

РЕФЕРАТ

Неотъемлемой частью оценки успеваемости студентов является анализ результатов тестирования, однако существующие на данный момент системы тестирования учитывают лишь правильность ответов.

Цель данной работы — разработка математической модели психологических особенностей человека на основе результатов тестирования. Это поможет создать тестирующую систему, способную давать нужные советы людям, которые прошли тестирование.

Для достижения поставленной цели

- был использован метод главных компонент, чтобы извлечь наиболее важные данные из анализируемых выборок;
- был использован критерий Пирсона (χ^2) для проверки гипотез о распределении выборок.

ABSTRACT

Analysis of testing results is a valuable tool in assessing students with their academic standing, however current testing systems focus only in the accuracy of answers.

This study aims to design a mathematical model of individual psychological characteristics on the basis of test results. This will help to implement testing system which is able to give relevant advices to person who has passed the test.

To perform the study

- principal components analysis was used to get most significant data from analyzed samples;
- Pearson's chi-squared test (χ^2) was used to check hypothesis about samples' distribution.

РЕФЕРАТ

Аналіз результатів тестування є невід'ємною частиною перевірки рівня знань студентів, проте існуючі на даний момент системи тестування враховують лише вірність відповідей.

Метою даної роботи є побудова математичної моделі психологічних характеристик людини на основі результатів тестування. Це допоможе створити тестуючу систему, що може надавати доречні поради людям, що пройшли тестування.

Для досягнення мети

- було використано метод головних компонент, щоб отримати найбільш важливі дані з аналізованих вибірок;
- було використано критерій Пірсона (χ^2) для перевірки гіпотез щодо розподілу вибірок.

ЗМІСТ

Вступ	9
1 Теоретичні відомості	11
1.1 Математичне підґрунтя	11
1.1.1 Метод головних компонент	11
1.1.2 Побудова гістограми для критерію Пірсона	15
1.1.3 Критерій узгодженості Пірсона χ^2	16
1.2 Дані з психології	24
1.2.1 Типи вищої нервової діяльності	24
1.2.2 Теппінг-тест (Tapping rate)	25
Висновки до розділу 1.	27
2 Моделювання результатів виконання тестових завдань	29
2.1 Моделювання результатів виконання теппінг-тесту.	29
2.1.1 Процес Пуассона.	29
2.1.2 Квадратична апроксимація методом найменших квадратів.	31
2.1.3 Аналіз квадратичної апроксимації	34
2.2 Моделювання результатів виконання тестових завдань.	40
2.2.1 Інформаційний метод прогнозування часу розв'язання завдань	40
2.2.2 Моделювання тесту	43
Висновки до розділу 2.	46
3 Охорона праці	49
3.1 Технічні рішення щодо безпечного виконання роботи	49
3.2 Електробезпека	50
3.3 Пожежна безпека	51
3.4 Виробничий шум	52
3.5 Виробниче освітлення	52
3.6 Мікроклімат	53

3.7 Організація оптимального режиму праці та відпочинку	54
3.8 Склад повітря робочої зони	55
3.9 Виробничі випромінювання	55
Висновки до розділу 3.	56
Перелік посилань	57
Додаток А	59
Додаток Б	61
Додаток В	62
Додаток Г	64
Додаток Д	66
Додаток Е	69
Додаток Є	71
Додаток Ж.	73
Додаток З	75

ВСТУП

Актуальність роботи полягає в тому, що дистанційне навчання стає дедалі популярнішим, а існуючі на даний момент системи тестування недостатньо гнучкі: вони аналізують лише відповіді на запитання, відносячи їх до вірних або невірних, а на цій базі роблять кінцевий висновок щодо знань студента. З одного боку, це підвищує їх об'єктивність, з іншого, навпаки: всі студенти різні і більшості з них потрібна допомога хоча б у вигляді порад. У викладачів може не вистачати часу на докладне знайомство з кожним студентом, але якщо буде програмно-технічний комплекс, що підкаже викладачам, до яких студентів який підхід краще мати, це безумовно підвищить якість навчання. Стрімкий розвиток комп'ютерної техніки й інформаційних технологій надає можливість визначати ритм складання тесту, а також індивідуальні особливості людини. Дані психологічних досліджень допоможуть правильно трактувати отримані значення, а добре вивчені та перевірені часом математичні методи надають великі можливості для систематизації та обробки результатів вимірювання.

Об'єкти дослідження: студенти, системи тестування.

Предмети дослідження: психологічні особливості студентів, моделі їх поведінки.

Задача: Побудувати математичну модель психологічних характеристик людини на основі результатів тестування.

Метою роботи є покращення якості навчання за допомогою порад студентам і викладачам практичних занять.

Завдання наступні:

- 1) Вивчити математичні методи та розділи психології, що дозволять розробити математичну модель психологічних характеристик людини, пояснити та обґрунтувати отримані результати
- 2) Розробити математичну модель людини, що складає тести

3) Розробити та проаналізувати модель складання тестів людьми різних типів

1 ТЕОРЕТИЧНІ ВІДОМОСТІ

1.1 Математичне підґрунтя

1.1.1 Метод головних компонент

Метод головних компонент (Principal component analysis) — метод, що дозволяє зменшити розмірність досліджуваної вибірки з мінімальними втратами інформації. [1]

Маємо m об'єктів, з яких треба зняти по n певних властивостей. На вході в нас є виборки \vec{X}_k , кожна з яких відповідає сукупності властивостей k -го об'єкту

$$\vec{X}_k = \begin{bmatrix} x_k^1 \\ x_k^2 \\ \vdots \\ x_k^n \end{bmatrix}, \quad k = \overline{1, m}.$$

Згрупуємо всі вимірювання в одну матрицю X

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_m^1 \\ x_1^2 & x_2^2 & \dots & x_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & \dots & x_m^n \end{bmatrix}.$$

Спочатку нам знадобиться знайти вибіркові середні значення для кожної

властивості

$$a_i = \frac{1}{m} \cdot \sum_{k=1}^m x_k^i, \quad i = \overline{1, n}.$$

Маємо вектор вибірових середніх значень

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}.$$

Центруємо отримані дані, що містяться в матриці X , віднявши від кожного стовбця вектор вибірових середніх \vec{a}

$$\tilde{X} = \begin{bmatrix} \tilde{x}_1^1 & \tilde{x}_2^1 & \dots & \tilde{x}_m^1 \\ \tilde{x}_1^2 & \tilde{x}_2^2 & \dots & \tilde{x}_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{x}_1^n & \tilde{x}_2^n & \dots & \tilde{x}_m^n \end{bmatrix} = \begin{bmatrix} x_1^1 - a_1 & x_2^1 - a_1 & \dots & x_m^1 - a_1 \\ x_1^2 - a_2 & x_2^2 - a_2 & \dots & x_m^2 - a_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^n - a_n & x_2^n - a_n & \dots & x_m^n - a_n \end{bmatrix}.$$

Обчислюємо вибірову коваріаційну матрицю властивостей. Вибіркову коваріацію i та j властивості рахуємо за формулою

$$\sigma_i^j = \frac{1}{m} \cdot \sum_{k=1}^m \tilde{x}_k^i \cdot \tilde{x}_k^j = \frac{1}{m} \cdot \sum_{k=1}^m \left[(x_k^i - a_i) \cdot (x_k^j - a_j) \right], \quad i, j = \overline{1, n}.$$

Маємо вибірку коваріаційну матрицю

$$K = \begin{bmatrix} \sigma_1^1 & \sigma_2^1 & \dots & \sigma_n^1 \\ \sigma_1^2 & \sigma_2^2 & \dots & \sigma_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1^n & \sigma_2^n & \dots & \sigma_n^n \end{bmatrix}.$$

Щоб отримувати лише потрібну інформацію, ми хочемо знайти таке ортогональне лінійне перетворення L вхідної матриці \tilde{X} , щоб отримати матрицю $Y = L \cdot \tilde{X}$, яка має діагональну вибірку коваріаційну матрицю K' з незростаючими зверху вниз значеннями. Діагональна вибірка коваріаційна матриця гарантує той факт, що отримані значення Y будуть некорельованими. Рангування значень діагональних елементів матриці K' за величиною дасть більш наочне уявлення про будову досліджуваних об'єктів, адже діагональні елементи — вибірккові дисперсії. Чим більше дисперсія, тим більше відповідна властивість змінюється від об'єкту до об'єкту, і тим більше корисної інформації вона нам надає.

Вибіркова коваріаційна матриця K' для $Y = L \cdot \tilde{X}$ має вигляд

$$K' = L \cdot K \cdot L^* = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}.$$

З лінійної алгебри відомо, що матриця L складається з координат власних ве-

кторів матриці K , а елементи λ_k — її власні числа, які існують і є невід’ємними через невід’ємну означеність матриці K . Вважаємо, що числа $\lambda_1, \dots, \lambda_n$ впорядковані від більшого до меншого для зручності подальших дій. Позначимо власний вектор матриці K , що відповідає власному числу λ_k , як \vec{l}_k . Тоді

$$\vec{l}_k = [l_k^1, l_k^2, \dots, l_k^n], \quad k = \overline{1, n}.$$

Матриця L має вигляд

$$L = \begin{bmatrix} l_1^1 & l_1^2 & \dots & l_1^n \\ l_2^1 & l_2^2 & \dots & l_2^n \\ \vdots & \vdots & \ddots & \vdots \\ l_n^1 & l_n^2 & \dots & l_n^n \end{bmatrix}.$$

Треба зменшити розмірність простору досліджуваних параметрів системи з n до $p < n$, але при цьому втратити якомога менше відомостей про досліджувані об’єкти. Введемо міру інформації, що залишається при зменшенні кількості компонент, що розглядаються

$$I = \frac{\lambda_1 + \dots + \lambda_p}{\lambda_1 + \dots + \lambda_n}.$$

Будемо вважати, що діємо продуктивно, тому починаємо обирати з перших компонент, адже саме вони є найбільш інформативними. Також бачимо, що інформативність змінюється в межах від 0 (нічого не дізнаємось) до 1 (зберегли усю інформацію).

Надалі буде розглядатися матриця головних компонент Y

$$Y = \begin{bmatrix} y_1^1 & y_2^1 & \dots & y_m^1 \\ y_1^2 & y_2^2 & \dots & y_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ y_1^p & y_2^p & \dots & y_m^p \end{bmatrix}.$$

1.1.2 Побудова гістограми для критерію Пірсона

Для подальшого аналізу потрібно здобути щільність розподілу головних компонент. Оскільки маємо справу з вибіркою і вибічковими характеристиками, потрібно побудувати гістограму, адже це і є вибіркова характеристика, що відповідає щільності. [2]

Побудуємо j -й стовбець гістограми для виборки з k -ї строки матриці Y

$$h_j^k = \frac{1}{m} \cdot \sum_{i=1}^m \mathbb{1}(y_i^k \in I_j^k), \quad j = \overline{1, N}, \quad k = \overline{1, p},$$

де I^k — набір напівінтервалів, що розбиває відрізок $\left[\min_{i=\overline{1, m}} y_i^k; \max_{i=\overline{1, m}} y_i^k \right]$ на N рівних частин. Для вибору N можна скористатися досить відомою формулою Стюрджеса (Sturges' formula) [3]

$$N = \lfloor \log_2 m \rfloor + 1.$$

Маємо матрицю гістограм

$$H = \begin{bmatrix} h_1^1 & h_2^1 & \dots & h_N^1 \\ h_1^2 & h_2^2 & \dots & h_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ h_1^p & h_2^p & \dots & h_N^p \end{bmatrix}$$

і напівінтервалів, що відповідають кожному стовбчику кожної гістограми

$$I = \begin{bmatrix} I_1^1 & I_2^1 & \dots & I_N^1 \\ I_1^2 & I_2^2 & \dots & I_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ I_1^p & I_2^p & \dots & I_N^p \end{bmatrix}.$$

1.1.3 Критерій узгодженості Пірсона χ^2

Гістограма може використовуватися не тільки для графічної інтерпретації отриманих даних, але й для віднесення вибірки до якогось відомого розподілу. Відповідь на питання “Чи дійсно вибірка y_1^k, \dots, y_m^k має розподіл F^k ?” може надати критерій узгодженості Пірсона.

Розглянемо вектор

$$\eta^k = \left[\frac{\nu_1^k - m \cdot \rho_1^k}{\sqrt{m \cdot \rho_1^k}}, \dots, \frac{\nu_N^k - m \cdot \rho_N^k}{\sqrt{m \cdot \rho_N^k}} \right].$$

Знайдемо його характеристичну функцію

$$\varphi_{\eta^k}(\lambda) = M e^{i \cdot (\lambda, \eta^k)}, \quad \lambda \in \mathbb{R}^N.$$

Для зручності перепозначимо індикатор

$$\mathfrak{I}_{i,j}^k = \mathbb{1}\{y_i^k \in I_j^k\}.$$

Подивимось, чому дорівнює скалярний добуток в експоненті

$$\begin{aligned} (\lambda, \eta^k) &= \sum_{j=1}^N \lambda_j \cdot \frac{\nu_j^k - m \cdot \rho_j^k}{\sqrt{m \cdot \rho_j^k}} = \sum_{j=1}^N \frac{\lambda_j}{\sqrt{m \cdot \rho_j^k}} \cdot \sum_{i=1}^m (\mathfrak{I}_{i,j}^k - \rho_j^k) = \\ &= \sum_{j=1}^N \sum_{i=1}^m \frac{\lambda_j}{\sqrt{m \cdot \rho_j^k}} \cdot (\mathfrak{I}_{i,j}^k - \rho_j^k) = \sum_{i=1}^m \sum_{j=1}^N \lambda_j \cdot \frac{\mathfrak{I}_{i,j}^k - \rho_j^k}{\sqrt{m \cdot \rho_j^k}}. \end{aligned}$$

Бачимо суму m незалежних однаково розподілених випадкових величин. Введемо позначення

$$\mathfrak{I}_j^k = \mathbb{1}\{y_1^k \in I_j^k\},$$

а також позначимо новий випадковий вектор

$$\zeta^k = \left[\frac{\mathfrak{I}_1^k - \rho_1^k}{\sqrt{m \cdot \rho_1^k}}, \dots, \frac{\mathfrak{I}_N^k - \rho_N^k}{\sqrt{m \cdot \rho_N^k}} \right].$$

Скалярний добуток прийняв вигляд

$$(\lambda, \eta^k) = \sum_{i=1}^m \sum_{j=1}^N \lambda_j \cdot \zeta_j^k = \sum_{i=1}^m (\lambda, \zeta^k) = m \cdot (\lambda, \zeta^k).$$

За рахунок незалежності випадкових величин ζ_j^k маємо

$$\varphi_{\eta^k}(\lambda) = M e^{i \cdot (\lambda, \eta^k)} = M e^{m \cdot i \cdot (\lambda, \zeta^k)} = \left(M e^{i \cdot (\lambda, \zeta^k)} \right)^m. \quad (1.1)$$

Розглянемо характеристичну функцію випадкового вектора ζ^k

$$\varphi_{\zeta^k}(\lambda) = M \left[\exp \left\{ i \cdot \sum_{j=1}^N \lambda_j \cdot \zeta_j^k \right\} \right]. \quad (1.2)$$

Легко побачити, що

$$\begin{aligned} (\lambda, \zeta^k) &= \sum_{j=1}^N \lambda_j \cdot \zeta_j^k = \sum_{j=1}^N \lambda_j \cdot \frac{\mathfrak{I}_j^k - \rho_j^k}{\sqrt{m \cdot \rho_j^k}} = \sum_{j=1}^N \left(\frac{\lambda_j}{\sqrt{m \cdot \rho_j^k}} \cdot \mathfrak{I}_j^k - \frac{\sqrt{\rho_j^k} \cdot \lambda_j}{\sqrt{m}} \right) = \\ &= \sum_{j=1}^N \mathfrak{I}_j^k \cdot \left(\frac{\lambda_j}{\sqrt{m \cdot \rho_j^k}} - \sum_{l=1}^N \frac{\sqrt{\rho_l^k} \cdot \lambda_l}{\sqrt{m}} \right). \end{aligned}$$

Тобто характеристична функція (1.2) приймає вигляд

$$\varphi_{\zeta^k}(\lambda) = M \left[\sum_{j=1}^N \mathfrak{I}_j^k \cdot \exp \left\{ \frac{i}{\sqrt{m}} \left(\frac{\lambda_j}{\sqrt{\rho_j^k}} - \sum_{l=1}^N \sqrt{\rho_l^k} \cdot \lambda_l \right) \right\} \right].$$

Перепозначимо вираз в круглих дужках

$$\mathfrak{z}^k = \frac{\lambda_j}{\sqrt{\rho_j^k}} - \sum_{l=1}^N \sqrt{\rho_l^k} \cdot \lambda_l.$$

Математичне очікування індикатора — ймовірність події, яку він перевіряє. Отже

$$\varphi_{\zeta^k}(\lambda) = \sum_{j=1}^N \rho_j^k \cdot \exp \left\{ \frac{i \cdot \mathfrak{z}^k}{\sqrt{m}} \right\}.$$

Якщо розмір вибірки m буде зростати, то характеристична функція η^k (1.1) буде поводитись наступним чином

$$\begin{aligned} \lim_{m \rightarrow \infty} \varphi_{\eta^k}(\lambda) &= \lim_{m \rightarrow \infty} \left(1 + \sum_{k=1}^N p_k \cdot \left[\exp \left\{ \frac{i \cdot \mathfrak{z}^k}{\sqrt{m}} \right\} - 1 \right] \cdot \frac{m}{m} \right)^m = \\ &= \lim_{m \rightarrow \infty} \exp \left\{ m \cdot \sum_{k=1}^N p_k \cdot \left[\exp \left\{ \frac{i \cdot \mathfrak{z}^k}{\sqrt{m}} \right\} - 1 \right] \right\}. \end{aligned}$$

Для $\exp \left\{ \frac{i \cdot \mathfrak{z}^k}{\sqrt{m}} \right\}$ використаємо співвідношення

$$e^\alpha - 1 \approx \alpha + \frac{\alpha^2}{2}, \quad \alpha \ll 1.$$

Маємо

$$\begin{aligned} \sum_{j=1}^N \rho_j^k \cdot \frac{i \cdot \mathfrak{z}^k}{\sqrt{m}} &= \sum_{j=1}^N \rho_j^k \cdot \frac{i}{\sqrt{m}} \cdot \left(\frac{\lambda_j}{\sqrt{\rho_j^k}} - \sum_{l=1}^N \sqrt{\rho_l^k} \cdot \lambda_l \right) = \\ &= \frac{i}{\sqrt{m}} \cdot \left(\sum_{j=1}^N \sqrt{\rho_j^k} \cdot \lambda_j - \sum_{l=1}^N \sqrt{\rho_l^k} \cdot \lambda_l \right) = 0, \end{aligned}$$

$$\begin{aligned} \sum_{j=1}^N \rho_j^k \cdot \left(\frac{i \cdot \mathfrak{z}^k}{\sqrt{m}} \right)^2 &= - \sum_{j=1}^N \frac{\rho_j^k}{m} \cdot \left(\frac{\lambda_j}{\sqrt{\rho_j^k}} - \sum_{l=1}^N \sqrt{\rho_l^k} \cdot \lambda_l \right)^2 = \\ &= - \frac{1}{m} \cdot \left[\sum_{j=1}^N \lambda_j^2 - \left(\sum_{l=1}^N \sqrt{\rho_l^k} \cdot \lambda_l \right)^2 \right], \end{aligned}$$

тому

$$\begin{aligned} \lim_{m \rightarrow \infty} \varphi_{\eta^k}(\lambda) &= \lim_{m \rightarrow \infty} \exp \left\{ - \frac{m}{m \cdot 2} \cdot \left[\sum_{j=1}^N \lambda_j^2 - \left(\sum_{l=1}^N \sqrt{\rho_l^k} \cdot \lambda_l \right)^2 \right] \right\} = \\ &= \exp \left\{ - \frac{1}{2} \cdot \left[\sum_{j=1}^N \lambda_j^2 - \left(\sum_{l=1}^N \sqrt{\rho_l^k} \cdot \lambda_l \right)^2 \right] \right\} = e^{-\frac{1}{2} \cdot (A^k \lambda, \lambda)}. \end{aligned}$$

Матриця A^k побудована наступним чином

$$A^k = \left\| \delta_{ij} - \sqrt{\rho_i^k} \cdot \sqrt{\rho_j^k} \right\|_{i=1}^n.$$

Симетричність матриці очевидна, тому треба довести її невід'ємну визначеність, щоб стверджувати, що вона є коваріаційною. Для цього візьмемо вектор

$$e^k = \left[\sqrt{\rho_1^k}, \dots, \sqrt{\rho_N^k} \right], \quad \|e^k\| = 1.$$

Тоді бачимо, що

$$(A^k \lambda, \lambda) = \|\lambda\|^2 - (\lambda, e^k)^2. \quad (1.3)$$

З нерівності Коші маємо

$$\|(\lambda, e^k)\| \leq \|\lambda\| \cdot \|e^k\| = \|\lambda\|.$$

Тобто матриця є дійсно невід’ємно визначеною і вектор η^k розподілений за нормальним законом з нульовим середнім і коваріаційною матрицею A^k .

Для подальших розрахунків розглянемо стандартний гаусівський вектор як суму випадкових нормально розподілених випадкових величин в стандартному базисі \mathbb{R}^N , який позначимо $[e_1, \dots, e_N]$

$$\xi = \sum_{j=1}^N \xi_j \cdot e_j \sim N(\vec{0}, I).$$

Згадуємо, що ортогональні перетворення U зберігають відстані, а також справедливо наступне

$$U\xi \sim N(0, UIU^{-1}) \sim N(\vec{0}, I).$$

Також ортонормований базис залишається ортонормованим базисом після ортогонального перетворення U . Оберемо такий оператор U , щоб набір $[e_1, \dots, e_N]$ під його дією перетворився на $[f_1, \dots, f_N]$, де

$$f_1 = e^k = \left[\sqrt{\rho_1^k}, \dots, \sqrt{\rho_N^k} \right].$$

Тоді маємо вектор

$$U\xi = \hat{\xi} = \sum_{j=1}^N \hat{\xi}_j \cdot f_j \sim N(\vec{0}, I).$$

Подивимось, який розподіл має наступний вектор

$$\Upsilon = \sum_{j=2}^N \hat{\xi}_j \cdot f_j = \hat{\xi} - \hat{\xi}_1 \cdot e^k.$$

Для цього розглянемо квадратичну форму

$$M(\Upsilon, \lambda)^2 = \sum_{j=2}^N (\lambda, f_j)^2 = \sum_{j=1}^N (\lambda, f_j)^2 - (\lambda, f_1)^2 = \|\lambda\|^2 - (\lambda, e^k)^2 = (A^k \lambda, \lambda).$$

З рівності (1.3) бачимо, що випадкові вектори η^k та Υ мають однаковий розподіл. Отже, розподіли їх норм теж співпадають. Оскільки сума $N - 1$ квадратів незалежних стандартних гаусових випадкових величин має розподіл Пірсона з $N - 1$ ступенями вільності

$$\|\Upsilon\|^2 = \sum_{j=2}^N \xi_j^2 \sim \chi_{N-1}^2.$$

Маємо

$$\|\eta^k\|^2 = \sum_{j=1}^N \frac{(\nu_j^k - m \cdot \rho_j^k)^2}{m \cdot \rho_j^k} = m \cdot \sum_{j=1}^N \frac{(h_j^k - \rho_j^k)^2}{\rho_j^k} \sim \chi_{N-1}^2.$$

Останнє співвідношення дає змогу перевіряти належність виборки y_1^k, \dots, y_m^k до розподілу F^k . Перевірка виглядає наступним чином.

Розглянемо випадкову величину

$$R^k = m \cdot \sum_{j=1}^N \frac{(h_j^k - \rho_j^k)^2}{\rho_j^k}. \quad (1.4)$$

Обираємо рівень значущості α для функції розподілу χ_{N-1}^2 і шукаємо відповідне до кількості ступенів вільності r_α . Рівень значущості — ймовірність помилки першого роду, тобто ймовірність того, що буде відкинута вірна гіпотезу

$$\mathbb{P}(\chi_{N-1}^2 \geq r_\alpha) = \alpha.$$

Якщо $R^k \leq r_\alpha$, то гіпотеза про те, що вибірка Y^k дійсно має розподіл F^k , приймається.

Розглянемо той випадок, коли ймовірність ρ_i^k відгадана невірно. Повернемося до формули (1.4)

$$R^k = \sum_{j=1}^N \frac{(\nu_j^k - m \cdot \rho_j^k)^2}{m \cdot \rho_j^k}.$$

Всі члени суми є невід'ємними. Якщо хоча б один елемент буде завеликим, то великою буде вся сума. Маємо випадкову величину η

$$\eta = \nu_i^k - m \cdot \rho_i^k = \sum_{j=1}^m (\xi_j - \rho_i^k), \quad \mathbb{1}(y_j^k \in I_i^k) = \xi_j.$$

Якщо ρ_i^k вгадано невірно, то воно не дорівнює математичному очікуванню індикатора. Додамо та віднімемо справжнє математичне очікування

$$\eta = \sum_{j=1}^m (\xi_j - M \xi_1 + M \xi_1 - \rho_i^k) = \sum_{j=1}^m (\xi_j - M \xi_1) + \sum_{j=1}^m (M \xi_1 - \rho_i^k).$$

Останній доданок є просто різницею, яку помножено на m

$$\eta = \sum_{j=1}^m (\xi_j - M \xi_1) + m \cdot (M \xi_1 - \rho_i^k).$$

Поділимо на \sqrt{m} , щоб скористатися центральною граничною теоремою

$$\frac{\eta}{\sqrt{m}} = \frac{1}{\sqrt{m}} \cdot \sum_{j=1}^m (\xi_j - M \xi_1) + \frac{1}{\sqrt{m}} \cdot m \cdot (M \xi_1 - \rho_i^k).$$

Перший доданок асимптотично має розподіл $N(0, \sigma^2)$, де σ^2 — дисперсія випад-

кової величини ξ_1 для достатньо великих m . Отже вся сума зростає пропорційно до \sqrt{m}

$$\frac{\eta}{\sqrt{m}} = \frac{1}{\sqrt{m}} \cdot \sum_{j=1}^m (\xi_j - M \xi_1) + \sqrt{m} \cdot (M \xi_1 - \rho_i^k) \sim \sqrt{m} \cdot (M \xi_1 - \rho_i^k).$$

Тобто зараз R^k буде зростати пропорційно до величини m , і буде великим у порівнянні з r_α , що призведе до відхилення невірної гіпотези.

1.2 Дані з психології

1.2.1 Типи вищої нервової діяльності

Для визначення того, які показники вимірювати і яким чином, скористуємось відомою класифікацією типів вищої нервової діяльності.

Згідно з Павловим[4] типи вищої нервової діяльності характеризуються трьома показниками: сила нервової системи (сильна або слабка), врівноваженість (врівноважена або неврівноважена) та рухливість (рухлива або інертна). Павлов розглядає 4 комбінації цих показників з 8 можливих:

- 1) Слабка
- 2) Сильна та неврівноважена
- 3) Сильна, врівноважена та інертна
- 4) Сильна, врівноважена та рухлива

Далі ці класи (комбінації) будуть називатися відповідно слабкий, неврівноважений, інертний та рухливий.

1.2.2 Теппінг-тест (Tapping rate)

Існують відомі залежності між типом вищої нервової діяльності та зміною максимального темпу рухів кистю руки з часом. Протягом 30 секунд людина намагається притримуватися максимально можливого для себе темпу. Показники темпу фіксуються через кожні 5 секунд, а далі по 6 отриманим точкам будується крива темпа руху. [5]

Для тесту можна використовувати ручку (олівець) і папір, або телеграф. Сучасні технології дозволяють проводити тест за допомогою клавіатури комп'ютера або екрану планшета.

З олівцем і папіром тест поводиться наступним чином:

- 1) На папері креслиться 6 квадратів
- 2) Людина починає ставити якомога більше точок в першому квадраті впродовж перших 5 секунд
- 3) Коли проходить 5 секунд, потрібно перейти до наступного квадрату і ставити точки там
- 4) Процедура повторюється до тих пір, доки не пройде 30 секунд — в кінці буде заповнено всі 6 квадратів

Далі підраховується кількість точок в кожному квадраті та малюється ламана, де горизонтальна вісь відповідає номеру часового проміжку (номеру квадрата), а вертикальна відповідає кількості точок в квадраті.

Трактуються отримані дані наступним чином:

- 1) Спадна ламана відповідає слабкому типу (рис. 1.1д). Вона спадає після перших 5 секунд тесту і не повертається до початкового рівня
- 2) Ламана, що спочатку зростає, а після 10-15 секунд спадає нижче початкового рівня (проміжна між рівною та опуклою) відповідає неврівноваженому типу (рис. 1.1б).

- 3) Опукла вниз ламана відповідає інертному типу (рис. 1.1г). Вона спочатку спадає, а на 25-30 секундах може зрости до початкового темпу
- 4) Опукла вгору ламана відповідає рухливому типу (рис. 1.1а). Це така лама-на, що зростає в перші 10-15 секунд тесту, а після 25-30 секунд повертає-ться або падає нижче початкового рівня
- 5) Також темп може залишатися приблизно на одному рівні протягом усього тесту, що є оптимальним для складання іспитів (рис. 1.1в).

Достовірним відхиленням достатньо вважати різницю в два і більше натиснення між двома сусідніми п'ятисекундними проміжками. [5]

Оскільки цей тест заснований на вимірюванні витривалості нервової систе-ми людини за умови максимального навантаження та перевіряє темп реагування (натиснення) на подразнювачі (внутрішній подразнювач — команда собі “треба тиснути”), було вирішено використовувати відомі вигляди кривих (рис. 1.1) при моделюванні результатів виконання завдань однакової складності.

Потрібно зауважити, що швидкість розв'язування задач може змінюватися з досвідом. Тобто, якщо студент зі слабкою нервовою системою буде тренувати-ся виконувати завдання, то його показники з часом перейдуть на якісно новий рівень. Щодо студентів з сильною нервовою системою: швидкість не завжди означає якість виконання завдань.

Завдання системи — класифікувати студентів в автоматичному режимі, з метою подальшого надання порад викладачам практичних занять щодо підвище-ння продуктивності роботи кожного студента. Це полегшує знаходження інди-відуального підходу. Наприклад, тим, хто надто швидко втомлюється, потрібно розв'язувати якомога більше базових завдань, що не є складними, але розв'язок яких повинен бути на рівні рефлексів. Студентам, які поспішають, буде корисно ретельно коментувати у письмовій формі хід своїх думок, щоб вгамуватися та підвищити свою уважність.

Для тесту було обрано саме 30-секундний проміжок часу, адже спочатку

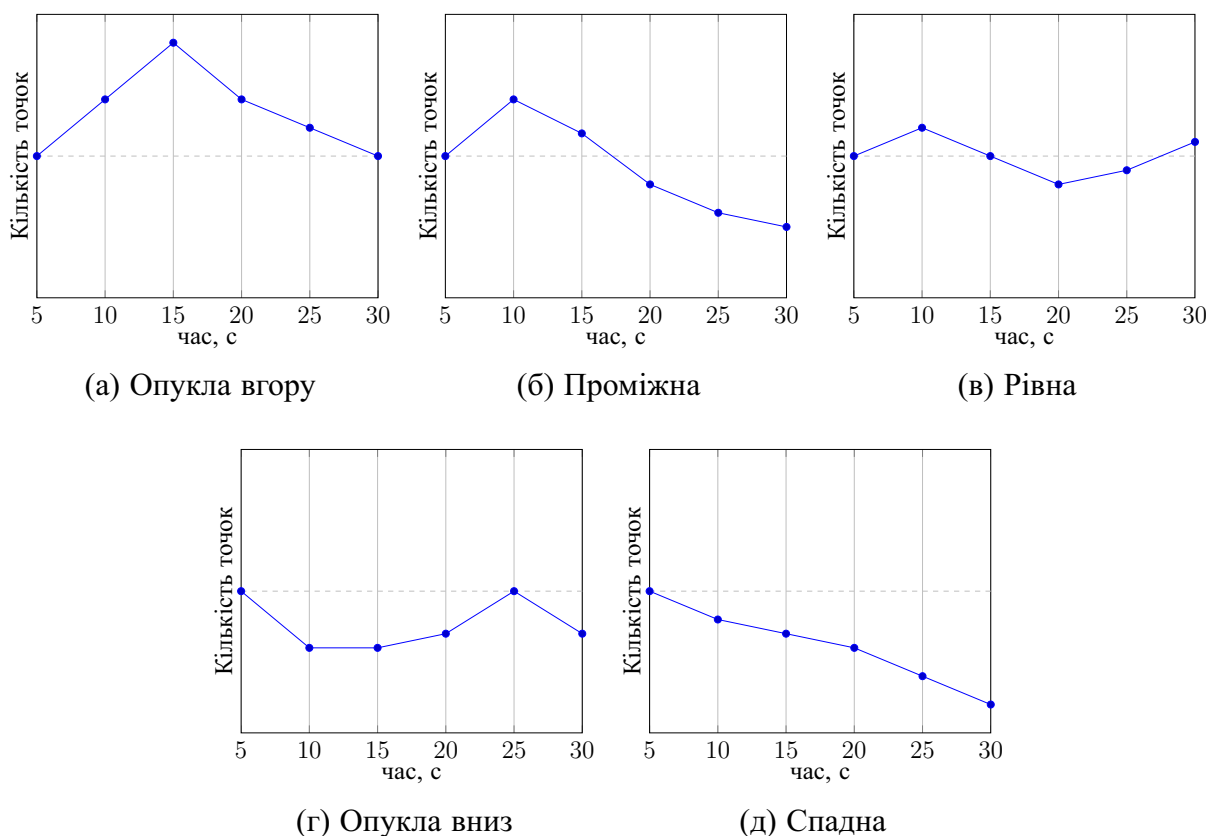


Рисунок 1.1 — Загальний вигляд залежностей кількості поставлених точок від часу. Пунктирна лінія — кількість точок в перші 5 секунд

виміри виконувалися протягом однієї хвилини і було виявлено, що найважливіша інформація отримується протягом перших 20-25 секунд, а далі лише марно втрачається час та сили тестованого. [5]

Висновки до розділу 1

В даному розділі було досліджено відомі статистичні методи обробки даних та психологічні особливості студентів, що допоможуть створити систему тестування, яка буде більш об'єктивною, ніж класичні аналоги. Також було розпочато моделювання, за допомогою якого будуть виконані перші кроки в створенні та тестуванні системи.

Метод головних компонент буде використано для виявлення найвагоміших

рис (головних компонент) поведінки студентів під час виконання тестів. Для його застосування було вирішено розбити час розв'язання тестових завдань на скінченну кількість відрізків — дискретизувати час.

За допомогою критерію узгодженості Пірсона буде перевірено гіпотези щодо розподілів головних компонент, які були знайдені на попередньому кроці. Оскільки вони будуть лінійними комбінаціями інших елементів виборки, можна припустити, що розподіл буде нормальним — залишиться визначити середнє та дисперсію кожної головної компоненти.

Дані з психології надали матеріал, на основі якого можна розділити людей на кілька категорій, кожна з яких має свої особливості. На основі цього було вирішено, які поради система буде надавати викладачеві практичних занять щодо роботи з різними студентами.

2 МОДЕЛЮВАННЯ РЕЗУЛЬТАТІВ ВИКОНАННЯ ТЕСТОВИХ ЗАВДАНЬ

2.1 Моделювання результатів виконання теппінг-тесту

2.1.1 Процес Пуассона

Щоб моделювати темп виконання студентами контрольних задач, використаємо відомі дані з психології щодо ритму виконання теппінг-тесту для людей з різними типами вищої нервової діяльності. Оскільки постукування є потоком однорідних випадкових подій, змоделюємо їх як реалізації процесу Пуассона.

Нехай $\{N(t) | t \geq 0\}$ — процес Пуассона з інтенсивністю $\lambda(t)$. Маємо 7 контрольних точок на часовій вісі $t_0 = 0, t_1 = 5, \dots, t_6 = 30$. Введемо відрізок часу T , за який проводиться вимірювання, та проміжки T_i , на які його розбито в експерименті

$$T = (0, 30], \quad T_i = (t_{i-1}, t_i], \quad i = \overline{1, 6}.$$

Нехай ν_i визначається наступним чином

$$\nu_i = N(t_i) - N(t_{i-1}), \quad i = \overline{1, 6},$$

тоді ν_i — випадкова величина, що має розподіл Пуассона з параметром λ_i , де [6]

$$\lambda_i = m(T_i) = \int_{t_{i-1}}^{t_i} \lambda(\tau) d\tau, \quad i = \overline{1, 6}.$$

Функція m — міра інтенсивності. [7]

Проміжки T_i утворюють розбиття множини T , а отже не перетинаються. Це означає, що випадкові величини ν_i незалежні між собою, і є можливість згенерувати процес натискання за допомогою 6 незалежних випадкових величин, що мають розподіл Пуассона.

Інтенсивність (параметр λ) є математичним очікуванням випадкової величини, що має розподіл Пуассона. Маючи результати реальних експериментів, можна обчислити середні значення кількості постукувань для кожного часового проміжку T_i , а отримані результати використовувати в якості параметрів λ_i .

Є наступні відомості щодо розподілу типів:

- Слабкий — 29%,
- Неврівноважений — 38%,
- Рухливий — 23%,
- Інертний — 10%.

Спростимо ці результати та додамо змішаний тип

- Змішаний — 3%,
- Слабкий — 25%,
- Неврівноважений — 35%,
- Рухливий — 25%,
- Інертний — 12%.

На основі цих спостережень за допомогою критерію Пірсона було підібрано необхідну інтенсивність процесу Пуассона. На рис. 2.2а зображено різні реалізації послідовності випадкових величин, що розподілені за законом Пуассона з параметрами

$$[\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6] = [40, 41, 46, 38, 35, 34]. \quad (2.1)$$

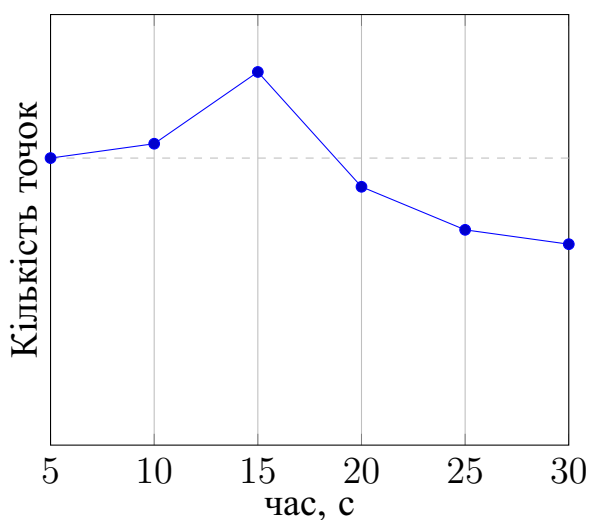


Рисунок 2.1 — “Опукла” ламана

2.1.2 Квадратична апроксимація методом найменших квадратів

Види результуючих ламаних (опукла вгору, опукла вниз, рівна, спадна) можна представити квадратичними функціями виду

$$y(t) = a \cdot t^2 + b \cdot t + c.$$

На рис. 2.2б зображено реалізації послідовності випадкових величин, що розподілені за законом Пуассона, та криві, що є результатом апроксимації цих реалізацій.

Щоб представити результати у вигляді квадратичних функцій, використаємо метод найменших квадратів. Нехай $\{y_i \mid i = \overline{1, 6}\}$ — результати поточного теплінг-тесту. Суть методу полягає в знаходженні таких коефіцієнтів a , b і c , з якими наступна величина приймає мінімальне значення

$$R^2(a, b, c) = \sum_{i=1}^6 (a \cdot t_i^2 + b \cdot t_i + c - y_i)^2.$$

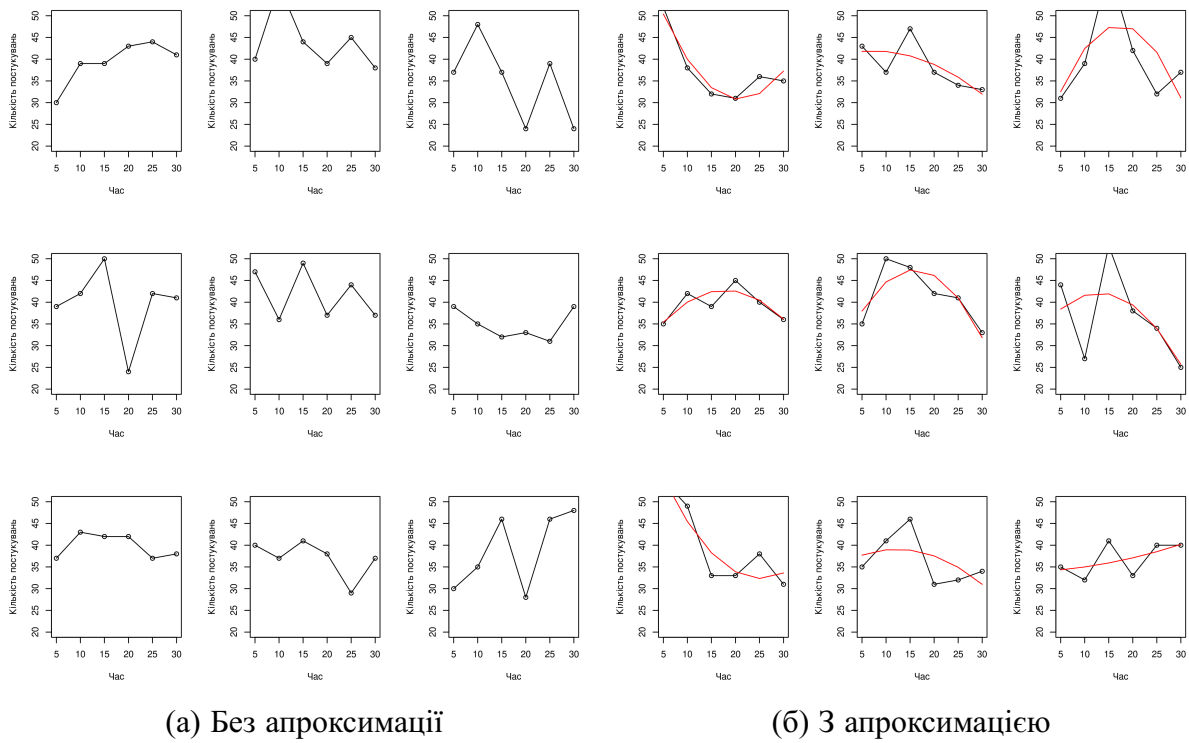


Рисунок 2.2 — Приклади результатів моделювання

Для цього треба взяти часткові похідні по кожному коефіцієнту, прирівняти ці похідні до нуля та розв'язати отриману систему

$$\begin{cases} \sum_{i=1}^6 2 \cdot (a \cdot t_i^2 + b \cdot t_i + c - y_i) = 0 \\ \sum_{i=1}^6 2 \cdot t_i \cdot (a \cdot t_i^2 + b \cdot t_i + c - y_i) = 0 \\ \sum_{i=1}^6 2 \cdot t_i^2 \cdot (a \cdot t_i^2 + b \cdot t_i + c - y_i) = 0. \end{cases}$$

Цю систему можна переписати в матричному вигляді наступним чином

$$\begin{bmatrix} \sum_{i=1}^6 t_i^2 & \sum_{i=1}^6 t_i & \sum_{i=1}^6 1 \\ \sum_{i=1}^6 t_i^3 & \sum_{i=1}^6 t_i^2 & \sum_{i=1}^6 t_i \\ \sum_{i=1}^6 t_i^4 & \sum_{i=1}^6 t_i^3 & \sum_{i=1}^6 t_i^2 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^6 y_i \\ \sum_{i=1}^6 y_i \cdot t_i \\ \sum_{i=1}^6 y_i \cdot t_i^2 \end{bmatrix}.$$

Введемо позначення:

$$\begin{aligned}\sum_{i=1}^6 1 &= N, \\ \sum_{i=1}^6 t_i^k &= X^k, \quad k = \overline{1, 4}, \\ \sum_{i=1}^6 y_i \cdot t_i^k &= YX^k, \quad k = \overline{0, 2}.\end{aligned}$$

Рівняння прийняло наступний вигляд

$$\begin{bmatrix} X^2 & X & N \\ X^3 & X^2 & X \\ X^4 & X^3 & X^2 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} Y \\ YX \\ YX^2 \end{bmatrix}.$$

Використаємо метод Крамера для розв'язання системи лінійних рівнянь. Визначник Δ

$$\begin{aligned}\Delta &= X^2 \cdot (X^2 \cdot X^2 - X \cdot X^3) - X \cdot (X^3 \cdot X^2 - X \cdot X^4) \\ &\quad + N \cdot (X^3 \cdot X^3 - X^2 \cdot X^4).\end{aligned}$$

Визначник Δ_a

$$\begin{aligned}\Delta_a &= Y \cdot (X^2 \cdot X^2 - X \cdot X^3) - YX \cdot (X \cdot X^2 - N \cdot X^3) \\ &\quad + YX^2 \cdot (X \cdot X - N \cdot X^2).\end{aligned}$$

Визначник Δ_b

$$\Delta_b = -Y \cdot (X^3 \cdot X^2 - X \cdot X^4) + YX \cdot (X^2 \cdot X^2 - N \cdot X^4) \\ - YX^2 \cdot (X^2 \cdot X - N \cdot X^3).$$

Визначник Δ_c

$$\Delta_c = Y \cdot (X^3 \cdot X^3 - X^2 \cdot X^4) - YX \cdot (X^2 \cdot X^3 - X \cdot X^4) \\ + YX^2 \cdot (X^2 \cdot X^2 - X \cdot X^3).$$

Відомо, що розв'язками є наступні вирази

$$a = \frac{\Delta_a}{\Delta}, \quad b = \frac{\Delta_b}{\Delta}, \quad c = \frac{\Delta_c}{\Delta}.$$

Бачимо, що параметри — лінійні комбінації y_i , а коефіцієнти при y_i — раціональні дробу. Тобто можна записати розв'язки наступним чином

$$a = \sum_{i=1}^6 y_i \cdot \frac{a_i}{\Delta}, \quad b = \sum_{i=1}^6 y_i \cdot \frac{b_i}{\Delta}, \quad c = \sum_{i=1}^6 y_i \cdot \frac{c_i}{\Delta}.$$

2.1.3 Аналіз квадратичної апроксимації

2.1.3.1 Задача

Запишемо умови, що накладаються на a , b і c , за яких парабола

$$y(t) = a \cdot t^2 + b \cdot t + c$$

відповідає раніше визначеним типам вищої нервової діяльності. Введемо такі позначення: t_{max} і t_{min} — точки, в яких парабола досягає локального максимуму та мінімуму на відрізку $[t_1, t_6]$ відповідно (t_1 і t_6 є точками першого та останнього спостереження).

Позначимо характеристики параболи, на основі яких буде відбуватися класифікація:

- 1) сумарне відхилення кривої на проміжку $\tau = [t^1; t^2]$ від положення $y(t^1)$ рахуємо за формулою

$$\sigma(\tau) = \int_{t^1}^{t^2} |\dot{y}(t)| dt;$$

- 2) від знаку a залежить те, чи t_g максимум або мінімум

$$M = \text{sign}(a);$$

- 3) якщо знаки похідних на кінцях відрізка $[t_1; t_6]$ різні, то функція досягає екстремуму саме на цьому відрізку

$$V = \text{sign}(\dot{y}(t_1) \cdot \dot{y}(t_6));$$

- 4) глобального екстремуму функція y досягає в точці

$$t_g = -\frac{b}{2 \cdot a};$$

- 5) позначимо площу трикутника, що обмежується прямими $t = t_1$, $t = t_g$, $y = \dot{y}(t)$ та $y = 0$

$$\sigma_S = \sigma([t_1; t_g]) = \frac{|(t_g - t_1) \cdot \dot{y}(t_1)|}{2},$$

а також площу трикутника, що обмежується прямими $t = t_6$, $t = t_g$, $y = \dot{y}(t)$ та $y = 0$

$$\sigma_F = \sigma([t_6; t_g]) = \frac{|(t_g - t_6) \cdot \dot{y}(t_6)|}{2};$$

- 6) найбільше відхилення від початкового положення за тієї умови, що глобальний екстремум знаходиться на відрізку $[t_1; t_6]$ ($V = -1$), рахується за формулою

$$\sigma_M = \max\{\sigma_S, \sigma_F\};$$

- 7) якщо глобальний екстремум досягається за межами відрізка $[t_1; t_6]$ ($V = 1$), відхилення буде наступним

$$\sigma_P = \max\{\sigma_S, \sigma_F\} - \min\{\sigma_S, \sigma_F\},$$

адже геометрично це буде різниця площ більшого і меншого трикутників, що були утворені прямими $t = t_1$, $t = t_6$, $t = t_g$, $y = \dot{y}(t)$ та $y = 0$;

2.1.3.2 Змішаний тип

Крива, що відповідає змішаному типу, весь час тримається приблизно на початковому рівні. Тобто, для певного ε , яке зазвичай беруть рівним 2, маємо такі випадки:

- 1) Функція y не монотонна на відрізку T

$$\begin{cases} V = -1, \\ \sigma_M < \varepsilon; \end{cases}$$

2) Функція y монотонна на відрізку T

$$\begin{cases} V = 1, \\ \sigma_P < \varepsilon. \end{cases}$$

2.1.3.3 Слабкий тип

Крива, що відповідає слабкому типу, спадає після моменту t_1 і не повертається до початкового положення.

Можливі наступні випадки:

1) t_g — точка глобального мінімуму

$$\begin{cases} V = 1, \\ t_g > t_6, \\ M = 1, \\ \sigma_P > \varepsilon; \end{cases}$$

2) t_g — точка глобального максимуму

$$\begin{cases} V = 1, \\ t_g < t_6, \\ M = -1, \\ \sigma_P > \varepsilon. \end{cases}$$

2.1.3.4 Неврівноважений тип

Крива, що відповідає неуврівноваженому типу, спочатку зростає, потім не пізніше моменту часу t_3 починає спадати нижче початкового рівня

$$\left\{ \begin{array}{l} M = -1, \\ V = -1, \\ t_g < t_3, \\ \sigma_F > \varepsilon. \end{array} \right.$$

2.1.3.5 Рухливий тип

Крива, що відповідає рухливому типу, спочатку зростає, не пізніше моменту t_3 починає спадати, а до моменту t_6 повертається до початкового рівня або спадає нижче нього

$$\left\{ \begin{array}{l} M = -1, \\ V = -1, \\ t_g \geq t_3, \\ \sigma_F - \sigma_S > 0. \end{array} \right.$$

2.1.3.6 Інертний тип

Крива, що відповідає інертному типу, спочатку спадає, а під кінець тесту зростає до початкового рівня

$$\begin{cases} M = 1, \\ V = -1, \\ 0 \leq \sigma_S - \sigma_F \leq -\varepsilon. \end{cases}$$

2.1.3.7 Результати класифікації

Було згенеровано 6 вибірок по 100 реалізацій процесу Пуассона з інтенсивністю, яку було вказано у формулі (2.1). Ці реалізації відповідають проходженню теплінг-тесту шістьма групами по сто студентів. Результати класифікації можна побачити на рис. 2.3.

Опис:

- 2.2% студентів мають змішаний тип вищої нервової діяльності;
- Студенти інертного типу зустрілися у 11.2% випадків;
- 35.8% студентів відносяться до неврівноваженого типу;
- Не вдалося віднести до того чи іншого типу 11.7% студентів;
- 19.7% — студенти рухливого типу;
- Слабкий тип зустрівся у 19.5% випадків.

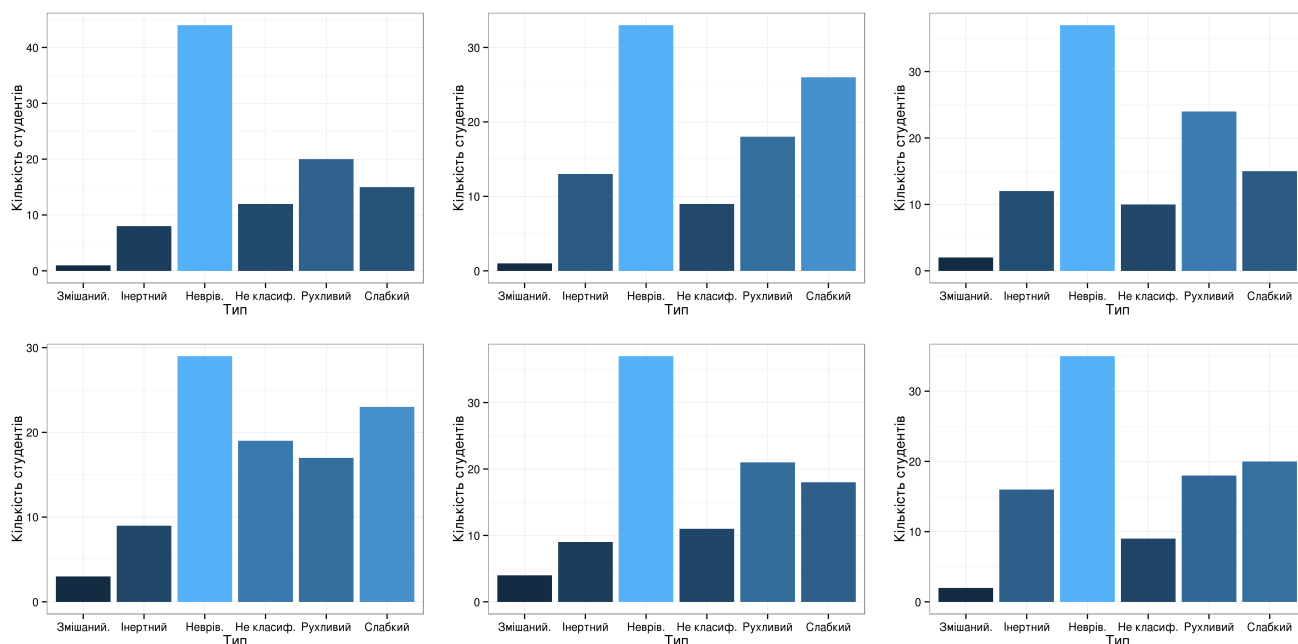


Рисунок 2.3 — Результати класифікації

2.2 Моделювання результатів виконання тестових завдань

2.2.1 Інформаційний метод прогнозування часу розв’язання завдань

Один з методів прогнозування часу розв’язання завдання людиною, що використовується в інженерній психології, є інформаційний метод. Вважаємо, що студент виконує завдання одне за одним. Кожне завдання має певну складність H_i — кількість інформації, що потрібно обробити для розв’язку завдання. Студент виконує кожне завдання з певною швидкістю V_i . Тоді час виконання тесту τ буде рахуватися за формулою

$$\tau = \sum_{i=1}^l \frac{H_i}{V_i},$$

де l — кількість задач. [8]

Вважаємо, що швидкість виконання завдання i рахується за формулою

$$V_i(t) = \frac{1}{\tau_i},$$

де τ_i — випадкова величина, що має показниковий розподіл з параметром λ_i , який залежить від типу вищої нервової діяльності людини та часу, що затрачено на виконання попередніх завдань:

$$\tau_i \sim \text{Exp}(\lambda_i).$$

Тоді τ_i буде часом, що витрачається на одиницю інформації завдання

$$\tau = \sum_{i=1}^l H_i \cdot \tau_i.$$

Якщо всі завдання мають однакову складність $H_i = H$, рівність приймає вигляд

$$\tau = H \cdot \sum_{i=1}^l \tau_i.$$

Щоб застосувати отримані з моделювання теппінг-тесту результати, потрібно горизонтально масштабувати параболи, до яких було апроксимовано ламані. Початкова парабола y має вигляд

$$y(t) = a^2 \cdot t + b \cdot t + c.$$

Потрібно отримати параболу λ

$$\lambda(t) = A^2 \cdot t + B \cdot t + C$$

таку, для якої виконується

$$\begin{aligned}\lambda(0) &= y(0), \\ \lambda(T) &= y(t_t), \\ \lambda\left(\frac{T}{2}\right) &= y\left(\frac{t_t}{2}\right)\end{aligned}$$

t_t — час виконання теппінг-тесту (30 секунд), T — час виконання тестових завдань (40 або 90 хвилин). Маємо розв'язок

$$\begin{aligned}A &= a \cdot \left(\frac{t_t}{T}\right)^2, \\ B &= b \cdot \frac{t_t}{T}, \\ C &= c,\end{aligned}$$

тобто

$$\lambda(t) = a \cdot \left(\frac{t_t}{T}\right)^2 \cdot t^2 + b \cdot \frac{t_t}{T} \cdot t + c.$$

Отже маємо

$$\begin{aligned}\tau_1 &\sim \text{Exp}\{\lambda(0)\}, \\ \tau_i &\sim \text{Exp}\left\{\lambda\left(\sum_{j=1}^i H_j \cdot \tau_j\right)\right\}, \quad i > 1.\end{aligned}$$

2.2.2 Моделювання тесту

Було змодельовано тестування на півтори години з 10 задачами, кожна з яких має складність $H = 0.5$. Номер задачі, яку виконує студент на даний момент, фіксується кожні 10 секунд. Для застосування методу головних компонент було змодельовано проходження тестування 600 студентами.

На рис. 2.4 можна побачити середньоквадратичні відхилення перших десяти головних компонент. Майже половину інформації (43.22%) можна отримати з першої компоненти.

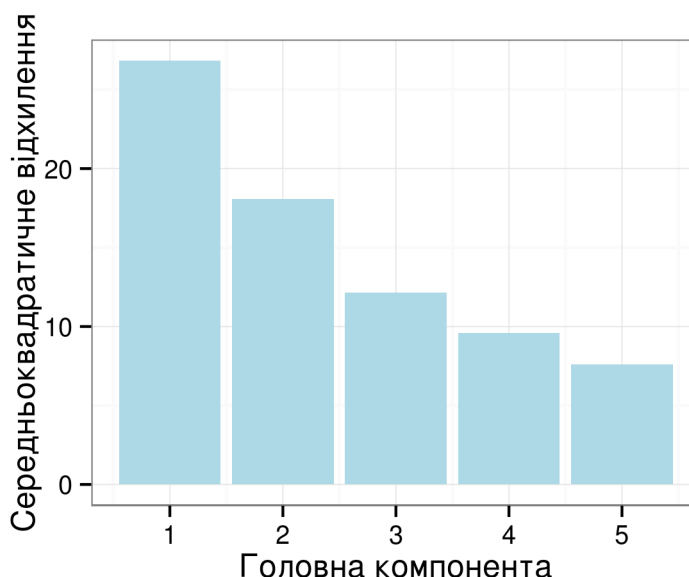


Рисунок 2.4 — Дисперсії першого десятку головних компонент

На рис. 2.5 наведено гістограми перших чотирьох головних компонент. Але з рис. 2.6 видно, що доцільно використовувати лише першу компоненту, а також не вдасться відрізнити від інших класів змішаний та неврівноважений тип.

На рис. 2.7 зображено склад першої головної компоненти з усіма класами та без змішаного і неврівноваженого типів.

За допомогою алгоритму CART (Classification and Regression Tree) було класифіковано дві групи студентів:

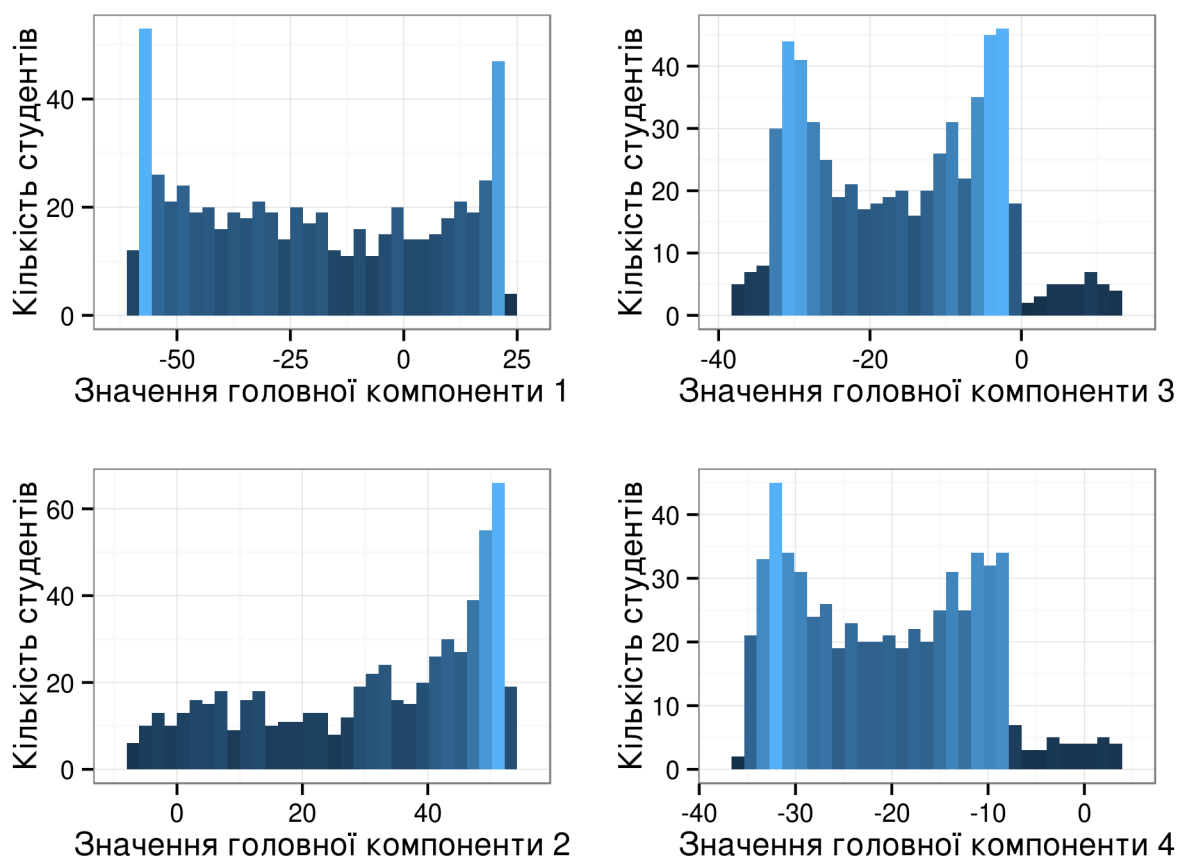


Рисунок 2.5 — Гістограми першої четвірки головних компонент

- 1) Ті, кому треба більше часу думати і не поспішати при виконанні завдань, щоб запобігти помилок — рухливий клас, а також не класифіковані студенти, адже в основному не вдалося віднести до того чи іншого типу тих студентів, активність яких не спадає з часом.
- 2) Ті, кому треба більше уваги приділяти елементарним задачам, щоб довести до автоматизму їх розв’язання і не витратити час на зайві дії — це студенти, що відносяться до слабкого та інертного типу.

Дерево класифікації зображено на рис. 2.8. Як можна побачити, критерій досить простий:

- ті студенти, перша головна компонента яких перевищує поріг, відносяться з ймовірністю 84% до тих, яким треба більше тренуватися;
- ті студенти, перша головна компонента яких не перевищує поріг, віднося-

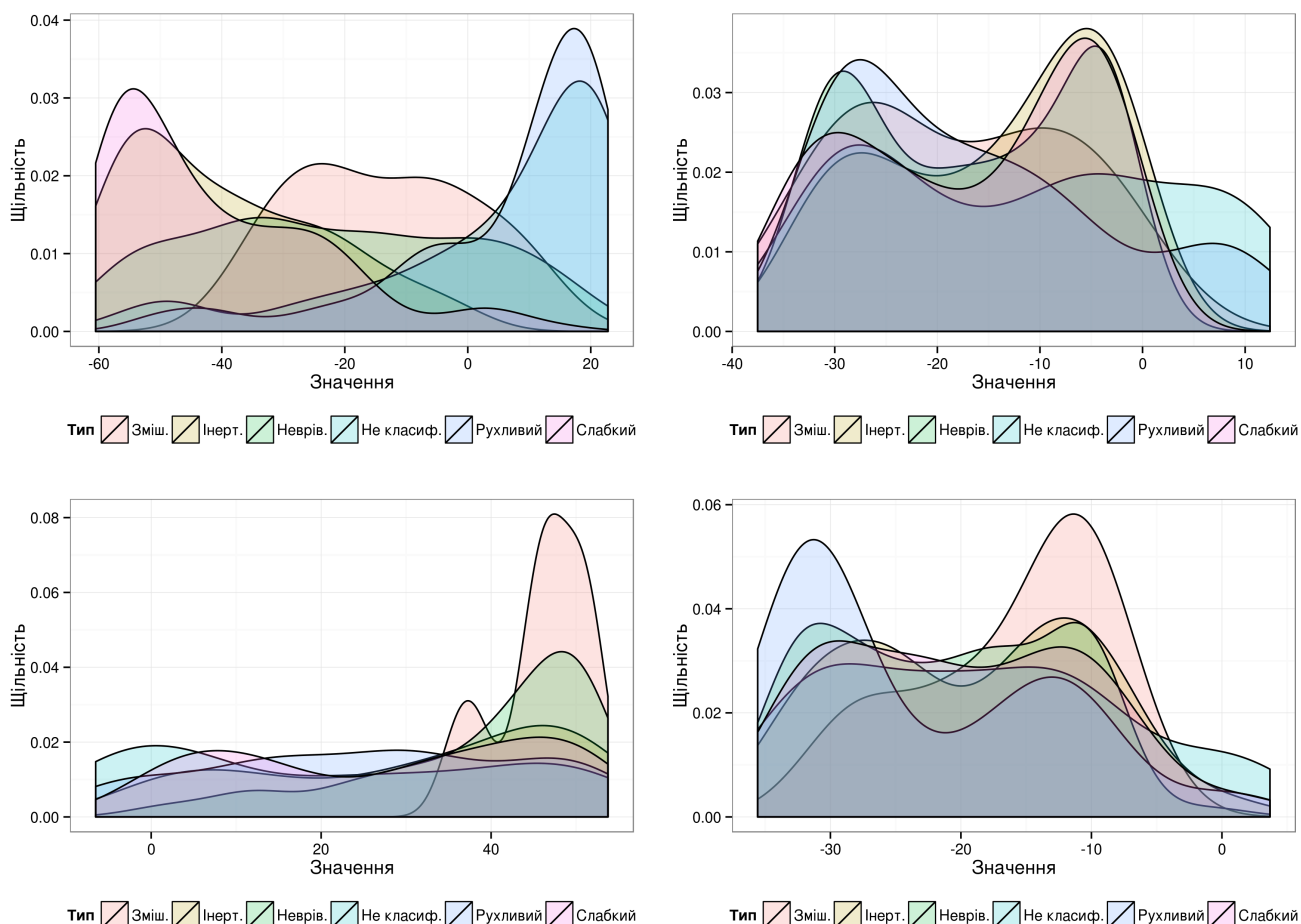


Рисунок 2.6 — Деталізовані гістограми першої четвірки головних компонент

ться з ймовірністю 94% до тих, яким треба більше думати.

Для перевірки результатів на поточній матриці повороту було класифіковано за першою компонентою ще вибірки по 600 студентів. В цих вибірках було вірно класифіковано людей, яким треба більше думати, з наступною точністю: 82%, 83%, 80%, 78%. Тих, яким треба більше працювати, тех було класифіковано з достатньою точністю: 94%, 94%, 93%, 96%. На рис. 2.9 зображено гістограми першої компоненти для кожної виборки.

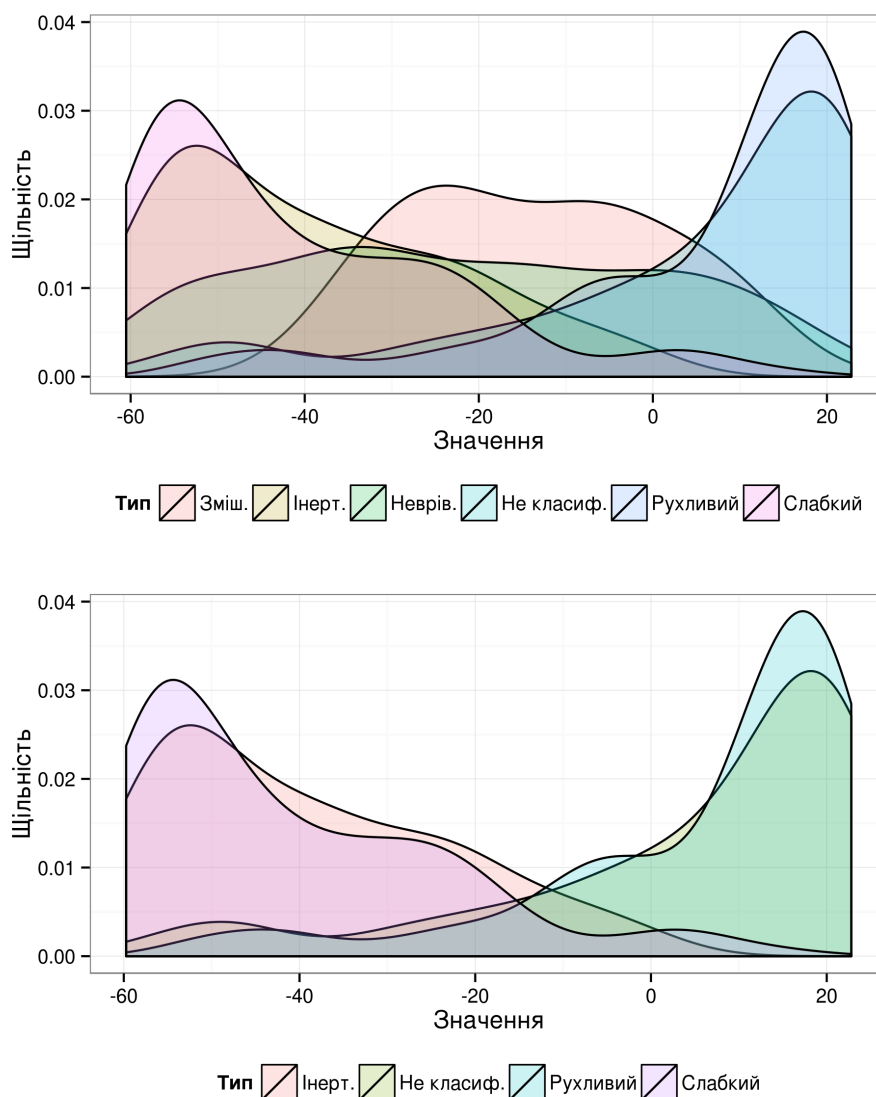


Рисунок 2.7 — Деталізовані гістограми першої головної компоненти

Висновки до розділу 2

В даному розділі за допомогою критерію Пірсона χ^2 було синтезовано параметри генератора моделей студентів, за допомогою якого було змодельовано студентів різних типів, а саме змішаного, інертного, невірноваженого, рухливого та слабкого. Також виявилися студенти, яких не вдалося віднести до жодної групи, але їх виявилось не так багато, щоб це негативно впливало на якість виборки.

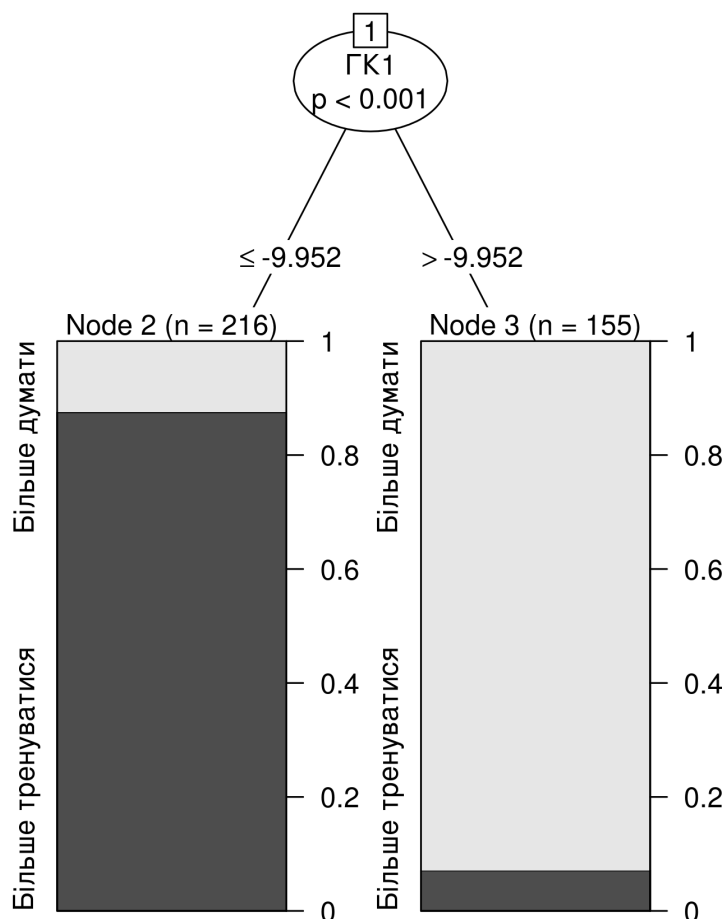


Рисунок 2.8 — Дерево класифікації

Маючи додатковими знаннями з інженерної психології та експертну оцінку в області психології вдалося побудувати модель проходження тестування людьми різних типів. Результати було проаналізовано за допомогою метода головних компонент. В ході аналізу виявилось, що найбільшу кількість корисної інформації містить лише перша компонента, за допомогою якої вдалося з достатньою точністю надати вірні поради студентам. Не вдалося знайти особливості студентів, що належать змішаному та неврівноваженому типів, які б дозволяли виявляти їх з достатньою точністю, при цьому не погіршуючи результати для

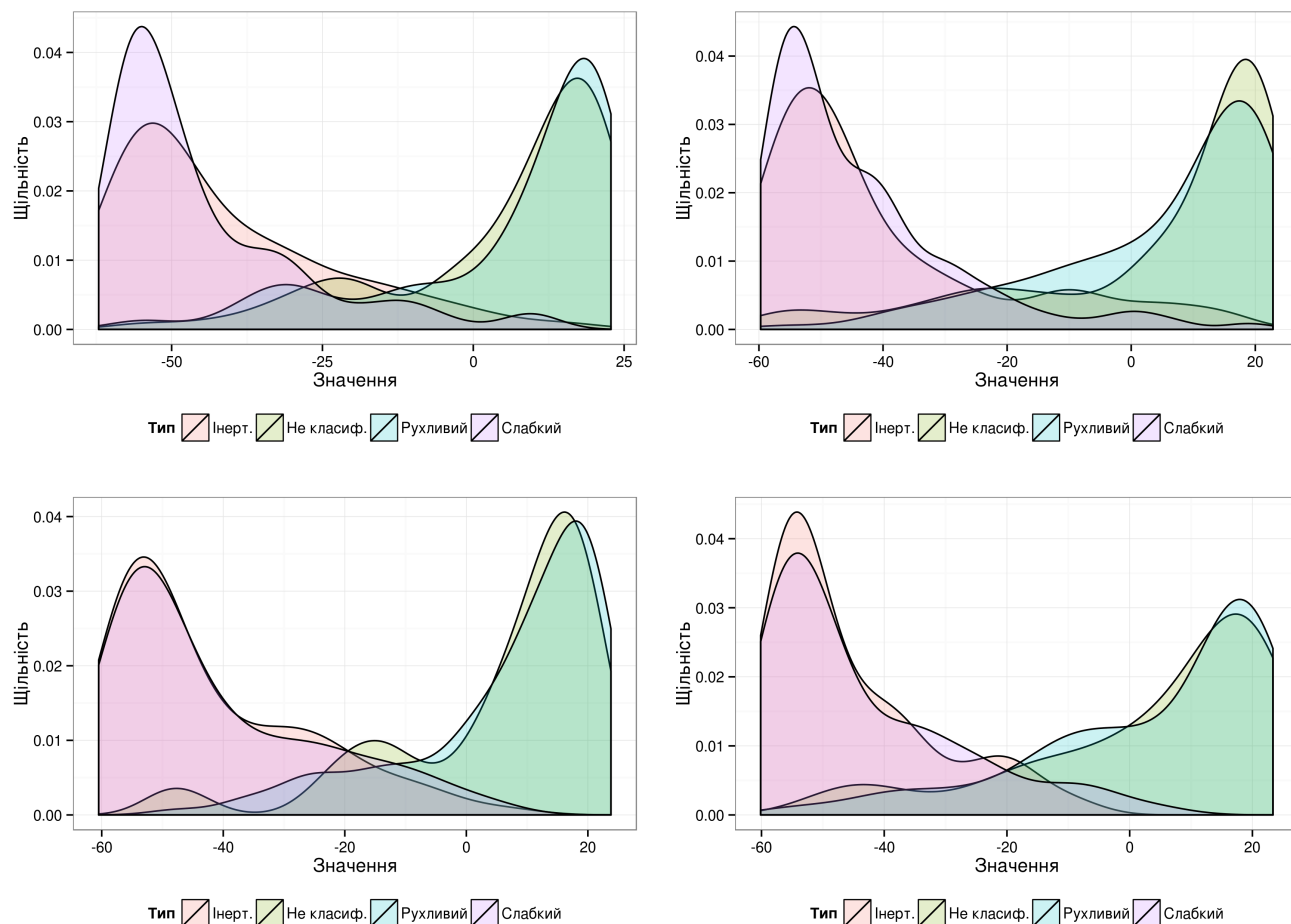


Рисунок 2.9 — Інші виборки

інших типів.

На отриманій матриці повороту та дереві класифікації було проаналізовано ще кілька вибірок. Аналіз показав, що перша компонента майже не змінює свого розподілу, що дозволяє і наладі класифікувати студентів з достатньою точністю.

3 ОХОРОНА ПРАЦІ

Робота відбувається у робочому приміщенні. У цьому приміщенні розроблялася математична модель психологічних особливостей людини на основі результатів тестування. В приміщенні розташовано один комп'ютер класу ноутбук Fujitsu LH532, Intel Core i3-2328M 2.2GHz, nVidia GeForce GT 620M, жорсткий диск 500 ГБ з монітором 1366x768 типу HD LED.

3.1 Технічні рішення щодо безпечного виконання роботи

В приміщенні працює одна людина, яка працює за описаним вище ноутбуком. Параметри кімнати: довжина — 7 метрів, ширина — 5 метрів, висота — 3 метри, площа — 35 метрів квадратних, об'єм 105 — метрів кубічних.

Оскільки у кімнаті працює одна людина, то на неї відводиться уся площа 35 метрів квадратних і об'єм 105 метрів кубічних. Згідно [9] і [10] обсяг простору для однієї людини в лабораторії з ЕОМ повинен бути не менше 20 метрів кубічних, а площа — не менше 6 метрів квадратних. Оскільки фактичні параметри перевищують нормативні, то приміщення задовольняє перерахованим вимогам.

Робочі місця знаходяться не менш ніж на один метр від стін з вікнами. Габарити робочого місця операторів ЕОМ: висота стола — 800 мм, ширина — 2500 мм, глибина — 1200 мм. Простір для ніг складає: висота — 650 мм, ширина — 1000 мм, глибина на рівні колін — 550 мм, а на рівні простягнутої ноги — 1200 мм. Робоче крісло оператора належить до підйомно-поворотного типу, з регулюванням висоти та повороту спинки. Висота спинки та її ширина мають розміри 750 мм та 550 мм відповідно. Крісло також обладнане стаціонарними підлокітниками довжиною 260 мм та шириною 70 мм.

Екран монітору розташовано на відстані 550 мм від очей оператора, а га-

барити робочого місця дозволять за необхідності розмістити його ще далі. Конструктивні особливості клавіатури передбачають її вільне переміщення з можливістю регулювання кута нахилу.

3.2 Електробезпека

Робоче приміщення обладнано стандартною трипровідною однофазною електричною мережею змінного струму з напругою 220 В.

Через те, що температура час від часу підвищується до 26 – 27С, а відносна вологість знаходиться на рівні 40 – 50%, приміщення відноситься до групи приміщень без підвищеної небезпеки, адже температура нижче 30С, а вологість не перевищує 60%.

Споживачі електроенергії в приміщенні — освітлювальні прилади і обчислювальна техніка (ноутбук).

Всі прилади, що працюють від електромережі, занулені. Електрична щитова знаходиться в кімнаті, що полегшує вимикання струму в разі нагальної потреби. Корпуси сучасних ноутбуків виготовлені із пластмаси, тому ураження струмом неможливе.

Уся електропроводка та мережеві кабелі знаходяться в закритому положенні у спеціальних пластикових закритих коробках. Провід марки ВВП 3 × 1.5 з мідними струмопровідними жилами з ПВХ пластика; плоский, з розділювальною основою. Номінальна змінна напруга 220В, частота — до 50 Гц.

Запобіжна система на трансформаторі спрацьовує при силі струму в 27 А. Усі параметри не перевищують граничних меж.

3.3 Пожежна безпека

Конструкція будівлі, де знаходиться приміщення, виконана із залізобетонних плит, тобто його конструктивні елементи не спалювані. Будівля відноситься до II ступеня вогнестійкості. Межа вогнестійкості конструкції приблизно 0.5 – 2.5 год.

Згідно до [11] по вибухонебезпечній і пожежній небезпеці приміщення відноситься до категорії В, тому що в ньому перебувають важкогорючі тверді й волокнисті речовини й матеріали: стіл із дерева, ноутбук, периферія, лінолеум та ін. Оскільки в приміщенні знаходяться тверді горючі речовини та матеріали, немає горючих рідин і не накопичуються горючий пил, за класифікацією воно належить до класу П-Па.

Для підвищення безпеки, зокрема і пожежної, мережа напругою з 220 В є проводом з мідною жилою. Розташовується у вінілопластикових трубах, що прокладені в стінах. Ізоляція проводів розрахована на напругу в 1.5 кВ.

В приміщенні встановлена пожежна сигналізація СД121-5, оповісник ІПК-1, комбінований з пульсуючою індикацією, що реагує на задимленість та температуру. Сигналізація приміщення підключена до централізованого інформаційного пункту, від якого у випадку виникнення пожежі сигнал передається на пожежну частину. Крім цього біля виходу з приміщення знаходиться вогнегасник типу ОУ-2. Це не відповідає вимогам пожежної безпеки згідно до [12], адже потрібно не менше двох вогнегасників на 20 квадратних метрів, отже на дане приміщення площею 35 квадратних метри потрібно як мінімум чотири вогнегасники типу ОУ-2.

Для приміщення, в якому працюють менш ніж 25 людей, та відстань від будь-якого робочого місця не перевищує 25 метрів згідно норм досить одного виходу евакуації. Параметри евакуаційного виходу відповідають встановленим

нормам. Двері відкриваються назовні. При нормі не менше 2 м ширина коридору 2.2 м. Висота до перекриття 3 м при нормі не менше 3 м. Висота дверей у коридорі 2 м, а ширина 1.8 м, що також відповідає нормі. Ширина дверей рівна 0.8 м (норма — 0.8 м).

3.4 Виробничий шум

Приміщення знаходиться в будівлі, яка перебуває на значній відстані від вулиці; гучних об'єктів поблизу немає. Також у приміщенні встановлені звукоізолюючі металопластикові вікна, а система кондиціонування спроектована таким чином, що гучні її елементи знаходяться на інших поверхах будівлі. Отже, джерелами шуму є лише комп'ютерна техніка, а точніше деякі її складові.

За результатами вимірювання звукового шуму на робочому місці, приладом Шумомір ВШВ-003-М2, рівень шуму коливався в межах від 25 до 35 дБА, що не перевищує поріг у 50 дБА та задовольняє санітарним нормам [10].

3.5 Виробниче освітлення

При незадовільному освітленні зменшується продуктивність праці користувачів персональних комп'ютерів.

Розробка дипломного проекту відбувається в приміщенні з двома вікнами, що забезпечує природне бокове освітлення, яке становить близько 400 лк.

У ролі штучного освітлення виступають 3 світильника з сумарно кількістю в 9 люмінесцентних ламп ЛБ 36, згідно технічних характеристик даного типу ламп, світловий потік рівний 3050 лм. Кожен світильник містить по три лампи. Розрахуємо фактичну освітленість робочого приміщення для порівняння з

нормою, враховуючи відстань кожного світильника та кількості ламп у ньому:

$$E_n = I/R^2 \times \cos \alpha = F/(4 \times \pi \times R^2),$$

$$E_1 = 3050/(4 \times 3.14 \times 4) \times \cos 45^\circ = 42.7\text{лк},$$

$$E_2 = 3050/(4 \times 3.14 \times 6.25) \times \cos 60^\circ = 32.7\text{лк},$$

$$E_3 = 3050/(4 \times 3.14 \times 2.25) \times \cos 75^\circ = 104\text{лк},$$

де I - сила світла, R - відстань до джерела світла, Φ - світловий потік, α - кут падіння променя світла.

Сумарне значення буде рівним

$$E_1 \times 3 + E_2 \times 3 + E_3 \times 3 = 42.7 \times 3 + 32.7 \times 3 + 104 \times 3 = 538.2\text{лк}$$

Для даного типу роботи, згідно [13], нормативне значення освітленості дорівнює $E_n = 500\text{лк}$. Оскільки $E_n < E$, то штучне освітлення робочого приміщення відповідає санітарним нормам.

3.6 Мікроклімат

Якість повітря в приміщенні впливає на самопочуття людини. У приміщеннях з комп'ютерною технікою, внаслідок того що третина споживаної нею енергії розсіюється у вигляді тепла, формуються специфічні умови мікроклімату а саме, підвищується температура до 26 – 27 С, відносна вологість падає до 40%.

У виробничих приміщеннях на робочих місцях з персональним комп'ютером мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й швидкості руху повітря. [10]

Розробка дипломного проекту відноситься до категорії легких — 1А робіт.

В холодний період температура тримається на рівні 21-23 (норма — 22-24), вологість 50-55% (норма — 40-60%), а швидкість руху повітря 0.04 — 0.08 м/с (норма — < 0.1 м/с). В теплий період температура тримається на рівні 22-25 (норма — 23-25), вологість 55-60% (норма — 40-60%), а швидкість руху повітря 0.03 — 0.08 м/с (норма — < 0.1 м/с). Для забезпечення постійного параметру мікроклімату в приміщенні встановлений побутовий кондиціонер, що вмикається за необхідністю. Отже, мікроклімат приміщення задовільняє норми наведені у [14].

3.7 Організація оптимального режиму праці та відпочинку

При роботі з ПК для підтримання продуктивної діяльності робітника та запобіганню професійних захворювань передбачаються регламентовані перерви для відпочинку у спеціально відведених для цього місцях. До таких місць належить гімнастичний зал, де можна виконувати вправи для підтримання тону опорно-рухової системи, адже в основному робота несе сидячий характер. При виконанні протягом робочої зміни робіт, що відносяться до різних видів трудової діяльності, за основну роботу з засобами обчислювальної техніки слід приймати таку, яка займає не менше 50% часу протягом робочого дня. Для забезпечення високого рівня праці організаційно передбачено:

- 1) перерви для вживання їжі (обідні перерви);
- 2) перерви для відпочинку й особистих потреб;
- 3) проведення вправ для очей через кожні 20-25 хвилин роботи;
- 4) проведення під час перерв наскрізного провітрювання приміщень;
- 5) здійснення під час перерв вправ фізкультурної паузи протягом 15-20 хвилин.

б) Перерви для відпочинку через кожну годину тривалістю 10 хвилин

Для зменшення негативного впливу монотонної роботи з ПК передбачено чергування операцій осмисленого тексту і числових даних, чергування редагування тексту та введення даних. Також, для зменшення нервового та емоційного рівнів напруження та зняття втоми з очей організаційно передбачено проведення під час перерв спеціальних комплексів вправ.

3.8 Склад повітря робочої зони

Основним джерелом забруднення повітря робочої зони є пил. Джерелами пилу в приміщенні є: документи, папір, книги, одяг, взуття, зовнішнє повітря. За допомогою пиломіру ФПГ-6 було встановлене фактичне значення концентрації пилу 4мг/м^3 , що не перевищило значення ГДК для нейтрального пилу, що не має отруйних властивостей, яке рівне 10мг/м^3 . Тому негативна дії на людину не здійснюється. Задля очищення приміщення від пилу проводиться періодичне вологе або сухе прибирання та провітрювання приміщення. Вуглекислий газ, що утворюється від дихання людей, видаляється вентиляцією.

3.9 Виробничі випромінювання

Джерелом випромінювань під час роботи є монітор ноутбука. Вид випромінювань — електромагнітні (ЕМ), ультрафіолетові, інфрачервоні, діапазон випромінювань — 5Гц–2ГГц.

Згідно з [14] гранично допустиме значення електромагнітного поля комп'ютера 25В/м . Монітор, що використовується на робочому місці, має електромагнітне поле не більше 1В/м . Для вимірювання використано вимірювач напруженості електричного і магнітного поля промислової частоти (50 Гц) ПЗ-50.

Дані вимоги дотримані на робочому місці і відповідають нормативним документам.

Висновки до розділу 3

У даному розділі досліджені питання охорони праці при роботі з візуальними дисплейними терміналами ЕОМ. Були виявлені й вивчені небезпечні й шкідливі виробничі фактори. Взагалом було виявлено, що обсяг і площа приміщення відповідають нормативним значенням, параметри робочого місця задовольняють нормам, показники мікроклімату цілком відповідають нормативним вимогам, відповідність нормам по природньому освітленні, наявність штучного освітлення, акустичні умови роботи в нормі, при дотриманні правил електробезпеки виключається поразки персоналу електричним струмом, але приміщення не повністю задовольняє нормам пожежної безпеки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Айвазян, С.А. Прикладная статистика: Классификация и снижение размерности : Справочное издание / С.А. Айвазян. — М.: Финансы и статистика, 1989. — 606 с.
2. Боровков, А. А. Математическая статистика / А. А. Боровков // Учебники для ВУЗов. Специальная литература. — СПб.: Лань, 2010. — 705 с.
3. Sturges, Herbert A. The Choice of a Class Interval / Herbert A. Sturges // *J-J-AM-STAT-ASSOC.* — 1926. — March. — Vol. 21, no. 153. — Pp. 65–66.
4. Павлов, И.П. Двадцатилетний опыт объективного изучения высшей деятельности (поведения) животных / И.П. Павлов. — М.: Рипол Классик, 1973. — 659 с.
5. Ильин, Е.П. Дифференциальная психофизиология / Е.П. Ильин // Серия Учебник нового века. — СПб.: Питер, 2001. — 464 с.
6. Булинский, А.В. Теория случайных процессов / А.В. Булинский, А.Н. Ширяев // Теория вероятностей. Математическая статистика : ТВМС. — М.: Физматлит, 2003. — 399 с.
7. Kingman, J.F.C. Poisson Processes / J.F.C. Kingman // Oxford studies in probability. — Oxford: Clarendon Press, 1992. — 112 pp.
8. Ломов, Б. Ф. Справочник по инженерной психологии / Б. Ф. Ломов. — Москва: Машиностроение, 1982. — 368 с.
9. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно-обчислювальних машин.

10. ДСанПіН 3.3.2-007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами ЕОМ.
11. ОНТП 24-86 Определение категорий помещений и зданий по взрывопожарной и пожарной опасности.
12. ППБУ-95 Правила пожежної безпеки України.
13. ДБН-В.2.5-28-2006 Природне і штучне освітлення.
14. ДСН 3.3.6.042-99 Державні санітарні норми параметрів мікроклімату у виробничих приміщеннях.

ДОДАТОК А

Лістинг коду програми, яка моделює результати теппінг-тесту

```

1  source("generate_sequence.r")
2  source("analyze_approximation.r")
3  wideScreen()
4  # Read command line arguments
5  args <- commandArgs(TRUE)
6  display_approximation <- FALSE
7  if (length(args) > 0 && args[1] == "approximation") {
8      display_approximation <- TRUE
9  }
10 # Init cells for charts
11 n <- 3
12 m <- 3
13 png('output.png', width=18, height=22, units="cm", res=300)
14 par(mfrow=c(n,m))
15 chart.xlab <- "Час"
16 chart.ylab <- "Кількість постукувань"
17 chart.ylim <- c(20,50)
18 # Parameters
19 sample.time <- seq(5, 30, 5)
20 sample.size = length(sample.time)
21 sample.prediction_degree = 2
22 sample.poly = poly(sample.time, sample.prediction_degree, raw=TRUE)
23 lambda.default = c(40, 41, 46, 38, 35, 34)
24 # Fill every cell
25 for (i in 1:(n*m)) {
26     # Generate non-homogeneous Poisson process trajectory
27     sample.current <- generate_trajectory(lambda.default)
28     # Plot
29     plot(sample.time, sample.current, xlab=chart.xlab, ylab=chart.ylab,
30         ylim=chart.ylim)
31     lines(sample.time, sample.current)
32     if (display_approximation) {
33         # Calculate approximation
34         abc <- find_abc(sample.current)
35         f <- function(t) {
36             return(abc[1]*t^2+abc[2]*t+abc[3])
37         }
38         # Draw approximation
39         lines(sample.time, Map(f, sample.time), xlab=chart.xlab, ylab=chart.ylab,
40             ylim=chart.ylim, col='red')
41         # Draw start rhythm
42         #abline(h=sample.model$coefficients[3]*sample.time[1]^2 +
43             #      sample.model$coefficients[2]*sample.time[1] +
44             #      sample.model$coefficients[1], col='green', lty=2, lwd=1)
45         # Draw minumum and maximum

```



```
46         #abline(v=extrema[['min']], col='gray', lty=3, lwd=1)
47         #abline(v=extrema[['max']], col='blue', lty=3, lwd=1)
48         # Display group
49         #print(get_group(sample.model, sample.time,
50             #           extrema[['min']], extrema[['max']]))
51     }
52 }
53 dev.off()
```

ДОДАТОК Б

Лістинг коду модуля, що генерує процес Пуассона

```

1 generate_trajectory <- function(lambda.default) {
2     # Generate non-homogeneous Poisson process
3     return(rpois(length(lambda.default), lambda.default))
4 }

```

Лістинг коду модуля з функцією, що розтягує параболу

```

1 t.stretch.start <- 0
2 t.stretch.finish <- 30
3
4 parabola.stretch <- function (a, b, c, t.finish.new) {
5     t.k <- t.stretch.finish/t.finish.new
6     function (t) {
7         x <- t.k * t
8         a*x^2+b*x+c
9     }
10 }

```

Лістинг коду модуля з функцією, що рахує відстань для критерію Пірсона χ^2

```

1 chi_squared.distance <- function(m, h, p) {
2     return(m * sum((h - p)^2 / p))
3 }

```

ДОДАТОК В

Лістинг коду програми, яка класифікує моделі студентів за типами вищої нервової діяльності

```

1  t.start <- 5
2  t.middle <- 15
3  t.finish <- 30
4  t.diff <- t.finish - t.start
5
6  X      <- seq(5, 30, 5)
7  X2     <- X^2
8  X3     <- X^3
9  X4     <- X^4
10 N      <- 6
11
12 SX     <- sum(X)
13 SX2    <- sum(X2)
14 SX3    <- sum(X3)
15 SX4    <- sum(X4)
16 SN     <- N
17
18 delta <- SX2 * (SX2^2 - SX * SX3) - SX * (SX3 * SX2 - SX * SX4) +
19         SN * (SX3^2 - SX2 * SX4)
20
21 delta.a.k <- c(SX2^2 - SX * SX3, -(SX * SX2 - SN * SX3), SX * SX - SN * SX2)/delta
22
23 delta.b.k <- c(-(SX3 * SX2 - SX * SX4), SX2^2 - SN * SX4, -(SX2 * SX - SN * SX3))/delta
24
25 delta.c.k <- c(SX3^2 - SX2 * SX4, -(SX2 * SX3 - SX * SX4), SX2 * SX2 - SX * SX3)/delta
26
27 #MX <- matrix(c(SX2, SX3, SX4, SX, SX2, SX3, SN, SX, SX2), nrow=3, ncol=3)
28
29 find_abc <- function(sample) {
30     SY.k <- c(sum(sample), sample %>% X, sample %>% X2)
31     return(c(SY.k %>% delta.a.k, SY.k %>% delta.b.k, SY.k %>% delta.c.k))
32 }
33
34 get_group <- function(a, b, c) {
35     EPSILON <- 2
36     if (a == 0) {
37         diff <- - b * t.diff
38         if (diff < -EPSILON) {
39             return(1)
40         } else if (diff < EPSILON) {
41             return(0)
42         }
43         return(-1)

```

```

44     }
45
46     d1 <- (2*a*t.start + b)
47     d6 <- (2*a*t.finish + b)
48     M <- sign(a)
49     V <- sign(d1*d6)
50     t.g <- - b / (2*a)
51     sigma.s <- abs((t.g-t.start) * d1)/2
52     sigma.f <- abs((t.g-t.finish) * d6)/2
53     sigma.m <- max(sigma.s, sigma.f)
54     sigma.p <- sigma.m - min(sigma.s, sigma.f)
55
56     if ((V == -1 && sigma.m < EPSILON) ||
57         (V == 1 && sigma.p < EPSILON)) {
58         return(0)
59     } else if (V == 1 && sigma.p > EPSILON && (
60         (t.g < t.start && M == -1) ||
61         (t.g > t.finish && M == 1) ||
62         (a == 0 && b < 0))) {
63         return(1)
64     } else if (M == -1 && V == -1 && t.g < t.middle &&
65         sigma.f > EPSILON) {
66         return(2)
67     } else if (M == -1 && V == -1 && t.g >= t.middle &&
68         sigma.f - sigma.s > 0) {
69         return(3)
70     } else if (M == 1 && V == -1 && (sigma.s - sigma.f >= -EPSILON)) {
71         return(4)
72     } else {
73         return(-1)
74     }
75 }

```

ДОДАТОК Г

Лістинг коду програми, яка генерує студентів та будує гістограми

```

1  library('ggplot2')
2
3  source("generate_sequence.r")
4  source("analyze_approximation.r")
5  source("chi_squared.r")
6  source("exam_draw.r")
7
8  n      <- 100
9  rows   <- 3
10 columns <- 2
11 sample.time <- seq(5, 30, 5)
12 sample.size = length(sample.time)
13 sample.prediction_degree = 2
14 sample.poly = poly(sample.time, sample.prediction_degree, raw=TRUE)
15 lambda.default = c(40, 41, 46, 38, 35, 34)
16 needed.percentage = c(.1, .03, .25, .35, .25, .12)
17
18 groups.names <- c("Не класиф.", "Змішаний.", "Слабкий",
19                  "Неврів.", "Рухливий", "Інертний")
20 groups.number <- length(groups.names)
21
22 global.groups = rep(0, groups.number)
23 global.percentage = rep(0, groups.number)
24 global.distances = rep(0, groups.number)
25
26 result = rep(-1, n)
27 plots <- list()
28
29 for (j in 1:(rows*columns)) {
30   for (i in 1:n) {
31     ## Generate non-homogeneous Poisson process trajectory
32     sample.current <- generate_trajectory(lambda.default)
33     # Calculate approximation
34     abc <- find_abc(sample.current)
35     # Get group
36     result[i] <- get_group(abc[1], abc[2], abc[3])
37   }
38   elements_in_group <- function(x) length(result[result==x])
39   groups <- unlist(Map(elements_in_group, -1:(groups.number-2)))
40   print(sprintf("%.2f", groups/n))
41   print(sprintf("%.2f", chi_squared.distance(n, tail(groups/n, 5), tail(needed.percentage,5))))
42   global.groups <- global.groups + groups
43
44   info <- data.frame(group=groups.names[result+2], stringsAsFactors=FALSE)
45   plots[[j]] <- ggplot(info, aes(group)) +

```

```

46     geom_histogram(aes(fill=..count..)) + theme_bw() +
47     theme(legend.position="none") +
48     labs(x="Тип", y="Кількість студентів") +
49     scale_fill_gradient("Кількість")
50 }
51
52 png('types.png', width=40, height=20, units="cm", res=300)
53 print(multiplot(plotlist=plots, cols=3))
54 dev.off()
55
56 print('Result:')
57 global.percentage = global.groups / (n*rows*columns)
58 print(sprintf("%.2f", global.percentage))
59 print(sprintf("%.2f", chi_squared.distance(n, tail(global.percentage, 5), tail(needed.percentage,5))))
60
61 variables.names <- c("NotClassified", "Neutral", "Weak", "Unstable",
62                     "Movable", "Inert")
63 file.conn <- file("poisson_types.tex")
64 file.out <- rep(NA, groups.number)
65 for (i in 1:groups.number) {
66     file.out[i] <- sprintf("\\def\\poisson%s{%.1f}", variables.names[i], global.percentage[i]*100)
67 }
68 writeLines(file.out, file.conn)
69 close(file.conn)

```

ДОДАТОК Д

Лістинг коду програми, яка синтезує параметр λ

```

1  library("iterators")
2  library("parallel")
3  library("foreach")
4  library("doParallel")
5
6  source("generate_sequence.r")
7  source("analyze_approximation.r")
8  source("chi_squared.r")
9
10 n      <- 1000
11 rows   <- 3
12 columns <- 2
13 sample.time <- seq(5, 30, 5)
14 sample.size = length(sample.time)
15 sample.prediction_degree = 2
16 sample.poly = poly(sample.time, sample.prediction_degree, raw=TRUE)
17 lambda.default = rep(40, 6)
18
19 groups.number <- 4
20 needed.percentage = c(.03, .25, .35, .25, .12)
21
22 dist.min <- 20
23
24 calculate.lambda <- function(lambda.current) {
25   result = rep(-1, n)
26   for (i in 1:n) {
27     ## Generate non-homogeneous Poisson process trajectory
28     sample.current <- generate_trajectory(lambda.current)
29     # Calculate approximation
30     abc <- find_abc(sample.current)
31     # Get group
32     result[i] = get_group(abc[1], abc[2], abc[3])
33   }
34   elements_in_group <- function(x) length(result[result==x])
35   groups <- unlist(Map(elements_in_group, 0:groups.number))
36   distance <- chi_squared.distance(n, groups/n, needed.percentage)
37   if (distance < dist.min) {
38     cat("Lambda:", sprintf("%d,", lambda.current),
39         sprintf("Distance: %.2f,", distance),
40         "Groups:", sprintf("%.2f,", groups/n),
41         "\n")
42     dist.min <-<- distance
43   }
44   return()
45 }
```

```

46
47 iterate.lambdas.seq <- function(lambda.current, lambda.pos) {
48     if (lambda.pos == 2) {
49         lambda.min <- 0
50         lambda.max <- 5
51     } else if (lambda.pos == 3) {
52         lambda.min <- -5
53         lambda.max <- 10
54     } else if (lambda.pos == 4) {
55         lambda.min <- -15
56         lambda.max <- -5
57     } else if (lambda.pos == 5) {
58         lambda.min <- -5
59         lambda.max <- 0
60     } else if (lambda.pos == 6) {
61         lambda.min <- -5
62         lambda.max <- 0
63     }
64
65     get_lambda <- function (i) {
66         lambda.current[lambda.pos] <- lambda.current[lambda.pos-1] + i
67         return(lambda.current)
68     }
69     lambdas <- Map(get_lambda, lambda.min:lambda.max)
70     for (l in lambdas) {
71         iterate.lambdas(l, lambda.pos+1)
72     }
73     return()
74 }
75
76 iterate.lambdas <- function(lambda.current, lambda.pos) {
77     if (lambda.pos == 7) {
78         calculate.lambda(lambda.current)
79     } else {
80         iterate.lambdas.seq(lambda.current, lambda.pos)
81     }
82     return()
83 }
84
85 lambda.current <- lambda.default
86 lambda.min <- 0
87 lambda.max <- 5
88 get_lambda <- function (i) {
89     lambda.current[2] <- lambda.current[1] + i
90     return(lambda.current)
91 }
92 lambdas <- Map(get_lambda, lambda.min:lambda.max)
93 cl <- makeCluster(detectCores(), outfile="")
94 registerDoParallel(cl, cores = detectCores() - 1)

```



```
95 foreach(i=lambdas, .options.multicore=c(silent=FALSE)) %dopar% {
96     source("generate_sequence.r")
97     source("analyze_approximation.r")
98     source("chi_squared.r")
99     options(warn=1)
100     iterate.lambdas(i, 3)
101 }
102 stopCluster(cl)
```

ДОДАТОК Е

Лістинг коду модуля, який моделює результати тестування

```

1  source("generate_sequence.r")
2  source("modify_parabola.r")
3
4  options(warn=1)
5
6  lambda.default = c(40, 41, 46, 38, 35, 34)
7
8  exam.cpm      <- 6
9  exam.minutes <- 90
10 exam.length  <- exam.cpm * exam.minutes
11
12 task.N <- 10
13 task.H <- exam.length / (task.N*mean(lambda.default)*3)
14 tasks  <- rep(task.H, task.N)
15
16 exam.pass <- function (intensity) {
17   get.tau <- function (t) {
18     result <- rexp(1, intensity(t))
19     if (is.nan(result)) print(t)
20     result
21   }
22   exam.H.passed <- rep(NA, exam.length)
23   exam.H.passed[1] <- get.tau(1)
24   for (i in 2:exam.length) {
25     exam.H.passed[i] <- get.tau(i) + exam.H.passed[i-1]
26   }
27   (function () {
28     is.positive <- function (x) {
29       x > 0
30     }
31     .exam.row <- function (accumulator, task.n.current, exam.H.left) {
32       tmp <- exam.H.left - task.H
33       positive <- tmp[is.positive(tmp)]
34       negative <- tmp[Negate(is.positive)(tmp)]
35       if (length(positive) == 0 || task.n.current == task.N) {
36         c(accumulator, rep(task.n.current, length(negative)), rep(0, length(positive)))
37       } else {
38         .exam.row(c(accumulator, rep(task.n.current, length(negative))), task.n.current+1, positive)
39       }
40     }
41     .exam.row(c(), 1, exam.H.passed)
42   })()
43 }
44
45 exam.generate <- function (n) {

```

```

46     result <- c()
47     groups <- rep(-1, n)
48     values <- matrix(nrow=n, ncol=exam.length)
49     abc <- NA
50     for (i in 1:n) {
51         repeat {
52             abc <- find_abc(generate_trajectory(lambda.default))
53             if (abc[3] > 0) break
54         }
55         new.parabola <- parabola.stretch(abc[1], abc[2], abc[3], exam.length)
56         groups[i] <- get_group(abc[1], abc[2], abc[3])
57         values[i,] <- exam.pass(new.parabola)
58     }
59     list(values=values, groups=groups)
60 }

```

ДОДАТОК Є

Лістинг коду модуля, який аналізує результати тестування

```

1  library('party')
2
3  groups <- -1:4
4
5  groups.unclassified <- -1
6  groups.neutral <- 0
7  groups.weak <- 1
8  groups.unstable <- 2
9  groups.movable <- 3
10 groups.inert <- 4
11
12 groups.investigated <- c(groups.unclassified, groups.weak, groups.movable,
13                          groups.inert)
14
15 advice.think.more <- 'Більше думати'
16 advice.learn.more <- 'Більше тренуватися'
17
18 get.advice <- function(x) {
19   if (is.element(x, c(groups.unclassified, groups.movable))) {
20     advice.think.more
21   } else if (is.element(x, c(groups.weak, groups.inert))) {
22     advice.learn.more
23   } else {
24     NA
25   }
26 }
27
28 get.students <- function (sample, rotation) {
29   investigated.indices <- which(is.element(sample$groups,
30                                           groups.investigated))
31   advices <- unlist(Map(get.advice, sample$groups[investigated.indices]))
32   data.frame(group=advices, FK
33             1=sample$values[investigated.indices,]%*%rotation)
34 }
35
36 get.fit <- function (sample) {
37   ctree(group ~ FK1, data=sample, controls=ctree_control(maxdepth=1))
38 }
39
40 get.pc.proportion <- function (standard.deviation, number) {
41   standard.deviation[number]^2/sum(standard.deviation^2)
42 }
43
44 get.prediction.quality <- function (cart, exam) {
45   advices <- unlist(Map(get.advice, exam$groups))

```

```

46     indices <- which(Negate(is.na)(advices))
47     prediction <- predict(students.cart, data.frame(K1=exam$values[indices,] %*% students.pca$rotation[,1]))
48     think.more <- prediction[advices[indices] == advice.think.more] == advice.think.more
49     learn.more <- prediction[advices[indices] == advice.learn.more] == advice.learn.more
50     list(think=sum(think.more)/length(think.more),
51          learn=sum(learn.more)/length(learn.more))
52 }

```

ДОДАТОК Ж

Лістинг коду модуля, що містить функції для побудови графіків для результатів аналізу тестування

```

1  multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
2    library(grid)
3
4    # Make a list from the ... arguments and plotlist
5    plots <- c(list(...), plotlist)
6
7    numPlots = length(plots)
8
9    # If layout is NULL, then use 'cols' to determine layout
10   if (is.null(layout)) {
11     # Make the panel
12     # ncol: Number of columns of plots
13     # nrow: Number of rows needed, calculated from # of cols
14     layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
15                      ncol = cols, nrow = ceiling(numPlots/cols))
16   }
17
18   if (numPlots==1) {
19     print(plots[[1]])
20
21   } else {
22     # Set up the page
23     grid.newpage()
24     pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
25
26     # Make each plot, in the correct location
27     for (i in 1:numPlots) {
28       # Get the i,j matrix positions of the regions that contain this subplot
29       matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
30
31       print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
32                                       layout.pos.col = matchidx$col))
33     }
34   }
35 }
36
37 get.gg.students.pca <- function (pca, count) {
38   info <- data.frame(sdev=pca$sdev[1:count],n=1:count)
39   ggplot(info, aes(x=n,y=sdev)) +
40     geom_bar(stat="identity", fill="lightblue") + theme_bw() +
41     labs(x="Головна компонента", y="Середньоквадратичне відхилення")
42 }
43

```

```

44 get.gg.students.pc <- function (exam, rotations, pca.number) {
45     info <- data.frame(value=exam$values %*% rotations[,pca.number],
46                       group=exam$groups)
47     ggplot(info, aes(value)) +
48       geom_histogram(aes(fill=..count..)) + theme_bw() +
49       labs(x=sprintf("Значення головної компоненти %d", pca.number),
50          y="Кількість студентів") +
51       theme(legend.position="none") +
52       scale_fill_gradient("Кількість")
53 }
54
55 groups.names <- c("Не класифіковано", "Змішаний", "Слабкий",
56                  "Невріважений", "Рухливий", "Інертний")
57
58 get.gg.students.pc.classified <- function (exam, rotation, groups) {
59     pca.distribution <- Reduce(function (result, g) {
60         rbind(result, data.frame(Значення
61                                =c(exam$values[exam$groups == g,]%*% rotation),Тип
62                                =groups.names[g+2], stringsAsFactors=FALSE))
63     }, groups, c())
64     ggplot(pca.distribution, aesЗначення(, fillТип=)) + theme_bw() +
65       ylab("Щільність") +
66       theme(legend.position="bottom") +
67       geom_density(alpha = 0.2)
68 }

```

ДОДАТОК 3

Лістинг коду програми, яка моделює та аналізує результати тестування

```

1  library('ggplot2')
2  library('party')
3
4  source("generate_sequence.r")
5  source("analyze_approximation.r")
6  source("modify_parabola.r")
7
8  source("exam_generator.r")
9  source("exam_analyzer.r")
10 source("exam_draw.r")
11
12 options(warn=1)
13
14 get.group <- function (sample, group.number) {
15     sample$values[sample$groups == group.number,]
16 }
17
18 groups.names <- c("Не класиф.", "Зміш.", "Слабкий",
19                  "Неврів.", "Рухливий", "Інерт.")
20
21 get.chart <- function(sample, i) {
22     get.chart.g(sample, i, (seq(length(groups.names))-2))
23 }
24
25 get.amount <- function(sample, pc, group.number, f) {
26     fpc <- get.group(sample, group.number) %*% students.pca$rotation[,pc]
27     fpc[f(fpc)]
28 }
29
30 exam <- exam.generate(600)
31
32 students.pca <- prcomp(exam$values, center = TRUE)
33
34 png('pca.png', width=10, height=8, units="cm", res=300)
35 print(get.gg.students.pca(students.pca, 5))
36 dev.off()
37
38 png('pca_hists.png', width=16, height=12, units="cm", res=300)
39 print(multiplot(get.gg.students.pc(exam, students.pca$rotation, 1),
40                get.gg.students.pc(exam, students.pca$rotation, 2),
41                get.gg.students.pc(exam, students.pca$rotation, 3),
42                get.gg.students.pc(exam, students.pca$rotation, 4),
43                cols=2))
44 dev.off()
45

```


[illegible]

95

96 test.thinkers.prediction.quality <- rep(NA, test.exams.number)

97 test.learners.prediction.quality <- rep(NA, test.exams.number)

98 for (i in 1:test.exams.number) {

99 prediction.quality <- get.prediction.quality(get.fit(

100 get.students(

101 test.exams[[i]], students.pca\$rotation[,1])),

102 test.exams[[i]])

103 test.thinkers.prediction.quality[i] <- round(prediction.quality\$think*100)

104 test.learners.prediction.quality[i] <- round(prediction.quality\$learn*100)

105 }

106 test.thinkers.prediction.quality <- paste(sprintf("%d\\%%", test.thinkers.prediction.quality), collapse=", ")

107 test.thinkers.prediction.quality <- paste("\\def\\testThinkersPredictionQuality{", test.thinkers.prediction.q

108 test.learners.prediction.quality <- paste(sprintf("%d\\%%", test.learners.prediction.quality), collapse=", ")

109 test.learners.prediction.quality <- paste("\\def\\testLearnersPredictionQuality{", test.learners.prediction.q

110

111 writeLines(c(first.component.proportion,

112 thinkers.prediction.quality, learners.prediction.quality,

113 test.thinkers.prediction.quality,

114 test.learners.prediction.quality), file.conn)

115 close(file.conn)