КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

Кваліфікаційна робота

на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:

ІНТЕЛЕКТУАЛЬНІ МЕТОДИ АНАЛІЗУ ЕЛЕКТРОКАРДІОГРАМ

Виконав: студент 4-го курсу
Антон БУДНИК _____
(підпис)

Науковий керівник:
професор кафедри теоретичної кібернетики
доктор фіз.-мат. наук, професор
Анатолій ПАШКО _____
(підпис)

Засвідчую, що в цій роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент _____
(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теоретичної
кібернетики
«____» _____ 2023 р.,
протокол № ___
Завідувач кафедри
доктор фіз.-мат. наук, професор
Юрій КРАК

_____
(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи складає 45 сторінок, включаючи 42 ілюстрації та 14 джерел посилань.

Об’єктом роботи є процес обробки та аналізу даних електрокардіограм (ЕКГ) з використанням інтелектуальних методів. Предметом роботи є відповідні інтелектуальні методи, що використовуються для обробки та аналізу ЕКГ, зокрема: статистичний аналіз, дисперсійний аналіз, перетворення Фур’є, кореляційний та факторний аналіз, а також кластеризація даних.

Метою роботи є проведення аналізу дослідження можливостей та ефективності застосування інтелектуальних методів та алгоритмів для обробки та аналізу даних електрокардіограм.

В ході роботи було проведено загальний огляд інтелектуальних методів обробки та аналізу даних електрокардіограм та продемонстровано результати використання відповідних методів та алгоритмів, які дозволяють ефективно вирішувати задачі обробки, аналізу та класифікації ЕКГ-сигналів.

Результатом роботи став програмний продукт, призначений для обробки та аналізу даних, отриманих з електрокардіограм. Цей продукт може спростити роботу лікарів у медичних дослідженнях, діагностиці та моніторингу серцево-судинних захворювань. Використання інтелектуальних методів дозволяє покращити точність діагностики, спростити аналіз даних та забезпечити швидку обробку великого обсягу інформації.

Отримані результати свідчать про високий потенціал інтелектуальних методів у цій області та можуть послужити основою для подальшого розвитку та удосконалення методів обробки та аналізу електрокардіограм.

ЗМІСТ

ВСТУП

Оцінка сучасного стану об’єкта дослідження. Інтелектуальний аналіз даних поєднує широкий спектр математичних інструментів, від класичного статистичного аналізу до новітніх кібернетичних методів, а також останні досягнення в галузі інформаційних технологій [1]. Технології інтелектуального аналізу здатні гармонійно поєднувати строго формалізовані методи з методами неформального аналізу, які включають якісний та кількісний аналіз даних. Математичні методи обробки інформації та аналізу даних широко застосовуються для дослідження різноманітних систем і процесів.

Актуальність роботи та підстави для її виконання. З урахуванням того, що серцево-судинні захворювання посідають перше місце в світі, зокрема, в Україні серед причин смерті, виникла необхідність використання інтелектуальних методів обробки та аналізу даних ЕКГ, які дозволяють покращити точність діагностики, спростити аналіз даних та забезпечити швидку обробку інформації для діагностики та моніторингу захворювань.

Мета й завдання роботи. Метою роботи є проведення аналізу дослідження можливостей та ефективності застосування інтелектуальних методів та алгоритмів для обробки та аналізу даних електрокардіограм. Для досягнення цієї мети поставлені такі завдання:

- провести попередню обробку даних та їх візуалізацію;
- здійснити однофакторний та двофакторний дисперсійний аналіз;
- виконати пряме та обернене перетворення Фур’є;
- здійснити кореляційний аналіз та факторний аналіз;
- виконати кластерний аналіз даних.

РОЗДІЛ 1 МЕТОДИ ОБРОБКИ ТА АНАЛІЗУ ДАНИХ

1.1 Статистичний аналіз

Математична статистика передбачає всебічний аналіз існуючої сукупності даних, на основі якого визначаються тип і параметри розподілу, якому підпорядковуються дані, і перевіряються гіпотези щодо параметрів цього розподілу. У прикладній статистиці вивчаються зв’язки між різними змінними, одні з них представляють як функції інших, і на цій основі робляться прогнози. Наприклад, як можуть змінюватися деякі параметри при певних змінах інших параметрів або розвитку подій у майбутньому.

На практиці найчастіше маємо справу з вибірковими характеристиками, які розраховуються на основі обмеженої кількості значень досліджуваного показника, що складають певну вибірку з генеральної сукупності. Це оцінки відповідних загальних статистичних характеристик (параметрів розподілу, які не є випадковими величинами). На відміну від характеристик вибірки, які є випадковими величинами, що змінюються від вибірки до проби.

Вибіркові характеристики: точкові оцінки та інтервальні оцінки параметрів. Точкові оцінки параметрів вибірки визначаються одним числом. Прикладами таких оцінок є середнє арифметичне та медіана вибірок. Інтервальні оцінки визначаються межами інтервалу – двома числами, між якими з заданою ймовірністю потрапляє оцінюваний параметр.

При малих обсягах вибірок, а також при їх значному відхиленні від нормального закону розподілу точкові оцінки можуть істотно відхилятися від справжніх значень оцінюваних параметрів. Тому поряд з точковими оцінками можна використовувати інтервальні оцінки параметрів [2].

Одним з основних завдань описової статистики є визначення центру, ширини, симетрії та довжини розподілу. Центр статистичного розподілу характеризується його математичним сподіванням, середнім, медіаною та модою. Як показники ширини розподілу найчастіше використовують

дисперсію і стандартне відхилення вибірки [2]. Крім того, використовуються середні відхилення, середня різниця Джині та інші оцінки.

Статистичний аналіз відіграє важливу роль в інтелектуальній обробці та аналізу даних. Його основна роль полягає у визначенні структури, закономірностей та залежностей у наборі даних для отримання корисної інформації та прийняття обґрунтованих рішень.

Основними функціями статистичного аналізу є:

1) Описовий аналіз: статистичні методи дозволяють нам описати набір даних за допомогою різних показників, таких як середнє значення, медіана, варіація тощо. Це допомагає зрозуміти основні характеристики даних та їхню структуру.

2) Виявлення закономірностей: Статистичний аналіз дозволяє виявляти статистичні закономірності та залежності між змінними. Наприклад, кореляційний аналіз можна використовувати, щоб визначити, чи існує статистичний зв’язок між двома змінними.

3) Перевірка гіпотез: статистичні методи дозволяють нам перевірити гіпотези щодо параметрів розподілу даних і встановити статистичну значущість результатів. Це дозволяє робити об’єктивні висновки на основі наявних даних.

Існує велика кількість різних статистичних методів перевірки гіпотез. При виборі методу вирішення конкретного завдання необхідно виходити з відповідей на наступні питання:

- мета перевірки гіпотези;
- шкали та одиниці вимірювання аналізованих даних;
- чи є незалежними або спряженими досліджувані сукупності;
- кількість даних, що необхідно порівняти.

Зазвичай ці методи використовують для порівняння двох вибірок. При більшій кількості застосовують методи дисперсійного аналізу.

Критерії та тести, що використовуються для порівняння зразків, поділяються на дві групи: параметричні та непараметричні. Особливістю параметричних критеріїв є припущення про те, що розподіл ознак у генеральній сукупності підпорядковується певному відомому закону. Дана відповідність повинна бути доведена перед застосуванням будь-якого параметричного тесту. Переважна більшість параметричних тестів розроблена для нормально розподілених даних. Однак, для деяких типів гіпотез існують параметричні тести, призначені для вибірок, що підлягають іншим законам розподілу [2].

Параметричні критерії є більш потужними, ніж непараметричні. Використання непараметричних критеріїв у тих випадках, коли можна використовувати параметричні критерії, призводить до збільшення ймовірності прийняття хибної нульової гіпотези.

1.2 Дисперсійний аналіз

Дисперсійний аналіз – це сукупність статистичних методів, які використовуються для перевірки гіпотез про зв’язок між певною ознакою та досліджуваними факторами, які не мають кількісного опису. Це також дозволяє встановити ступінь впливу факторів та їх взаємодії [3].

Факторами є контрольовані чинники, які впливають на кінцевий результат. Рівні факторів або способи обробки представлені значеннями, що характеризують конкретний прояв цих факторів. Ці значення зазвичай представлені в номінальній або порядковій шкалі вимірювання. Характеристики ознак, які вимірюються, називаються відгуками.

У дисперсійному аналізі використовуються різні типи групування залежно від кількості та розміру інтервалів:

- групи з однаковою кількістю спостережень;
- групи з різною кількістю спостережень;
- групи з встановленою пропорцією кількості спостережень.

Також існують певні свої особливості обробки даних в залежності від типу групування [4].

Однофакторний дисперсійний аналіз використовується для оцінки впливу певного фактора на відгук. Його також можна використовувати для порівняння різних факторів, щоб визначити відмінності в їх впливі, що називається контрастом факторів. Попереднім етапом є перевірка нульової гіпотези про відсутність впливу досліджуваного фактора (факторів).

При однофакторному дисперсійному аналізі вихідні дані подаються у вигляді таблиць, де кількість стовпчиків дорівнює кількості рівнів фактора, а кількість значень у кожному стовпчику відповідає кількості спостережень на відповідний рівень фактора [5]. Метою аналізу є перевірка нульової гіпотези про те, що середні значення сукупностей, які розглядаються, рівні.

Двофакторний дисперсійний аналіз використовується для нормально розподілених пов’язаних вибірок. Дані представлені у вигляді таблиці, де стовпчики відповідають рівням першого фактору, а рядки – рівням другого фактору. Розмірність таблиці даних становить n×k, де n і k – номери рівнів першого та другого факторів відповідно.

Дисперсійний аналіз відіграє важливу роль у інтелектуальних методах обробки даних. З його допомогою відбувається перевірка гіпотези про взаємозв’язок характеристик ознак і досліджуваних факторів, щоб встановити ступінь їх впливу та взаємодії. Цей аналітичний інструмент допомагає виявити статистично значущі зв’язки між змінними та встановити достовірність отриманих результатів.

У контексті інтелектуальних методів обробки та аналізу даних дисперсійний аналіз можна використовувати для виявлення важливих факторів або особливостей, які впливають на конкретний результат або відгук. Це дозволяє встановити, які з факторів є статистично значущими та суттєво впливають на досліджувані ознаки. Крім того, дисперсійний аналіз дозволяє оцінити взаємодію між різними факторами та ідентифікує взаємодії, які можуть бути значущими для розуміння досліджуваної проблеми.

1.3 Кореляційний аналіз та факторний аналіз

Кореляція (або кореляційний зв’язок) між випадковими характеристиками величин (ознаками) вказує на наявність статистичного або ймовірнісного зв’язку між ними [6]. Відповідно до цього зв’язку, зміна одних ознак веде до закономірної зміни середніх значень інших пов’язаних ознак.

Кореляційний аналіз складається з набору методів, які дозволяють виявити кореляційний зв’язок між ознаками. Через це даний аналіз можна використовувати для формалізації моделей зв’язків між компонентами системи або між процесами, які відбуваються в ній.

Важливо розуміти, що наявність кореляційного зв’язку не завжди означає, що існує причинно-наслідковий зв’язок між ознаками. Такий зв’язок може бути обумовлений тим, що обидві ознаки мають причинно-наслідковий зв’язок з іншим фактором.

Кореляційний аналіз є важливим етапом у вирішенні різних проблем статистичного аналізу даних. У випадку аналізу залежностей та побудови регресійних моделей, кореляційний аналіз дозволяє встановити наявність зв’язку між змінними та оцінити його ступінь [7]. При класифікації даних, кореляційний аналіз надає вихідну інформацію у вигляді коваріаційних і кореляційних матриць та інших характеристик для порівнянь між парами ознак. Це дозволяє визначити подібні об’єкти, сформувати класи схожих об’єктів та провести класифікацію. У випадку зменшення розмірності простору ознак, кореляційний аналіз допомагає ідентифікувати ознаки, які можуть бути представлені через інші наявні дані без втрати суттєвої інформації за допомогою кореляційних матриць.

Алгоритми факторного аналізу дозволяють зменшити розмірність набору даних, що описують досліджувані об’єкти. Замість безпосереднього вимірювання великої кількості ознак, факторний аналіз дозволяє представити ці ознаки меншим числом змінних, які називаються факторами. Фактори є

функціями початкових ознак і зазвичай вони відображають скриті властивості об’єктів, які не можна безпосередньо виміряти [8].

Методи факторного аналізу допомагають агрегувати дані та виявляти загальні закономірності. Взаємозв’язок між факторами та початковими ознаками представляється у вигляді факторної матриці або матриці факторних навантажень. Ця матриця має розмірність m×r, де m – кількість початкових ознак, а r – кількість факторів. Факторна матриця показує ступінь зв’язку між кожною з ознак і кожним з факторів, а для її побудови використовують кореляційну матрицю початкових ознак.

Вибір кількості факторів (r) зазвичай залежить від мети аналізу, і його значення обирається таким чином, щоб зберегти якомога більше інформації при зменшенні кількості початкових факторів (m). Факторна матриця дозволяє виділити групи ознак, які найбільше пов’язані з кожним фактором, що дозволяє змістовно тлумачити і називати фактори.

Кореляційний аналіз та факторний аналіз відіграють важливу роль в інтелектуальних методах обробки та аналізу даних. Основні функції цих аналітичних методів включають:

1) Виявлення зв’язків: Кореляційний аналіз дозволяє виявити наявність статистичних або ймовірнісних зв’язків між випадковими величинами або ознаками. Факторний аналіз виявляє внутрішні залежності та властивості даних шляхом знаходження факторів, які пояснюють багатовимірну структуру даних.

2) Зменшення розмірності даних: Факторний аналіз допомагає зменшити розмірність набору даних, замінюючи його меншим числом факторів. Це дозволяє зберегти суттєву частину інформації, що має значення при обробці та аналізі великих обсягів даних.

3) Побудова моделей: Кореляційний аналіз і факторний аналіз допомагають в побудові моделей зв’язків між ознаками та факторами. Це може бути корисно для прогнозування, класифікації або зрозуміння структури даних.

4) Виділення груп: Факторний аналіз може допомогти виділити групи ознак, які мають схожі характеристики та сильний зв’язок з одним або кількома факторами. Це дозволяє проводити групування об’єктів за схожістю та знаходити спільні властивості в масиві даних.

5) Тлумачення результатів: Кореляційний аналіз та факторний аналіз надають змогу тлумачити результати аналізу шляхом ідентифікації факторів та їх зв’язку з ознаками. Це дозволяє розуміти сутність даних і надає підґрунтя для прийняття рішень та подальшого дослідження.

Таким чином, кореляційний аналіз та факторний аналіз є потужними інструментами для розуміння, аналізу та інтерпретації даних у контексті інтелектуальної обробки та аналізу даних. Вони часто використовуються в поєднанні з іншими статистичними методами, такими як кореляційно-регресійний аналіз та кластерний аналіз.

1.4 Кластерний аналіз

Класифікація може використовуватись для розпізнавання образів, прогнозування результатів, прийняття рішень та багатьох інших завдань аналізу даних. У загальному випадку вона включає поділ досліджуваної сукупності об’єктів на групи або класи відповідно до їхніх спільних характеристик та властивостей. Мета класифікації може варіюватися від формування однорідних груп до виявлення природного розшарування в досліджуваній сукупності.

Для класифікації даних можуть бути використані методи параметричного і непараметричного дискримінантного аналізу, якщо наявні навчальні вибірки. В іншому випадку, коли навчальні дані відсутні, застосовуються методи кластерного аналізу, таксономії та статистичних гіпотез. Кластерний аналіз дозволяє вирішувати завдання класифікації, виявлення структури та побудови нових класифікацій для слабо вивчених явищ.

Методи кластерного аналізу включають ієрархічні агломеративні методи, ієрархічні дивізимні методи, ітеративні методи групування та інші [9]. Агломеративні методи об’єднують кластери в нові, тоді як дивізимні методи розчленовують кластери на окремі групи. Ітеративні методи вимагають задання початкових умов та часто використовують результати агломеративних методів в якості початкових кластерів.

Для формування кластерів використовуються міри подібності та відмінності даних. Ці міри можуть бути виражені у вигляді відстаней, зв’язків або інформаційної статистики. Вибір підходящої міри залежить від мети дослідження, природи даних і відомостей про їх розподіл.

Після отримання результатів класифікації важливо перевірити адекватність отриманої класифікаційної моделі, враховуючи початкові умови та властивості даних. Надійні результати кластерного аналізу можуть допомогти розкрити глибше розуміння досліджуваних об’єктів та їх взаємозв’язків.

Один з популярних методів класифікації даних є метод k-середніх (англ. k-means). Цей метод базується на принципі кластеризації, де об’єкти групуються в кластери відповідно до їхньої подібності. Алгоритм цього методу складається з таких кроків:

1) Вибір кількості кластерів (k), що є попередньо заданою величиною, яка визначається при дослідженні.
2) Випадковий вибір початкових центроїдів для кожного кластера.
3) Призначення кожного об’єкта до найближчого центроїда на основі обраних мір подібності.
4) Перерахунок центроїдів на основі призначених об’єктів.
5) Повторення кроків 3 і 4 до збіжності, коли кластери стабілізуються або до досягнення заданої умови зупинки.

Загалом, класифікація даних та кластерний аналіз є важливими інструментами для організації та розуміння великих обсягів даних. Вони дозволяють групувати об’єкти за спільними характеристиками, знаходити структуру та зв’язки в даних. Метод k-середніх є одним із широко використовуваних методів кластерного аналізу, який дозволяє ефективно розбити дані на кластери ітеративним шляхом. Всі ці методи є важливим для інтелектуальної обробки та аналізу даних різних галузей, включаючи медицину, біологію, соціальні науки та бізнес-аналітику.

РОЗДІЛ 2 ЗАСОБИ РЕАЛІЗАЦІЇ

2.1 Мова програмування R та середовище розробки RStudio

Мова програмування R є потужним і широко використовуваним інструментом для статистичного аналізу та обробки даних. Розроблена у 1990-х роках, вона зарекомендувала себе як важливий інструмент у галузі статистики та аналізу даних. Завдяки своїм функціональним можливостям і широкому спектру пакетів для статистичного моделювання та візуалізації даних, вона набула значного визнання серед дослідників та статистиків.

R є інтерпретованою мовою з векторно-орієнтованою семантикою, що спрощує обробку та аналіз даних [10]. Код на R легко читати та розуміти, що робить його доступним навіть для початківців. Крім того, R також підтримує інтеграцію з іншими мовами програмування, що дає можливість розширити його функціональність та використовувати його у складних проектах.

Одна з основних переваг мови R полягає у її розширюваності. R має велику кількість пакетів, які надають додаткові функції та можливості для різних аспектів статистичного моделювання, обробки та аналізу даних, а також багато іншого. Це дозволяє користувачам з легкістю виконувати складні аналітичні завдання та розробляти власні програми.

R також привертає увагу своєю зручністю в роботі з даними. Вона має багато вбудованих функцій для маніпулювання даними, обробки пропущених значень, фільтрації, сортування та агрегування даних. Інтерфейс R дозволяє використовувати графічні засоби для візуалізації даних, що сприяє зрозумінню та аналізу результатів.

Оскільки R є відкритим програмним забезпеченням, що робить його доступним для широкої аудиторії користувачів. Ця відкритість також сприяє активному розвитку та підтримці мови програмування R. Завдяки активній спільноті користувачів та розробників, мова постійно оновлюється та

вдосконалюється. Все це робить R однією з найпопулярніших мов програмування у сфері статистики та аналізу даних.

RStudio – це інтегроване середовище розробки для мови програмування R. Воно створене з урахуванням потреб аналітиків даних, статистиків та програмістів, що працюють з R. RStudio надає зручні інструменти для роботи з R, полегшуючи процес розробки, налагодження і виконання аналітичних проектів.

Однією з ключових особливостей RStudio є його інтуїтивно зрозумілий та організований інтерфейс. Він складається з різних панелей, які дозволяють легко переглядати та редагувати код, спостерігати за результатами, візуалізувати дані та керувати проектами. Це сприяє підвищенню продуктивності розробника і полегшує навігацію у складних проектах.

RStudio також надає широкий набір інструментів для підтримки розробки на R. Включаючи автодоповнення коду, перевірку синтаксису, систему контролю версій, інтерактивну довідку та інструменти для документування коду. Це допомагає знизити кількість помилок, покращує якість коду і спрощує роботу з командами та функціями R.

Перевагою RStudio є можливість використання його на різних платформах, включаючи Windows, macOS і Linux, що робить програмування на R універсальним і доступним для широкого кола користувачів. Крім того, RStudio підтримує інтеграцію з іншими популярними інструментами, такими як LaTeX і RMarkdown [11]. Це дозволяє легко комбінувати розробку коду з документацією, створювати звіти та публікувати результати аналізу даних.

Загалом, RStudio є потужним та ефективним середовищем розробки для мови програмування R, яке допомагає статистикам та аналітикам даних максимально використовувати можливості R для аналізу, візуалізації та моделювання даних.

2.2 Мова програмування Python та середовище розробки PyCharm

Python – це високорівнева інтерпретована мова програмування загального призначення. Однією з особливостей Python є його філософія читабельності коду, що досягається за рахунок використання відступів. Це робить програмний код Python легким для розуміння та обслуговування, що сприяє написанню чітких та логічних програм для різноманітних проектів.

Python підтримує різні парадигми програмування, зокрема об’єктно-орієнтовану, структурну, функціональну та аспектно-орієнтовану [12]. Ця гнучкість дозволяє розробникам вибирати підхід, що найкраще відповідає їхнім потребам. Деякі інші парадигми підтримуються за допомогою розширень.

Динамічне введення тексту, комбінація підрахунку посилань і збирач сміття дозволяють ефективно керувати пам’яттю в Python. Також мова програмування має властивість динамічної роздільної здатності імен, що дає змогу пов’язувати між собою імена змінних та методів під час виконання програми.

У Python є багато різних модулів та бібліотек, що розширюють його функціональність. Наприклад, для машинного навчання та штучного інтелекту доступні такі бібліотеки, як TensorFlow, Keras, Pytorch, Scikit-learn та інші. Python також популярний у сфері обробки природної мови і має відповідні інструменти для цього, завдяки простому синтаксису та великої кількості модулів. Окрім цього, Python застосовується в інформаційній безпеці, де він використовується для розробки експлойтів. Python має засоби для роботи з базами даних, мережами, графікою, обробки зображень та багато іншого.

Python також відомий своєю переносимістю, що означає, що програми, написані цією мовою, можуть працювати на різних операційних системах без змін. Це робить Python ідеальним вибором для розробки кросплатформових додатків.

Python є однією з найпопулярніших мов програмування завдяки своїм перевагам у розробці, широкому спектру застосувань та активній спільноті розробників. Python є потужним інструментом, зважаючи на простоту використання, читабельність коду і багатий набір бібліотек, дана мова програмування продовжує здобувати популярність для розробки різноманітних проектів.

PyCharm – це потужне інтегроване середовище розробки для мови програмування Python, розроблене чеською компанією JetBrains [13]. PyCharm надає широкі можливості для розробки проектів на Python. Підтримка різних версій мови Python, починаючи з Python 2 та включно з останніми версіями Python 3, а також має багато функцій, які полегшують розробку, налагодження та тестування програм. Інтерфейс PyCharm є інтуїтивно зрозумілим і дружнім до користувача, що дозволяє зосередитися на роботі без зайвих перешкод.

Однією з ключових особливостей PyCharm є його підтримка автоматичного завершення коду та інтелектуального аналізу. Під час введення коду середовище розробки пропонує підказки, які допомагають знайти помилки, а ще пропонує оптимальні шляхи вирішення проблем. Також середовище має вбудовані інструменти для перевірки стилю коду та контролю якості.

PyCharm надає зручні можливості для налагодження коду за допомогою вбудованого відлагоджувача, що дозволяє крок за кроком виконувати код, спостерігати зміни та виявляти помилки. Крім того, PyCharm підтримує використання віртуальних середовищ, що дозволяє ізолювати проекти та їх залежності, забезпечуючи чисте середовище розробки.

PyCharm також має інтегровану систему керування версіями, що дозволяє працювати з репозиторіями Git, Mercurial, Subversion та іншими. Це полегшує спільну роботу над проектами та відстеження змін в коді.

Загалом, PyCharm є одиним з найпопулярніших інструментів серед розробників Python, оскільки середовище розробки надає зручний інтерфейс, багатофункціональність та підтримку великої кількості інструментів, що сприяє покращенню продуктивності та якості розробки програмного забезпечення.

РОЗДІЛ 3 ОБРОБКА ТА АНАЛІЗ ЕЛЕТРОКАРДІОГРАМ

3.1 Реалізація мовою програмування R

Завдання. У файлі міститься запис кардіограми людини по 12 каналах. Час запису – 10 секунд (див. рис. 1). Дискретність: 500 точок за 1 секунду. Структура файлу: 1-й канал, 2-й канал, … 12-й канал (амплітуда у відносних одиницях). Довжина запису N = 5000, Δt = 1/500 = 0.002 [14].

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
1	0.709790	16.15700	16.27800	-8.46040	-7.799800	16.19600	-8.429900	4.67820	8.230700	38.68600	114.69000	91.21100
2	1.025700	15.68400	15.47500	-8.38260	-7.245600	15.55900	-7.709700	3.00320	7.423400	39.77900	113.46000	90.81800
3	1.294500	15.33000	14.83700	-8.34060	-6.797600	15.06500	-6.658200	1.05110	6.229900	40.22200	112.02000	89.73300
4	1.424300	15.24700	14.61000	-8.36450	-6.624000	14.91200	-5.111200	-1.29150	4.409300	39.53600	110.23000	87.49000
5	1.269900	15.61300	15.11600	-8.47090	-6.959400	15.35000	-3.109800	-3.89350	1.939200	37.54000	108.12000	83.95100
6	0.649390	16.57300	16.68200	-8.64120	-8.057600	16.61400	-0.827540	-6.44780	-1.006600	34.39400	105.75000	79.35500

Рисунок 1. Частина даних файлу А3.txt

Візуалізація даних. Побудовано графік кардіограми по кожному каналу. На рисунку 2 зображено перший з дванадцяти каналів електрокардіограми.

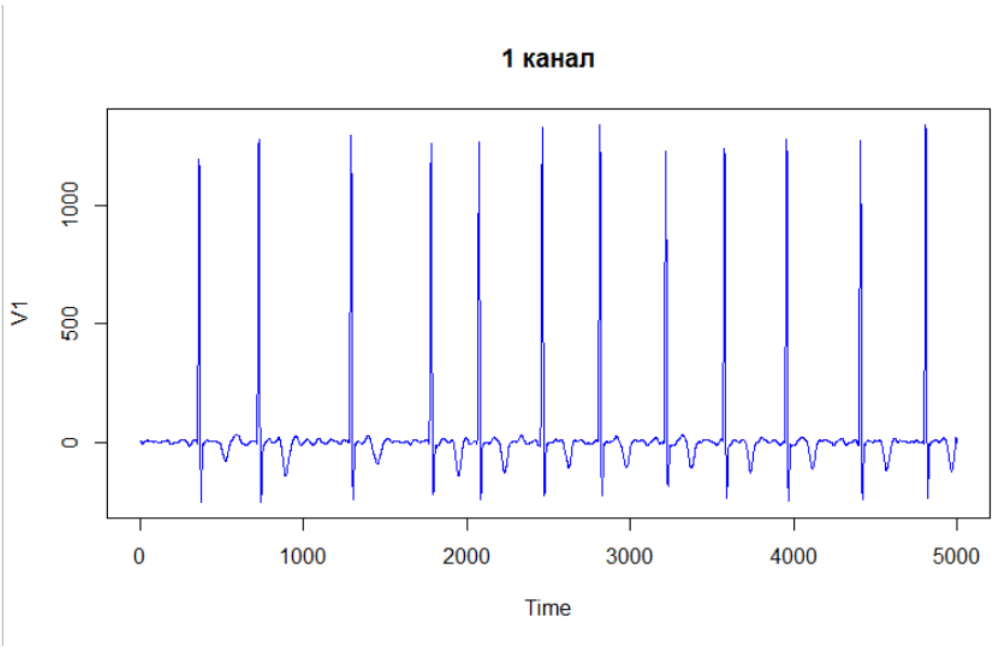

Рисунок 2. Перший канал запису електрокардіограми

Попередня обробка. Для заданих змінних знайдено та оцінено основні статистичні параметри (див. рис. 3): середнє арифметичне, середнє гармонічне, середнє геометричне, дисперсію, середню різницю Джині, моду, медіану, коефіцієнт асиметрії, коефіцієнт ексцесу та перевірено гіпотезу про нормальний закон розподілу.

```
[1] "Канал  1"
[1] "Середнє арифметичне:  17.31012638976"
[1] "Середнє гармонічне:  14.9953264213565"
[1] "Середнє геометричне:  7.29092157272097"
[1] "Дисперсія  30870.0781548314"
[1] "Середня різниця Джині:  92.8527842426486"
[1] "Мода:  -12.528"
[1] "Медіана:  0.230665"
[1] "Коефіцієнт асиметрії:  5.38202658653817"
[1] "Коефіцієнт ексцесу:  34.3226773446728"
[1] "Перевірка на нормальний розподіл: "

        Shapiro-Wilk normality test

data:  x
W = 0.33954, p-value < 2.2e-16
```

Рисунок 3. Основні статистичні параметри даних першого каналу

А також побудовано гістограми, перша з них зображена на рисунку 4.

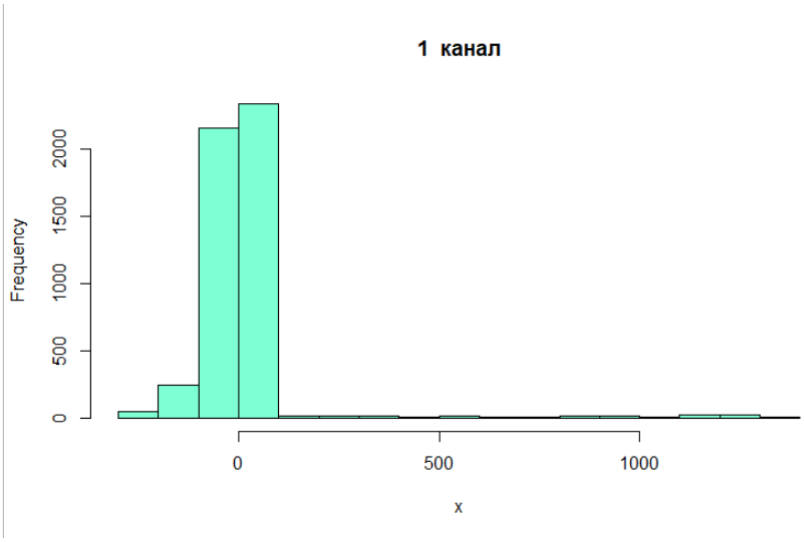

Рисунок 4. Гістограма даних першого каналу

Однофакторний дисперсійний аналіз. Проведена перевірка чи є результати вимірювання різними рівнями одного фактору (12 рівнів). Результати перевірки на рисунку 5.

```
[1] "Максимум:  27182162465.4133"
[1] "Сума:  65159826492.7458"
[1] "g:  0.417161369642988"
[1] "ga(k, n):  0.2411"
```

Рисунок 5. Перевірка даних електрокардіограми

Дисперсії не рівні, оскільки $g > g_a(k, n)$, тому результати вимірювання не є різними рівнями одного фактору.

Двофакторний дисперсійний аналіз. Побудовано таблицю двофакторного експерименту (див. рис. 6) за правилом – кожен канал розбито на 5 частин (по 1000 даних у кожній частині).

```
> print(mean1)
        V1        V2        V3        V4       V5        V6        V7        V8        V9       V10       V11
1 13.29921 -1.713908 -11.52937 -4.507044 14.44597 -6.683022 -1.494176 -4.067593 -12.16186 -34.00406 -19.1245
        V12
1 0.1807929
> print(mean2)
        V1        V2       V3        V4       V5        V6        V7        V8       V9       V10       V11
1 14.08644 -2.189479 -12.6455 -5.198143 15.81914 -7.303274 -5.236228 -8.982549 -14.1876 -30.51072 -26.77853
        V12
1 -1.459899
> print(mean3)
        V1        V2        V3        V4       V5        V6        V7       V8        V9       V10       V11
1 22.37127 -2.050125 -17.83165 -8.998848 23.97264 -10.31365 -3.217212 4.71441 -8.100563 -52.76618 -50.99244
        V12
1 -15.51351
> print(mean4)
        V1       V2        V3      V4       V5        V6        V7      V8       V9      V10      V11
1 25.10556 2.950505 -16.86127 -13.265 24.83566 -8.264848 -9.106881 20.4672 19.00226 19.83955 48.43946
       V12
1 57.34482
> print(mean5)
        V1        V2        V3       V4       V5        V6        V7       V8       V9       V10       V11
1 11.68815 -4.977906 -12.17367 -1.39343 16.94102 -7.858491 -4.157723 -21.2084 -35.2306 -67.44321 -49.12623
        V12
1 -14.02725
```

Рисунок 6. Таблиця двофакторного експерименту

Знаходимо суму по рядках та стовпчиках. Результати обчислень на рисунку 7.

```
> print(columnsSum)
        V1         V2         V3         V4         V5         V6         V7         V8         V9
 86.550632  -7.980913 -71.041457 -33.362467  96.014425 -40.423281 -23.212220  -9.076932 -50.678358
       V10        V11        V12
-164.884618  -97.582253   26.524966
> print(sumMean1)
[1] -67.35955
> print(sumMean2)
[1] -84.58635
> print(sumMean3)
[1] -118.7259
> print(sumMean4)
[1] 170.487
> print(sumMean5)
[1] -188.9677
```

Рисунок 7. Сума по стовпчиках та рядках

Обчислюємо показники Q_1, Q_2, Q_3, Q_4 за формулами (див. рис. 8). Результати обчислень на рисунку 9.

$$Q_1 = \sum_{i=1}^{k}\sum_{j=1}^{m} x_{ij}^2; \quad Q_2 = \frac{1}{m}\sum_{i=1}^{k} X_i^2; \quad Q_3 = \frac{1}{k}\sum_{j=1}^{m} X_{j'}^2;$$

$$Q_4 = \frac{1}{mk}\left(\sum_{i=1}^{k} X_i\right)^2 = \frac{1}{mk}\left(\sum_{j=1}^{m} X_{j'}\right)^2.$$

Рисунок 8. Формули показників Q_1, Q_2, Q_3, Q_4

```
> print(Q1)
[1] 30439.46
> print(Q2)
[1] 6087.891
> print(Q3)
[1] 7546.884
> print(Q4)
[1] 1393.486
```

Рисунок 9. Значення показників Q_1, Q_2, Q_3, Q_4

Знаходимо оцінки дисперсій за формулами (див. рис. 10). Результати обчислень на рисунку 11.

$$S_0^2 = \frac{Q_1 + Q_4 - Q_2 - Q_3}{(k-1)(m-1)}; \quad S_A^2 = \frac{Q_2 - Q_4}{k-1}; \quad S_B^2 = \frac{Q_3 - Q_4}{m-1}.$$

Рисунок 10. Формули оцінки дисперсій

```
> S0_2 <- ((Q1 + Q4 - Q2 - Q3) / 44)
> print(S0_2)
[1] 413.5947
>
> SA_2 <- ((Q2 - Q4) / 11)
> print(SA_2)
[1] 426.7641
>
> SB_2 <- ((Q3 - Q4) / 4)
> print(SB_2)
[1] 1538.35
> print(SA_2 / S0_2)
[1] 1.031841
> print(SB_2 / S0_2)
[1] 3.719462
> df(0.05, 11, 44)
[1] 0.0003782347
> df(0.05, 4, 44)
[1] 0.1875275
```

Рисунок 11. Значення оцінок дисперсій

Оскільки $S_A^2 / S_0^2 > F(11, 44)$ і $S_B^2 / S_0^2 > F(4, 44)$, отже, фактори A та B є значущими.

Перетворення Фур'є. Виконано перетворення Фур'є та побудовано графіки для кожної змінної. На рисунку 12 зображено графік для першого каналу.

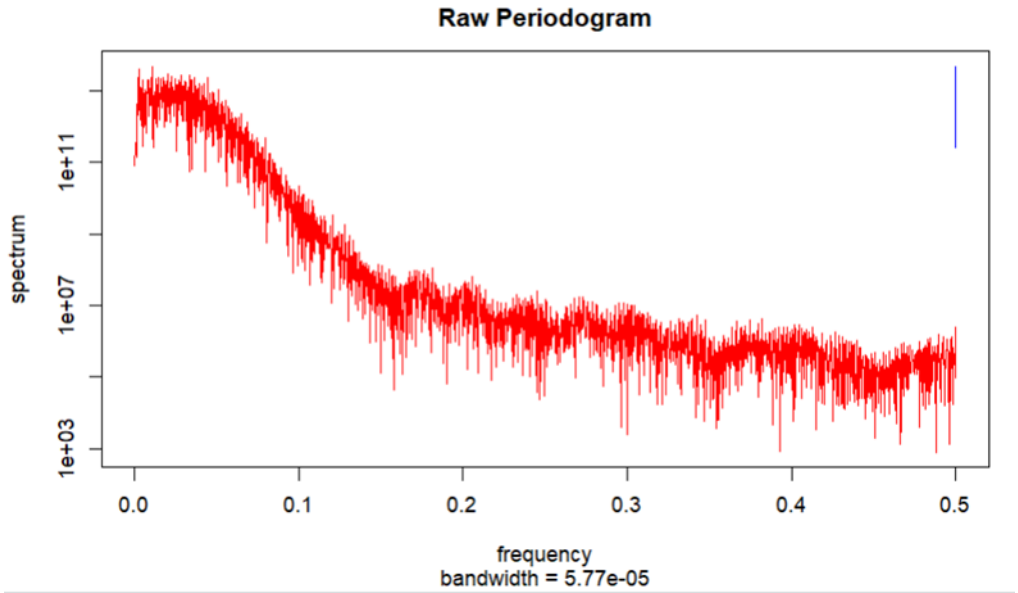

Рисунок 12. Графік перетворення Фур'є

Виконано обернене перетворення Фур'є та побудовано відповідні графіки змінних. На рисунку 13 зображено графік для першого каналу.

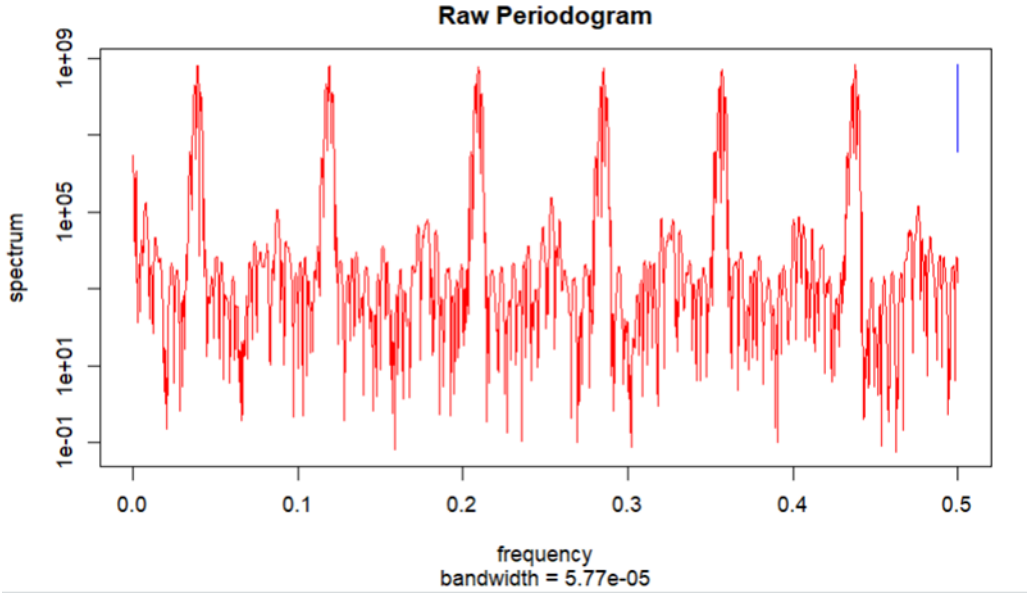

Рисунок 13. Графік оберненого перетворення Фур'є

Кореляційний аналіз. Кроки для проведення кореляційного аналізу:

Крок 1. Нормалізація всіх змінних та обчислення кореляційної матриці (див. рис. 14).

	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	1.00000000	0.6180194	-0.6153946	-0.9193088	0.9175085	0.00598229	-0.86494630	0.8961545	0.8835883
V2	0.61801936	1.0000000	0.2381844	-0.8773417	0.2554137	0.78929479	-0.48283479	0.8596780	0.8882761
V3	-0.61539461	0.2381844	1.0000000	0.2565490	-0.8772388	0.78378296	0.58175134	-0.2471002	-0.2031555
V4	-0.91930875	-0.8773417	0.2565490	1.0000000	-0.6875300	-0.39819183	0.76844140	-0.9776602	-0.9841914
V5	0.91750845	0.2554137	-0.8772388	-0.6875300	1.0000000	-0.39074168	-0.81919849	0.6662767	0.6372010
V6	0.00598229	0.7892948	0.7837830	-0.3981918	-0.3907417	1.00000000	0.05784645	0.3918510	0.4382570
V7	-0.86494630	-0.4828348	0.5817513	0.7684414	-0.8191985	0.05784645	1.00000000	-0.7140209	-0.7164700
V8	0.89615452	0.8596780	-0.2471002	-0.9776602	0.6662767	0.39185103	-0.71402094	1.0000000	0.9942845
V9	0.88358833	0.8882761	-0.2031555	-0.9841914	0.6372010	0.43825703	-0.71647003	0.9942845	1.0000000
V10	0.92005832	0.8266844	-0.3125686	-0.9756895	0.7144116	0.33042378	-0.75480211	0.9777927	0.9835839
V11	0.92236928	0.8238348	-0.3181346	-0.9754551	0.7193246	0.32538268	-0.75536131	0.9660388	0.9743838
V12	0.92714260	0.8327198	-0.3139319	-0.9825752	0.7209229	0.33389145	-0.76603412	0.9628160	0.9736940

	V10	V11	V12
V1	0.9200583	0.9223693	0.9271426
V2	0.8266844	0.8238348	0.8327198
V3	-0.3125686	-0.3181346	-0.3139319
V4	-0.9756895	-0.9754551	-0.9825752
V5	0.7144116	0.7193246	0.7209229
V6	0.3304238	0.3253827	0.3338914
V7	-0.7548021	-0.7553613	-0.7660341
V8	0.9777927	0.9660388	0.9628160
V9	0.9835839	0.9743838	0.9736940
V10	1.0000000	0.9957690	0.9901704
V11	0.9957690	1.0000000	0.9972090
V12	0.9901704	0.9972090	1.0000000

Рисунок 14. Кореляційна матриця

Крок 2. Аналіз кореляційної матриці: виділяємо групу параметрів, парна кореляція між якими велика (коефіцієнт кореляції близький по модулю до 1). Наприклад, параметри V1, V4, V11, V12 (див. рис. 14).

Крок 3. Знайдено часткові та множинні коефіцієнти кореляції. Розглянемо приклад для параметрів V1, V4, V11, V12 на рисунку 15.

```
> rab_c <- ((rab - rac * rbc) / sqrt((1 - rac) * (1 - rbc)))
> print(rab_c)
[1] -0.04999628
>
> rac_b <- ((rac - rab * rbc) / sqrt((1 - rab) * (1 - rbc)))
> print(rac_b)
[1] 0.01315997
>
> rab_cd <- ((rab - rac * rbc * rad * rbd) / sqrt((1 - rac) * (1 - rbc) * (1 - rad) * (1 - rbd)))
> print(rab_cd)
[1] -11.68383
>
> rac_bd <- ((rac - rab * rbc * rad * rcd) / sqrt((1 - rab) * (1 - rbc) * (1 - rad) * (1 - rcd)))
> print(rac_bd)
[1] 3.359408
>
> rad_bc <- ((rad - rab * rbd * rac * rcd) / sqrt((1 - rab) * (1 - rbd) * (1 - rac) * (1 - rcd)))
> print(rad_bc)
[1] 3.353862
>
> ra_bc <- sqrt((rab * rab + rac * rac - 2 * rab * rac * rbc) / (1 - rbc * rbc))
> print(ra_bc)
[1] 0.926645
>
> ra_bcd <- (1 - (1 - rab * rab) * (1 - rac_b * rac_b) * (1 - rad_bc * rad_bc))
> print(ra_bcd)
[1] 2.586907
```

Рисунок 15. Часткові та множинні коефіцієнти кореляції параметрів

Факторний аналіз. Кроки для проведення факторного аналізу:

Крок 1. Знаходження власних чисел кореляційної матриці. Побудова таблиці з власних чисел, частки дисперсії та сумарної дисперсії (див. рис. 16).

№п/п	Власні числа	Частка дисперсії	Сумарна дисперсія
1	0,00005802613	11,99994197387	99,99951644892
2	0,0001939835	11,9998060165	99,998383471
3	0,0002603562	11,9997396438	99,997830365
4	0,0005563688	11,9994436312	99,995363593
5	0,0008707985	11,9991292015	99,992743346
6	0,00233681	11,99766319	99,980526583
7	0,003560053	11,996439947	99,970332892
8	0,04381784	11,95618216	99,63485133
9	0,05517045	11,94482955	99,54024625
10	0,2559188	11,7440812	97,8673433
11	2,775137	9,224863	76,8738583
12	8,862089	8,862089	73,85074167

Рисунок 16. Таблиця власних чисел, частки дисперсії та сумарної дисперсії

Крок 2. Власні значення упорядковуються за абсолютним рівнем вкладу кожного головного компонента до загальної дисперсії, що відображено на графіку кам'янистого осипу (див. рис. 17). Обчислення критерію інформативності.

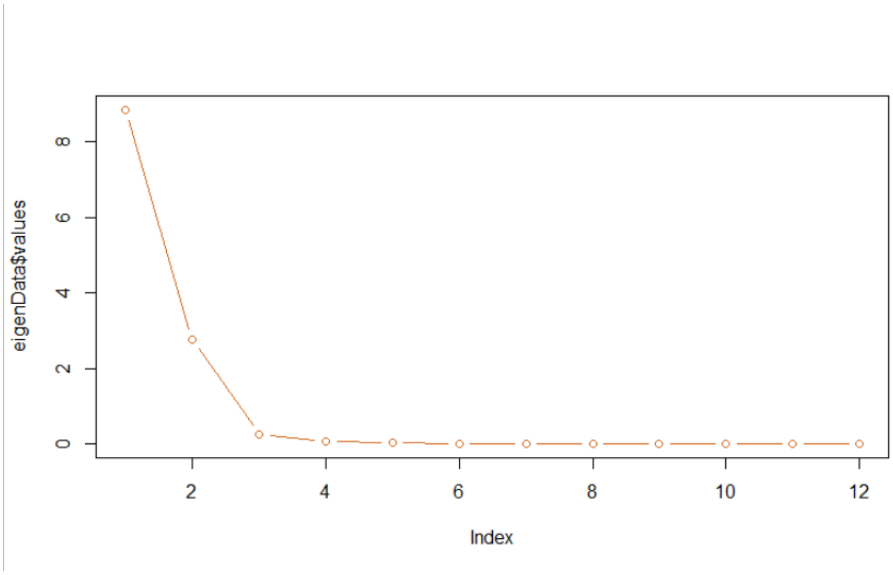

Рисунок 17. Графік кам'янистого осипу

Критерій інформативності по першим трьом власним числам приблизно дорівнює 0,97867.

Крок 3. До власного вектора максимального власного числа додаємо ще власні вектори другого і третього власного числа, разом дані вектори утворюють перші три головних фактори (див. рис. 18).

	[,1]	[,2]	[,3]
1	2.242883e-01	6.259673e-02	-1.952618e-01
2	2.218658e-01	6.045680e-02	-1.896223e-01
3	2.198723e-01	5.782138e-02	-1.842347e-01
4	2.190435e-01	5.455898e-02	-1.798642e-01
5	2.204837e-01	5.104997e-02	-1.779104e-01
6	2.254287e-01	4.823800e-02	-1.799601e-01
7	2.346623e-01	4.742792e-02	-1.868944e-01
8	2.478437e-01	4.990729e-02	-1.981208e-01
9	2.632237e-01	5.658855e-02	-2.114682e-01
10	2.781503e-01	6.774329e-02	-2.239665e-01
11	2.900384e-01	8.272709e-02	-2.330282e-01
12	2.030372e-01	8.764914e-02	-1.691685e-01

Рисунок 18. Перші три головних фактори

Побудова графіків основних компонент (див. рис. 19) та перевірка властивостей.

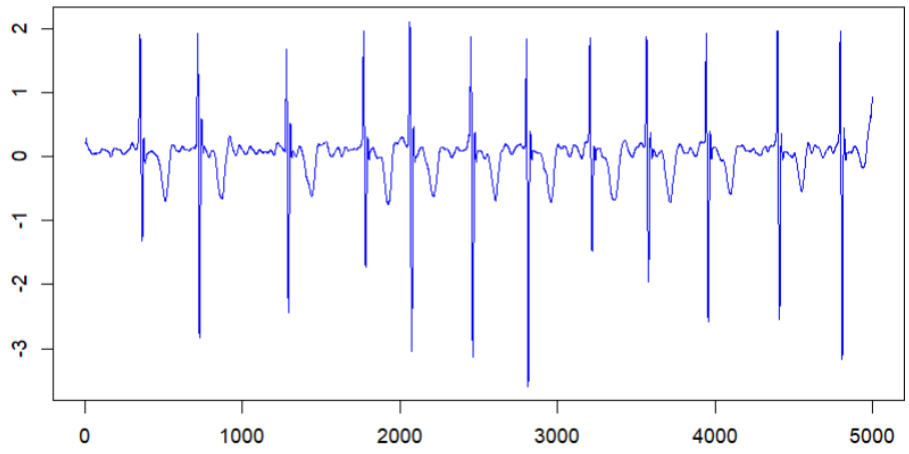

Рисунок 19. Графік першої головної компоненти

Кластерний аналіз. Будемо вважати, що записана кардіограма (12 каналів) являє собою множину багатовимірних точок деякого евклідового простору. Результати спостережень являють собою 5000 точок, кожна точка є вектором розмірності 12 (у випадку головних компонент – розмірності 3).

Використано алгоритм кластеризації, а саме метод k-середніх, проведено розбиття множини точок на k підмножин. Результатом є перелік точок, що входять до кожного кластеру.

Спочатку виконаємо кластерний аналіз для початкових даних для п’яти (див. рис. 20) та семи (див. рис. 21) кластерів.

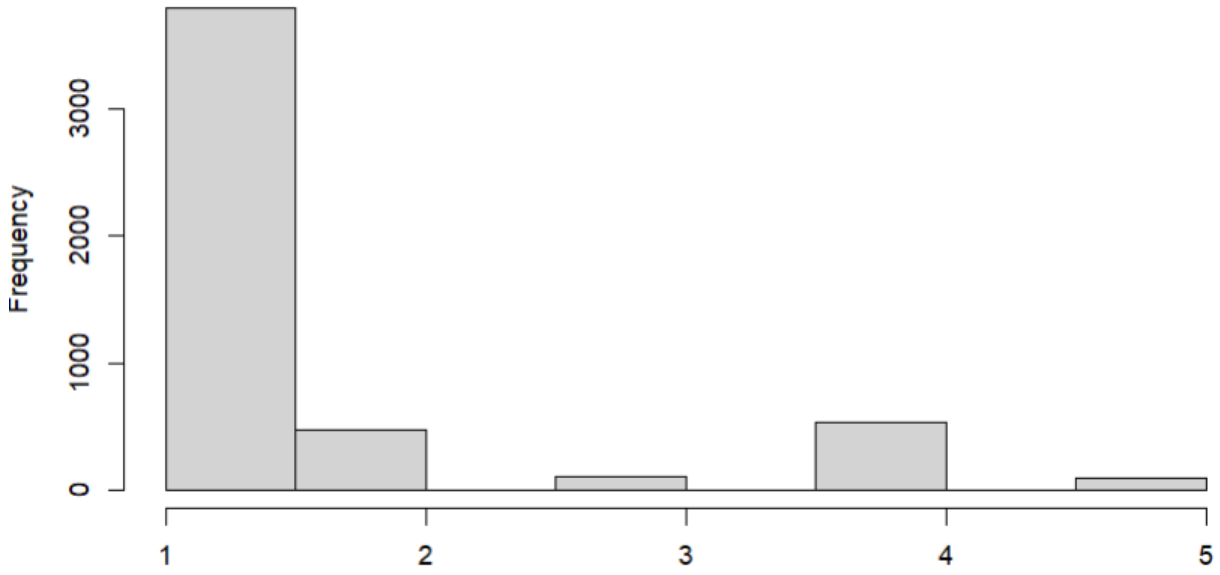

Рисунок 20. Гістограма кластерів початкових даних (k = 5)

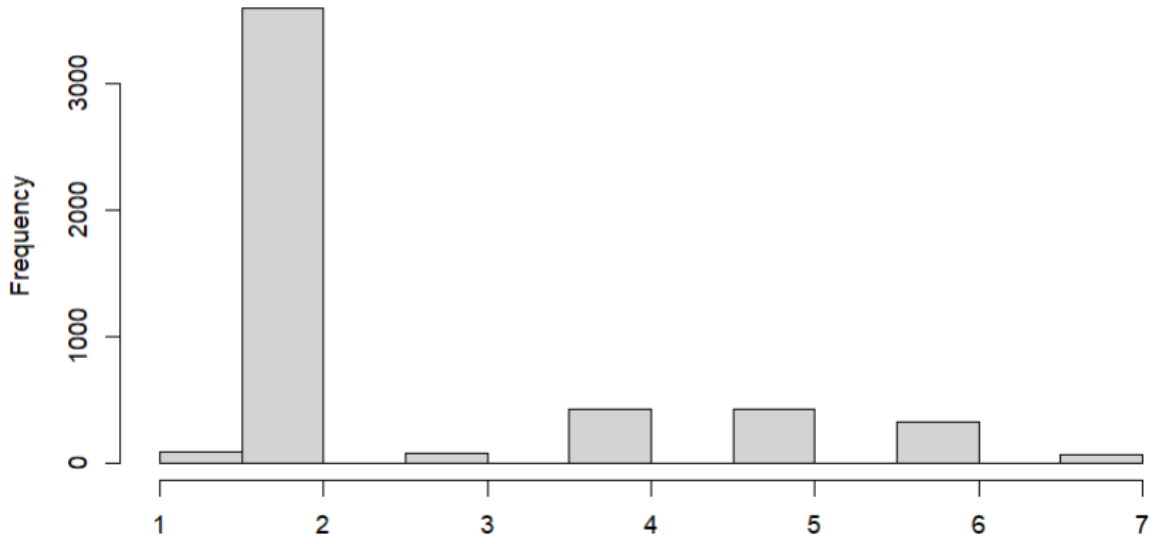

Рисунок 21. Гістограма кластерів початкових даних (k = 7)

Тепер виконаємо кластеризацію даних трьох головних факторів для п’яти (див. рис. 22) та семи кластерів (див. рис. 23).

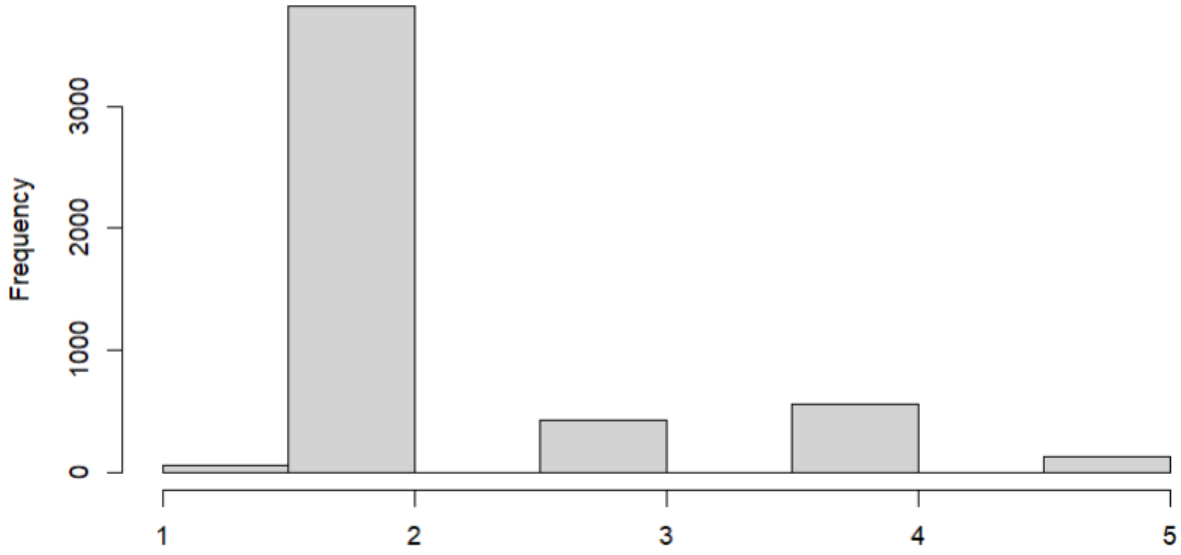

Рисунок 22. Гістограма кластерів даних факторного аналізу (k = 5)

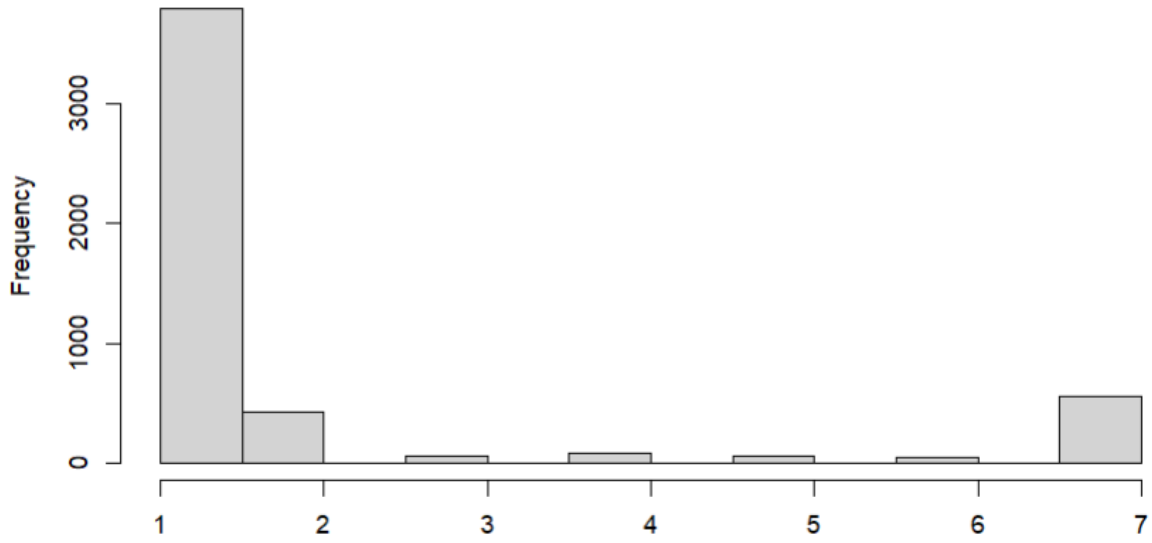

Рисунок 23. Гістограма кластерів даних факторного аналізу (k = 7)

3.2 Реалізація мовою програмування Python

Завдання. У файлі міститься запис кардіограми людини по 12 каналах. Час запису – 10 секунд (див. рис. 24). Дискретність: 500 точок за 1 секунду. Структура файлу: 1-й канал, 2-й канал, … 12-й канал (амплітуда у відносних одиницях). Довжина запису N = 5000, Δt = 1/500 = 0.002 [14].

```
[5]  df.head()
```

	0	1	2	3	4	5	6	7	8	9	10	11
0	-14.514	-29.542	-34.6050	24.821	6.3932	-30.119	84.482	5.1443	-27.501	-34.985	-7.7625	19.211
1	-17.619	-29.093	-30.7140	26.356	3.0571	-28.203	82.611	5.4154	-26.429	-33.400	-8.4981	18.072
2	-22.706	-28.381	-24.5790	28.752	-2.4137	-25.054	80.767	6.0363	-24.514	-31.825	-8.6064	16.926
3	-30.530	-27.219	-15.2590	32.270	-10.8970	-20.113	78.916	7.2030	-21.301	-30.276	-7.7247	15.759
4	-40.268	-25.520	-3.4883	36.405	-21.6340	-13.682	77.005	8.9045	-16.927	-28.766	-5.8652	14.563

Рисунок 24. Частина даних файлу A13.txt

Візуалізація даних. Побудовано графік кардіограми по кожному каналу. На рисунку 25 зображено перші чотири з дванадцяти каналів ЕКГ.

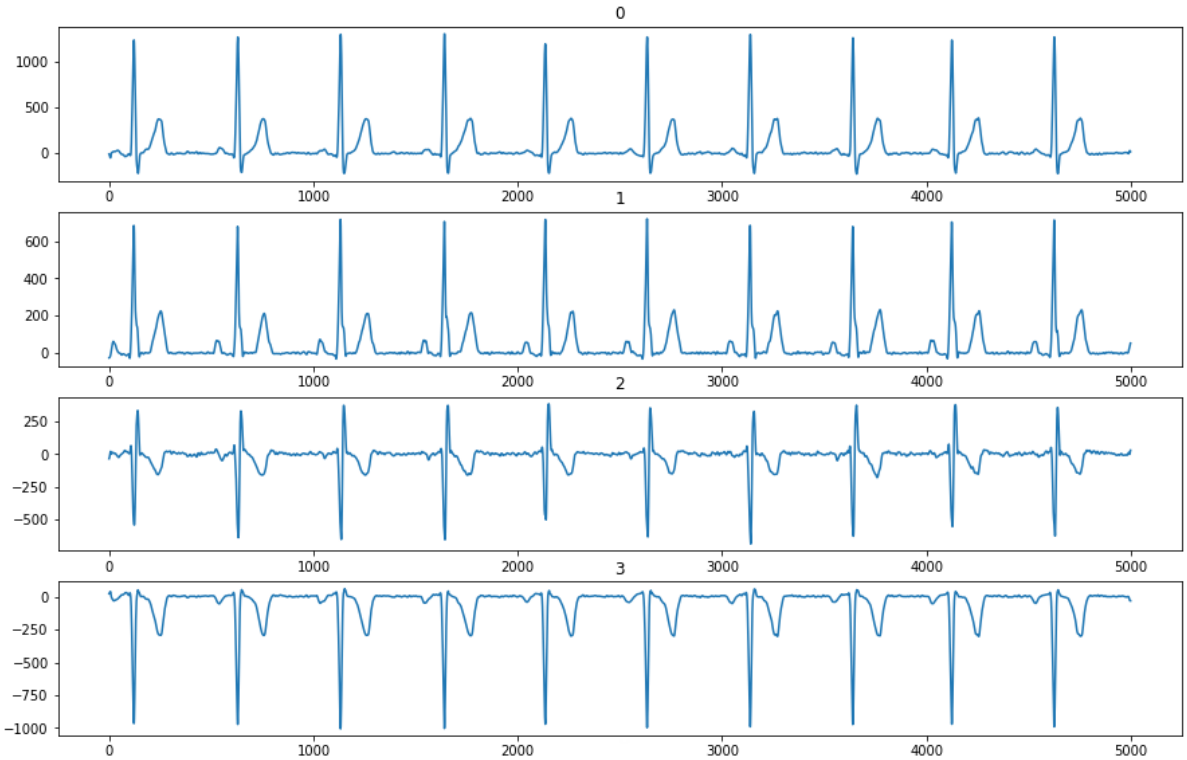

Рисунок 25. Перші чотири канали запису електрокардіограми

Попередня обробка. Для заданих змінних знайдено та оцінено основні статистичні параметри (див. рис. 26), медіану та дисперсію (див. рис. 27).

	zero	one	two	three	four	five	six	seven	eight	nine	ten	eleven
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	64.476566	41.427255	-19.946345	-54.508454	43.532617	7.423669	-25.518986	69.248284	112.151698	102.830755	98.485079	72.621542
std	201.009121	105.999845	108.364117	151.174117	152.893707	39.773720	122.006665	382.291574	367.475396	348.732384	338.286075	290.575493
min	-232.100000	-35.751000	-688.790000	-1006.200000	-298.780000	-163.940000	-794.630000	-1775.700000	-1239.300000	-903.480000	-648.330000	-554.930000
25%	-10.390500	-5.918550	-12.992250	-36.002000	-8.726225	-8.478375	-12.694500	-8.769375	-9.426000	-22.502250	-11.800250	-18.912500
50%	-2.537050	-2.246200	1.419950	1.792200	-1.567650	-3.018950	-2.344600	-1.464750	-0.760165	0.868825	0.488675	3.252450
75%	33.587000	44.579250	10.244000	7.148300	20.866250	6.901000	7.862625	126.115000	126.687500	78.802250	49.213000	40.171500
max	1304.100000	721.180000	386.780000	61.918000	976.350000	272.030000	85.309000	1035.200000	1710.500000	2120.800000	2315.100000	2070.500000

Рисунок 26. Основні статистичні параметри

```
zero      -2.537050
one       -2.246200
two        1.419950
three      1.792200
four      -1.567650
five      -3.018950
six       -2.344600
seven     -1.464750
eight     -0.760165
nine       0.868825
ten        0.488675
eleven     3.252450
dtype: float64
```

```
zero       40404.666685
one        11235.967081
two        11742.781839
three      22853.613745
four       23376.485500
five        1581.948774
six        14885.626345
seven     146146.847630
eight     135038.166542
nine      121614.275392
ten       114437.468379
eleven     84434.116869
dtype: float64
```

Рисунок 27. Медіана (зліва) та дисперсія (справа)

Побудовано гістограми для кожного каналу, перші три зображені на рисунку 28.

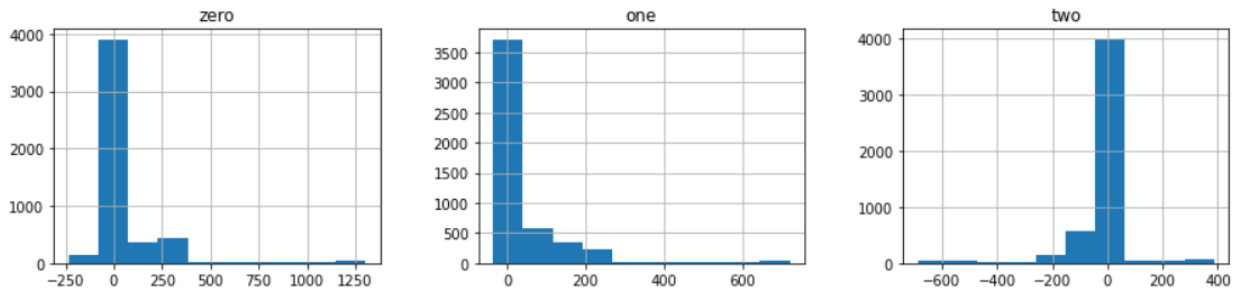

Рисунок 28. Гістограма даних перших трьох каналів

Однофакторний дисперсійний аналіз. Проведена перевірка чи є результати вимірювання різними рівнями одного фактору (12 рівнів). Результати перевірки на рисунку 29.

	df	sum_sq	mean_sq	F	PR(>F)
C(channel)	11.0	1.500373e-28	1.363975e-29	1.363975e-29	1.0
Residual	59988.0	5.998800e+04	1.000000e+00	NaN	NaN

Рисунок 29. Перевірка даних електрокардіограми

З наведеного аналізу робимо висновок, що між каналами немає статистичної різниці.

Двофакторний дисперсійний аналіз. Побудовано таблицю двофакторного експерименту (див. рис. 30) за правилом – кожен канал розбито на відрізки по 2 секунди, а кожну частину помічено відповідним числом від 1 до 5.

	zero	one	two	three	four	five	six	seven	eight	nine	ten	eleven	sample
0	-0.392970	-0.669522	-0.135272	0.524756	-0.242910	-0.943906	0.901598	-0.167683	-0.380033	-0.395191	-0.314076	-0.183810	1
1	-0.408417	-0.665286	-0.099366	0.534909	-0.264730	-0.895734	0.886263	-0.166974	-0.377116	-0.390646	-0.316251	-0.187729	1
2	-0.433724	-0.658569	-0.042751	0.550759	-0.300511	-0.816561	0.871149	-0.165350	-0.371904	-0.386129	-0.316571	-0.191673	1
3	-0.472648	-0.647607	0.043256	0.574030	-0.355996	-0.692333	0.855978	-0.162298	-0.363161	-0.381687	-0.313964	-0.195689	1
4	-0.521094	-0.631579	0.151877	0.601382	-0.426222	-0.530644	0.840315	-0.157848	-0.351258	-0.377357	-0.308468	-0.199805	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4995	-0.204884	-0.102352	0.194014	0.189116	-0.217711	0.381542	0.403896	-0.129993	-0.287623	-0.182039	-0.091098	-0.236615	5
4996	-0.205735	-0.051361	0.244703	0.172129	-0.236624	0.519950	0.433386	-0.118353	-0.271310	-0.170130	-0.076814	-0.219366	5
4997	-0.217620	-0.004002	0.312311	0.163860	-0.267400	0.681539	0.472851	-0.102653	-0.249747	-0.154602	-0.059060	-0.192389	5
4998	-0.235425	0.038215	0.385869	0.161320	-0.302900	0.849388	0.517668	-0.083822	-0.224948	-0.137185	-0.039508	-0.159547	5
4999	-0.254663	0.077196	0.458928	0.160798	-0.338418	1.015428	0.564281	-0.063311	-0.198872	-0.119200	-0.019407	-0.124562	5

Рисунок 30. Таблиця двофакторного експерименту

Перегруповуємо дані та виконуємо аналіз (див. рис. 31).

	df	sum_sq	mean_sq	F	PR(>F)
C(channel)	11.0	5.268409e-28	4.789463e-29	4.789357e-29	1.000000
C(sample)	4.0	6.554184e+00	1.638546e+00	1.638510e+00	0.161433
C(channel):C(sample)	44.0	4.012181e+01	9.118594e-01	9.118393e-01	0.638548
Residual	59940.0	5.994132e+04	1.000022e+00	NaN	NaN

Рисунок 31. Двофакторний аналіз

Отже, жодна з частин не має суттєвого впливу на кардіограму, тобто можна стверджувати, що серцебиття є стабільним.

Перетворення Фур'є. Виконано пряме та обернене перетворення Фур'є. Побудовано відповідні графіки для кожної змінної. На рисунку 32 зображено початковий графік одного з каналів ЕКГ та відповідне перетворення Фур'є.

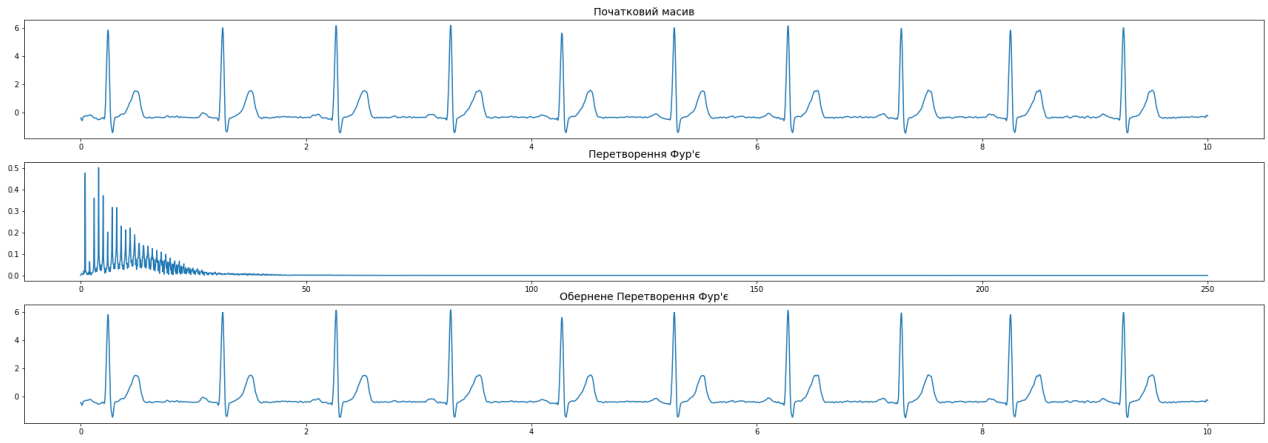

Рисунок 32. Графіки прямого та оберненого перетворення Фур'є

Кореляційний аналіз. Обчислення кореляційної матриці для нормалізованих змінних (див. рис. 33).

	zero	one	two	three	four	five	six	seven	eight	nine	ten	eleven
zero	1.000000	0.924682	-0.926323	-0.989687	0.991254	-0.049926	-0.593749	0.311544	0.793519	0.882034	0.932605	0.937533
one	0.924682	1.000000	-0.717296	-0.968616	0.867153	0.328988	-0.641200	0.186380	0.685902	0.793849	0.875065	0.884732
two	-0.926323	-0.717296	1.000000	0.864342	-0.967452	0.416780	0.464709	-0.372705	-0.773311	-0.835575	-0.853158	-0.853824
three	-0.989687	-0.968616	0.864342	1.000000	-0.962534	-0.090348	0.622207	-0.271430	-0.768473	-0.864960	-0.927609	-0.934063
four	0.991254	0.867153	-0.967452	-0.962534	1.000000	-0.179372	-0.557084	0.341367	0.801689	0.881564	0.920343	0.923573
five	-0.049926	0.328988	0.416780	-0.090348	-0.179372	1.000000	-0.229523	-0.293527	-0.172301	-0.106125	-0.021720	-0.003194
six	-0.593749	-0.641200	0.464709	0.622207	-0.557084	-0.229523	1.000000	0.551170	-0.019049	-0.234700	-0.382262	-0.444091
seven	0.311544	0.186380	-0.372705	-0.271430	0.341367	-0.293527	0.551170	1.000000	0.810612	0.635428	0.510799	0.439328
eight	0.793519	0.685902	-0.773311	-0.768473	0.801689	-0.172301	-0.019049	0.810612	1.000000	0.954580	0.910675	0.870248
nine	0.882034	0.793849	-0.835575	-0.864960	0.881564	-0.106125	-0.234700	0.635428	0.954580	1.000000	0.965870	0.943713
ten	0.932605	0.875065	-0.853158	-0.927609	0.920343	-0.021720	-0.382262	0.510799	0.910675	0.965870	1.000000	0.976955
eleven	0.937533	0.884732	-0.853824	-0.934063	0.923573	-0.003194	-0.444091	0.439328	0.870248	0.943713	0.976955	1.000000

Рисунок 33. Кореляційна матриця

Для аналізу кореляційної матриці виділяємо групу параметрів, парна кореляція між якими велика (коефіцієнт кореляції близький по модулю до 1) . Наприклад, параметри zero, one, four (див. рис. 33).

Знайдено часткові та множинні коефіцієнти кореляції. Зображено приклад для параметрів zero, one, four на рисунку 34.

```
[54] def partial_corr(ab, bc, ac):
       return (ab - ac * bc) / (((1-ac*ac) ** (1/2)) * ((1-bc*bc) ** (1/2)))

     partial_corr(corr_matrix["zero"]["one"], corr_matrix["zero"]["four"], corr_matrix["one"]["four"])

     0.99069843394116

[56] def mult_corr_coef(ab, bc, ac):
       return np.sqrt((ab*ab + ac*ac - 2*ab*ac*bc) / (1 - bc*bc))

     mult_corr_coef(corr_matrix["zero"]["one"], corr_matrix["zero"]["four"], corr_matrix["one"]["four"])

     0.9977008701016992
```

Рисунок 34. Часткові та множинні коефіцієнти кореляції параметрів

Факторний аналіз. Знаходимо власні числа кореляційної матриці (див. рис. 35).

```
fa = FactorAnalyzer(rotation=None)
fa.fit(df)
ev,v = fa.get_eigenvalues()
```

```
[31] ev
```

```
array([8.53293487e+00, 2.16658961e+00, 1.11041415e+00, 1.06583048e-01,
       3.25363271e-02, 2.62784216e-02, 1.74825113e-02, 3.06244673e-03,
       2.23901729e-03, 1.38858217e-03, 4.14589203e-04, 7.64317872e-05])
```

Рисунок 35. Таблиця власних чисел

На рисунку 36 побудовано графік кам’янистого осипу (діаграму осипання).

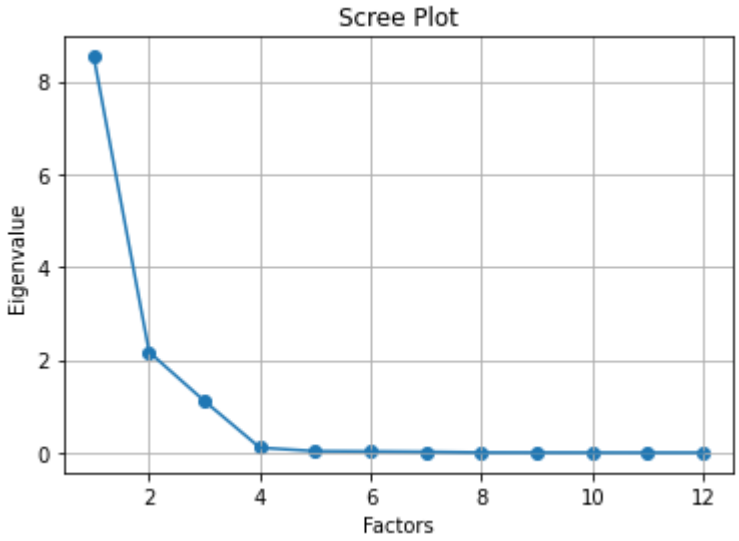

Рисунок 36. Графік кам’янистого осипу

Проводимо факторний аналіз (див. рис. 37).

```
#%%
fa.set_params(n_factors=3, rotation = 'varimax')
fa.fit(df)
fa.loadings_
```

```
array([[ 0.99282164, -0.05859168, -0.05957387],
       [ 0.93589944, -0.11263927,  0.31377826],
       [-0.90248621,  0.00976093,  0.42305758],
       [-0.98931483,  0.07911277, -0.07904046],
       [ 0.97798433, -0.03979994, -0.18771632],
       [-0.00249897, -0.16511156,  0.98407165],
       [-0.55944793,  0.7967041 , -0.10276452],
       [ 0.35116314,  0.90980966, -0.14640239],
       [ 0.82836385,  0.55364309, -0.07928772],
       [ 0.91419292,  0.34650449, -0.04469842],
       [ 0.96185993,  0.20412991,  0.01770539],
       [ 0.96318203,  0.1290813 ,  0.02308944]])
```

```
[34] fa.get_factor_variance()
```

```
(array([8.42241442, 1.99872864, 1.33199069]),
 array([0.70186787, 0.16656072, 0.11099922]),
 array([0.70186787, 0.86842859, 0.97942781]))
```

Рисунок 37. Факторний аналіз

Загальна описана частка дисперсії приблизно дорівнює 0.98. Також трансформуємо дані для подальшого використання (див. рис. 38).

```
[35] pca_df = pd.DataFrame(fa.transform(df.values))
     pca_df
```

	0	1	2
0	-0.422592	0.031750	-0.972048
1	-0.428145	0.054296	-0.919720
2	-0.436213	0.094130	-0.831982
3	-0.447347	0.158629	-0.692615
4	-0.459861	0.241882	-0.510290
...	...	...	...
4995	-0.216727	-0.115848	0.365756
4996	-0.201828	-0.063041	0.515681
4997	-0.190459	0.010951	0.691416
4998	-0.181645	0.095123	0.873690
4999	-0.173843	0.181327	1.053416

5000 rows × 3 columns

Рисунок 38. Трансформовані дані

Кластерний аналіз. Будемо вважати, що записана кардіограма (12 каналів) являє собою множину багатовимірних точок деякого евклідового простору. Результати спостережень являють собою 5000 точок, кожна точка є вектором розмірності 12 (у випадку головних компонент – розмірності 3).

Для кластеризації було обрано метод k-середніх. Спочатку виконаємо кластерний аналіз для початкових даних для п’яти (див. рис. 39) та семи кластерів (див. рис. 40).

Рисунок 39. Гістограма кластерів початкових даних (k = 5)

Рисунок 40. Гістограма кластерів початкових даних (k = 7)

Тепер виконаємо кластеризацію даних трьох головних факторів для п’яти (див. рис. 40) та семи кластерів (див. рис. 41).

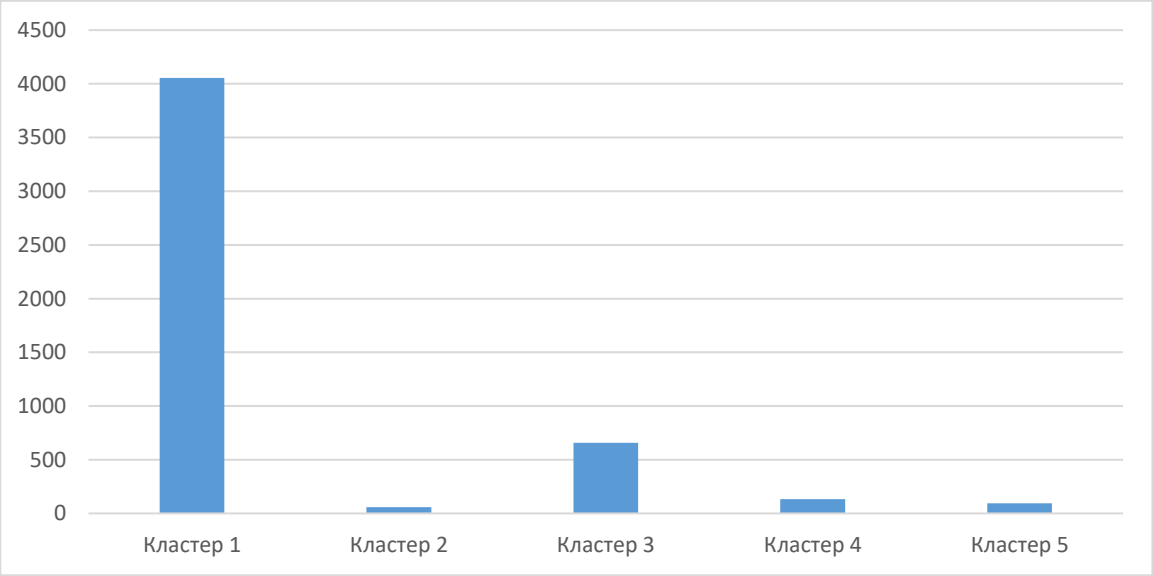

Рисунок 41. Гістограма кластерів даних факторного аналізу (k = 5)

Рисунок 42. Гістограма кластерів даних факторного аналізу (k = 7)

3.3 Порівняння двох реалізацій

Порівняння програмних реалізацій інтелектуальних методів обробки та аналізу даних: перша мова програмування R, друга – Python.

1. Статистичний аналіз:

    1) R: Мова R має багатий набір пакетів, таких як dplyr, ggplot2 та інші, що дозволяють здійснювати широкий спектр статистичного аналізу від попередньої обробки до візуалізації даних.

    2) Python: З використанням бібліотеки Pandas, Python надає потужні функціональні можливості для статистичного аналізу даних. Модулі як Statsmodels та Matplotlib дозволяють проводити попередню обробку та візуалізацію даних.

2. Дисперсійний аналіз:

    1) R: У R доступні пакети, такі як stats та Anova, які надають функції для проведення однофакторного та двофакторного дисперсійного аналізу.

    2) Python: Бібліотека SciPy у Python надає інструменти для проведення дисперсійного аналізу, включаючи функції для однофакторного та двофакторного експерименту.

3. Перетворення Фур’є:

    1) R: У R для обчислення прямого та оберненого перетворення Фур’є можна використовувати пакети, такі як fft та signal.

    2) Python: Бібліотека NumPy в Python надає функції для обчислення перетворення Фур’є, проте потрібно самостійно запрограмувати пряме та обернене перетворення Фур’є за відповідними формулами.

4. Кореляційний аналіз:
    1) R: У R кореляційний аналіз можна виконувати за допомогою функції cor, яка обчислює кореляційну матрицю для подальшого аналізу.
    2) Python: Бібліотека Pandas у Python має функцію для обчислення кореляційної матриці між змінними для подальшого аналізу.
5. Факторний аналіз:
    1) R: У R доступні пакети, такі як psych та factanal, які надають функціональні можливості для проведення факторного аналізу.
    2) Python: Для факторного аналізу в Python можна використовувати бібліотеку FactorAnalyzer.
6. Кластерний аналіз:
    1) R: У R для кластерного аналізу можна використовувати пакети, такі як cluster та factoextra, які надають функції для проведення кластерного аналізу та візуалізації кластерів.
    2) Python: Бібліотеки SciPy та scikit-learn в Python надають інструменти для кластерного аналізу, включаючи алгоритми кластеризації та візуалізацію кластерів.

Обидві програмні реалізації мають свої переваги та недоліки. Вибір між R та Python залежить від особистих уподобань, рівня знань та потреб у додаткових функціях чи інших бібліотеках.

ВИСНОВКИ

Інтелектуальних методів обробки та аналізу даних поєднують формальні (кількісні) та неформальні (якісні) підходи разом з різноманітним математичним апаратом, включаючи класичний статистичний аналіз, кібернетичні методи та сучасні інформаційні технології. Дані методи широко використовують та застосовують при дослідженні різноманітних систем і процесів. Саме тому виникла потреба у використанні інтелектуальних методів обробки електрокардіограм для діагностики та моніторингу серцево-судинних захворювань (ССЗ), оскільки ССЗ займають перше місце в світі та Україні серед причин смерті.

У даній роботі було проведено дослідження можливостей застосування інтелектуальних методів та алгоритмів для обробки та аналізу даних ЕКГ. При написанні роботи поставлені та виконані такі завдання:

- проведено попередню обробку даних та їх візуалізація;
- здійснено однофакторний та двофакторний дисперсійний аналіз;
- виконано пряме та обернене перетворення Фур’є;
- здійснено кореляційний аналіз та факторний аналіз;
- виконано кластеризацію даних за допомогою методу k-середніх.

На основі проведених досліджень можна зробити наступні висновки:

1) Інтелектуальні методи та алгоритми для обробки та аналізу даних електрокардіограм демонструють свою ефективність у виявленні та аналізі потенційних патологічних змін у серцевій діяльності.

2) Результати проведеного дослідження підтверджують високу релевантність та перспективи використання інтелектуальних методів обробки та аналізу даних у медичній практиці для діагностики та прогнозування серцево-судинних захворювань.

3) Використання інтелектуальних методів та алгоритмів може сприяти розробці нових апаратних та програмних засобів для покращення діагностики та моніторингу серцево-судинної системи.

4) Продовження досліджень у даній тематиці є доцільним з метою вдосконалення методів обробки та аналізу електрокардіограм та розробки нових підходів для покращення діагностики та лікування серцевих захворювань.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Колодчак О. М. Інтелектуальний аналіз даних / О. М. Колодчак. // Вісник Національного університету "Львівська політехніка". – 2013. – №773. – С. 49-58.
2. Бахрушин В.Є. Методи аналізу даних : навчальний посібник для студентів / В.Є. Бахрушин. – Запоріжжя : КПУ, 2011. – 268 с.
3. Тютюник В. В. Методологічні основи планування експерименту / В. В. Тютюник. – Харків. – 422 с.
4. Яременко Л. І. Кількосні методи в поведінкових науках / Л. І. Яременко, І. В. Лупан. – Кропивницький, 2019. – 224 с.
5. Пашко А. О. Методичні матеріали до курсу "Інтелектуальна обробка даних" / А. О. Пашко. – Київ, 2019. – 55 с.
6. Denysiuk V. Review of statistical data analysis software / V. Denysiuk. // Polish Journal of science. – 2020. – №27. – P. 14-23.
7. Котенко В. В. Застосування методу Пірсона для отримання залежностей розподілу хімічних елементів у межах родовища каоліну / В. В. Котенко, С. І. Башинський, І. А. Піскун. // Технічна інженерія. – 2021. – №88.
8. Яровий А.Т., Страхов Є.М. Багатовимірний статистичний аналіз. – Одеса: Астропринт, 2015. – 132 с.
9. Бізнес-аналітика багатовимірних процесів [Електронний ресурс] / [Т. С. Клебанова, Л. С. Гур’янова, Л. О. Чаговець та ін.]. – 2020. – Режим доступу до ресурсу: http://ebooks.git-elt.hneu.edu.ua/babap/.
10. Evans R. R Programming / Robin Evans., 2014. – 82 p.
11. Майборода Р.Є. Комп'ютерна статистика – професійний старт (з використанням R). Підручник / Р.Є. Майборода. – Київ: В-во КНУ, 2018. – 482 с.
12. Yogesh R. Python: Simple though an Important Programming language [Електронний ресурс] / R. Yogesh // International Research Journal of

Engineering and Technology. – 2019. – Vol. 06. – P. 1856-1858. – Режим доступу: https://www.irjet.net/archives/V6/i2/IRJET-V6I2367.pdf.

13. JetBrains Strikes Python Developers with PyCharm IDE [Електронний ресурс] // eWeek. – 2013. – Режим доступу до ресурсу: https://archive.ph/20130122124720/http://www.eweek.com/c/a/Application-Development/JetBrains-Strikes-Python-Developers-with-PyCharm-10-IDE-304127/.
14. Пашко А. О. Методичні рекомендації з дисципліни "Інтелектуальна обробка даних" / А. О. Пашко. – Київ, 2022. – 12 с.