Отримано, що при застосуванні еліпсу передбачення для визначення наявності викидів у двовимірних негаусівських даних мають місце помилкові результати. Крім того, дуже часто ліва частина еліпсу передбачення для двовимірних негаусівських даних опиняється в області, де реальні дані не існують. Це призводить до необхідності використання трансформованого еліпсу передбачення.

В роботі також було створено програму для побудови трансформованого еліпсу передбачення для двовимірних негаусівських даних на основі двовимірного нормалізуючого перетворення Джонсону для системи моделювання Scilab 6.0.0. За результатами випробувань зроблено висновок про працездатність розробленої програми для побудови трансформованого еліпсу передбачення для двовимірних негаусівських даних для системи моделювання Scilab 6.0.0.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Friendly, M. Elliptical Insights: Understanding Statistical Methods Through Elliptical Geometry [Text] / Michael Friendly, Georges Monette, John Fox // Statistical Science. – 2013. – Vol. 28, No. 1. – p. 1–39.
2. Chandola, V. Anomaly Detection: A Survey [Text] / V. Chandola, A. Banerjee, V. Kumar // Technical Report 07-017, 2007. – 72 p.
3. Prykhodko, S.B. Constructing the transformed prediction ellipses on the basis of normalizing transformations for bivariate non-Gaussian data [Text] / S. B. Prykhodko, N. V. Prykhodko, O. O. Kudin, T. G. Smykodub // Проблемы информационных технологий. – 2017. – № 1 (021). – С.134-138. – ISSN 1998-7005
4. Kitchenham, B. An empirical study of maintenance and development estimation accuracy [Text] / B. Kitchenham, S.L. Pfleeger, B. McColl, S. Eagan // The Journal of Systems and Software. – 2002. – 64. – p.57-77.
5. Prykhodko S. Detecting bivariate outliers on the basis of normalizing transformations for non-Gaussian data / S. Prykhodko, N. Prykhodko, L. Makarova, O. Kudin, T. Smykodub, A. Prykhodko // Advanced Information Systems and Technologies: proceedings of the V international scientific conference, Sumy, May 17-19 2017 / Edited by S. I. Protsenko, V. V. Shendryk. – Sumy: Sumy State University, 2017. – p.95-97.
6. Campbell, S.L. Modeling and Simulation in Scilab/Scicos [Text] / Stephen L. Campbell, Jean-Philippe Chancelier, Ramine Nikoukhah. – Springer, 2005. – 313 p.

7. Алексеев, Е.Р. Scilab: Решение инженерных и математических задач [Текст] / Е. Р. Алексеев, О. В. Чеснокова, Е. А. Рудченко. – М.: ALT Linux; БИНОМ. Лаборатория знаний, 2008. – 269 с.

8. Scilab for very beginners [E-resource]. – Access mode: https://www.scilab.org/sites/default/files/Scilab_beginners_0.pdf

ДОДАТОК А. Текст програми

```
//Побудова трансформованого еліпсу передбачення для двовірних
негаусівських даних
f=figure();//Создание графического окна.
//Устанавливаем размеры окна.
set(f,'position',[0,0,700,300])
//Устанавливаем заголовок окна.
set(f,'figure_name','Побудова трансформованого еліпсу передбачення
для двовірних негаусівських даних');

//plot2d(x,y,-9);

lab_text0=uicontrol(f,'style','text','string',...
'Файл з даними','position',...
[100,150,100,20]);
edit_text=uicontrol(f,'style','edit','string',...
'D:\Data_F\Effort133.txt','position',[50,120,200,20]);

BtDatInp=uicontrol('style','pushbutton','string','Ввести',...
'CallBack','[x,y,f1,n]=DataInput()','position',[300,120,100,20]);

lab_text1=uicontrol(f,'style','text','string',...
'Параметри нормалізуючого перетворення Джонсона','position',...
[100,270,300,20]);
lab_text2=uicontrol(f,'style','text','string',...
'Вибір типу перетворення','position',...
[500,270,200,20]);
//Создание текстовых полей для подписей полей ввода коэффициентов.
//Для X
//Подпись Gamma_X=.
lab_ax=uicontrol(f,'style','text','string','Gamma_X=','position',...
[50,250,100,20]);
//Подпись Nu_X=.
lab_bx=uicontrol(f,'style','text','string','Nu_X=','position',...
[150,250,100,20]);
//Подпись Fi_X=.
lab_cx=uicontrol(f,'style','text','string','Fi_X=','position',...
[250,250,100,20]);
```

```

//Подпись Lambda_X=.
lab_dx=uicontrol(f,'style','text','string','Lambda_X','position',...
[350,250,100,20]);
//Для Y
//Подпись Gamma_Y=.
lab_ay=uicontrol(f,'style','text','string','Gamma_Y','position',...
[50,200,100,20]);
//Подпись Nu_Y=.
lab_by=uicontrol(f,'style','text','string','Nu_Y','position',...
[150,200,100,20]);
//Подпись Fi_Y=.
lab_cy=uicontrol(f,'style','text','string','Fi_Y','position',...
[250,200,100,20]);
//Подпись Lambda_Y=.
lab_dy=uicontrol(f,'style','text','string','Lambda_Y','position',...
[350,200,100,20]);
//Поле редактирования для ввода коэффициента a.
edit_ax=uicontrol(f,'style','edit','string','1.895316845','position',
...
[50,230,100,20]);
//Поле редактирования для ввода коэффициента b.
edit_bx=uicontrol(f,'style','edit','string','0.79528285184','position
',...
[150,230,100,20]);
//Поле редактирования для ввода коэффициента c.
edit_cx=uicontrol(f,'style','edit','string','23.21750577','position',
...
[250,230,100,20]);
//Поле редактирования для ввода коэффициента d.
edit_dx=uicontrol(f,'style','edit','string','2799.03185','position',..
..
[350,230,100,20]);
//
edit_ay=uicontrol(f,'style','edit','string','2.65439','position',...
[50,180,100,20]);
//Поле редактирования для ввода коэффициента b.
edit_by=uicontrol(f,'style','edit','string','0.94859248','position',..
..
[150,180,100,20]);
//Поле редактирования для ввода коэффициента c.
edit_cy=uicontrol(f,'style','edit','string','96.3673326','position',..
..
[250,180,100,20]);
//Поле редактирования для ввода коэффициента d.
edit_dy=uicontrol(f,'style','edit','string','26668.57295','position',
...
[350,180,100,20]);
//
//Текстовое поле, определяющее вывод результатов.
textresult=uicontrol(f,'style','text','string','', 'position',...
[5,80,650,20]);
//Флажок, отвечающий за выбор типа преобразования.
radio_bikv_x=uicontrol('style','radiobutton','string',...

```

```

'SU (включити), SB (виключити)', 'value', 1, 'position', ...
[500, 230, 300, 20]);
//
radio_bikv_y=icontrol('style','radiobutton','string',...
'SU (включити), SB (виключити)', 'value', 1, 'position', ...
[500, 180, 300, 20]);
//

BtNr_pred_int=icontrol('style','pushbutton','string','Знайти','CallB
ack',...
'[y_pezy_min,y_pezy_max,xi_z,pezy_min,pezy_max,zxi,zx,zy]=pred_el(x,y
,n)',...
'position',[50,50,100,20]);

BtOutput=icontrol('style','pushbutton','string','Побудувати','CallBa
ck',...
'[f2,f3]=Output_yr(y_pezy_min,y_pezy_max,xi_z,x,y,pezy_min,pezy_max,z
xi,zx,zy)',...
'position',[175,50,100,20]);

BtClose=icontrol('style','pushbutton','string','Закрити',...
'CallBack','_Close()', 'position',[300,50,100,20]);

function [x, y, f1, n]=DataInput()
dir_name=get(edit_text,'string');

file_PC=mopen(dir_name,'r');
file_PC2=mopen('D:\Data_F\Effort133_out.txt','w+');
i=1;
//n=52;
while (meof(file_PC))<1
//for i=1:n
    x(i)=mfscanf(file_PC,'%f'); y(i)=mfscanf(file_PC,'%f');
    i=i+1;
end;
n=i-1;
for i=1:n
    fprintf(file_PC2,'%f\t%f\n',x,y);
    // fprintf(file_PC2,'%d\n',x);
end;
fclose(file_PC);
fclose(file_PC2);
f1=figure(); //Создание графического окна.
//Устанавливаем размеры окна.
set(f1,'position',[50,50,700,300]);
//Устанавливаем заголовок окна.
set(f1,'figure_name','Емпіричні дані');
//plot(x,y,"*r");
plot2d(x,y,-9);
endfunction

//Функція побудови трансформованого еліпсу передбачення
//для двовірних негаусівських даних.

```

```

function [y_pezy_min, y_pezy_max, xi_z, pezy_min, pezy_max, zxi, zx,
zy]=pred_el(x, y, n)

//Считываем значение переменных из текстовых полей и
//преобразовываем их к числовому типу.
ax=eval(get(edit_ax,'string'));
bx=eval(get(edit_bx,'string'));
cx=eval(get(edit_cx,'string'));
dx=eval(get(edit_dx,'string'));
ay=eval(get(edit_ay,'string'));
by=eval(get(edit_by,'string'));
cy=eval(get(edit_cy,'string'));
dy=eval(get(edit_dy,'string'));
//Проверяем значение флажка, если флажок выключен,
if (get radio_bikv_x, 'value')==0 & (get radio_bikv_y, 'value')==0)

    xA=0;
    yA=0;
    for i=1:n
        xA=xA+x(i);
        yA=yA+y(i);
    end;
    xA=xA/n;
    yA=yA/n;
    S2x=0;
    S2y=0;
    sxy=0;
    for i=1:n
        S2x=S2x+(x(i)-xA)^2;
        S2y=S2y+(y(i)-yA)^2;
        sxy=sxy+(x(i)-xA)*(y(i)-yA);
    end;
    S2x=S2x/n;
    S2y=S2y/n;
    sxy=sxy/n;

    Dfn=2;
    Dfd=n-2;
    Q=0.05;
    P=1-Q;
    [F]=cdf('F',Dfn,Dfd,P,Q);
    test_st=2*(n*(n-1)*F/n/(n-2);

    xmin=xA-(S2x*test_st)^0.5;
    xmax=xA+(S2x*test_st)^0.5;
    step=(xmax-xmin)/(n-1);
    //    step=(zxmax-zxmin)/n;
    cS=(S2x*S2y-sxy^2)/S2x/S2y;
    a=1/S2y;
    for i=1:n
        xi(i)=xmin+step*(i-1);
    //    zxi(i)=zxmin+step*i;
        c=cS*test_st-(xi(i)-xA)^2/S2x;

```

```

    b=2*sxy*(xi(i)-xA)/S2x/S2y;
    dis=b*b+4*a*c;
    sqdis=dis^0.5;
    pey_min(i)=yA+(b-sqdis)/2/a;
    pey_max(i)=yA+(b+sqdis)/2/a;

end;

zxA=0;
zyA=0;
for i=1:n
    zx(i)=ax+bx*log((x(i)-cx)/(dx-x(i)+cx));
    zy(i)=ay+by*log((y(i)-cy)/(dy-y(i)+cy));
    zxA=zxA+zx(i);
    zyA=zyA+zy(i);
end;
zxA=zxA/n;
zyA=zyA/n;
S2zx=0;
S2zy=0;
szxzy=0;
for i=1:n
    S2zx=S2zx+(zx(i)-zxA)^2;
    S2zy=S2zy+(zy(i)-zyA)^2;
    szxzy=szxzy+(zx(i)-zxA)*(zy(i)-zyA);
end;
S2zx=S2zx/n;
S2zy=S2zy/n;
szxzy=szxzy/n;

Dfn=2;
Dfd=n-2;
Q=0.005;
P=1-Q;
[F]=cdf('F',Dfn,Dfd,P,Q);
test_st=2*(n*n-1)*F/n/(n-2);

zxmin=zxA-(S2zx*test_st)^0.5;
zxmax=zxA+(S2zx*test_st)^0.5;
step=(zxmax-zxmin)/(n-1);
// step=(zxmax-zxmin)/n;
cS=(S2zx*S2zy-szxzy^2)/S2zx/S2zy;
a=1/S2zy;
for i=1:n
    zxi(i)=zxmin+step*(i-1);
// zxi(i)=zxmin+step*i;
c=cS*test_st-(zxi(i)-zxA)^2/S2zx;
b=2*szxzy*(zxi(i)-zxA)/S2zx/S2zy;
dis=b*b+4*a*c;
sqdis=dis^0.5;
pezy_min(i)=zyA+(b-sqdis)/2/a;
pezy_max(i)=zyA+(b+sqdis)/2/a;

```



```

end;

for i=1:n
    xi_z(i)=cx+dx/(1+exp(-(zxi(i)-ax)/bx));
    y_pezy_min(i)=cy+dy/(1+exp(-(pezy_min(i)-ay)/by));
    y_pezy_max(i)=cy+dy/(1+exp(-(pezy_max(i)-ay)/by));

end;

//
//f_2=figure();//Создание графического окна.
//Устанавливаем размеры окна.
//set(f_2,'position',[100,100,700,300]);
//Устанавливаем заголовок окна.
//set(f_2,'figure_name',...
//'Еліпс передбачення і трансформований еліпс передбачення для
негаусівських даних');

//plot2d(xi,pey_max,1);
//plot2d(x,y,-9);

//plot2d(xi,pey_min,1);

//plot2d(xi_z,y_pezy_max,5);

//plot2d(xi_z,y_pezy_min,5);
//

if dis>0
    set(textresult,'string',...
        sprintf("Еліпс передбачення знайдено\Det=%1.4f",dis));
else
    set(textresult,'string',...
        sprintf("Еліпс передбачення не знайдено\Det=%1.4f",dis));

end;

else

    set(textresult,'string','Еліпс передбачення не знайдено');

end;

endfunction

function Outputy(x, y)
f1=figure();//Создание графического окна.
//Устанавливаем размеры окна.
set(f1,'position',[50,50,700,300]);
//Устанавливаем заголовок окна.
set(f1,'figure_name','Емпіричні дані');
plot2d(x,y,-9);
endfunction

```

```

//function [f2]=Output_yr(x,y,yr,piyr_min,piyr_max)
function [f2, f3]=Output_yr(y_pezy_min, y_pezy_max, xi_z, x, y,
pezy_min, pezy_max, zxi, zx, zy)
f2=figure(); //Создание графического окна.
//Устанавливаем размеры окна.
set(f2, 'position', [100,100,700,300]);
//Устанавливаем заголовок окна.
set(f2, 'figure_name', 'Еліпс передбачення');

plot2d(zxi, pezy_max, 5);
plot2d(zx, zy, -9);
//plot2d(xi, yr, 1);
plot2d(zxi, pezy_min, 5);

f3=figure(); //Создание графического окна.
//Устанавливаем размеры окна.
set(f3, 'position', [150,150,700,300]);
//Устанавливаем заголовок окна.
set(f3, 'figure_name', 'Трансформований еліпс передбачення');

plot2d(xi_z, y_pezy_max, 5);
plot2d(x, y, -9);
//plot2d(xi, yr, 1);
plot2d(xi_z, y_pezy_min, 5);
endfunction

//Функция закрытия окна.
function _Close()
close(f);
close(f1);
close(f2);
close(f3);
endfunction

```