КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп'ютерних наук та кібернетики

Кафедра прикладної статистики

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 124 Системний аналіз

на тему:

Кластеризація населення України за даними демографічних таблиць

Виконала студентка 4 курсу
Туру Яна Федорівна

(підпис)

Керівник дипломної роботи
Доцент, кандидат фізико-математичних наук
Лівінська Ганна Володимирівна

(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 прикладної статистики

«06» червня 2022 р.,
протокол № 11
Завідувач кафедри
Розора І. В.

(підпис)

Київ - 2022

ЗМІСТ

Анотації.

Дипломна робота складається зі вступу, 4 розділи, висновків, списку використаних джерел (23 найменування). Загальний обсяг роботи становить 45 сторінок, основний текст роботи викладено на 29 сторінках. Ключові слова: ієрархічна кластеризація, метод К-середніх, групування даних у вибірці.

В роботі виконано кластеризацію хвороб населення України за даними демографічних таблиць 2016 року. Для реалізації методик кластерного аналізу з метою дослідження даних була написана комп'ютерна програма, що дозволяє проводити кластеризацію за вказаними ознаками. Мова програмування – R. Середовище виконання – RStudio. Зроблено висновки щодо кластеризації за данними демографічних таблиць в Україні.

Thesis consists of an introduction, 4 chapters, conclusions, a list of sources used (23 titles). The total volume of the work is 45 pages, the main text of the work is set out on 29 pages. Keywords: hierarchical clustering, K-means method, grouping of data in the sample.

The paper clusters diseases of the population of Ukraine according to demographic tables in 2016. To implement the methods of cluster analysis in order to study the data, a computer program was written that allows clustering on these grounds. Programming language - R. Work environment - RStudio.. Conclusions are made on clustering according to demographic tables in Ukraine.

Вступ.

Дана робота присвячена кластеризації таблиць захворюваності України 2016 року. Важливість цього напрямку не викликає сумніву, оскільки в останні роки кількість інформації в світі збільшується шаленими темпами. В більшості випадків обсяг наявних даних не піддається ручній обробці. В такому випадку виникає потреба впорядкувати та часто класифікувати інформацію, об’єднуючи в одну групу тематично близькі об’єкти.

Кластерний аналіз — це набір методів, метою яких є класифікація на групи схожих між собою об’єктів. Зокрема, алгоритми побудови кластерів можуть суттєво відрізнятись залежно від того, що відносити в один кластер і як їх ефективно виділяти. Серед популярних концепцій кластеризації є виявлення груп елементів, які утворюються ґрунтуючись на відстані між ними, на щільності ділянок у просторі даних, інтервалах або на конкретних статистичних розподілах. Тому задача кластеризації може бути сформульована як задача багатокритеріальної оптимізації.

Класифікувати можна будь-які об'єкти, що підлягають групуванню. Використовувати для цього можна людей, тварин, хімічні елементи, зірки, тощо.

В цій роботі ми проаналізуємо методики кластерного аналізу із застосуванням до демографічної таблиці України 2016 року за такими ознаками: хвороби ендокринної системи, хвороби вуха, хвороби ока та придаткового апарату, хвороби органів дихання, хвороби органів травлення та хвороби сечостатевої системи.

1. Основні ідеї та методи кластерного аналізу.

1.1. Кластеризація як нечітка класифікація

Кластерний аналіз має своєю задачею розбиття заданої вибірки об’єктів, що характеризуються кількома ознаками на підмножини, які називаються кластерами, так, щоб кожен кластер складався зі схожих об’єктів, а об’єкти різних кластерів істотно відрізнялися. Саме завдання кластеризації відноситься до статистичної обробки багатовимірних даних.

Кластеризація - це нечітка класифікація, заснована на взаємному розміщенні об'єктів, що класифікуються. В основному своєму значенні термін cluster може бути переведений як скупчення. Близькі по значенню об'єкти відносяться до одного класу, а далекі об'єкти, навпаки - до різних класів.

1.2. Мінімізація функціонала якості

Ідея алгоритму кластеризації - мінімізувати деякий функціонал якості, тобто підібрати оптимальну з певних міркувань класифікацію.

Відзначимо кілька природних критеріїв. По-перше, середня внутрикластерна відстань повинна бути якомога меншою. З іншого боку, середня міжкластерна відстань повинна бути якомога більша. Щоб врахувати і міжкласові, і внутрикласові відстані, як функціонал якості інколи використовують відношення цих величин.

1.3. Формальний опис класифікації

Уявімо деяку множину $X = \{x_1, ..., x_m\}$, елементи якої ми будемо називати об'єктами. На множині об'єктів визначимо функцію $f: X \to \{1, ..., n\}$, яку будемо називати ознакою. Ця функція кожному

об'єкту ставить у відповідність номер групи, до якої він належить. Всі об'єкти множини X, які відносно f належать одній і тій самій групі, називають класом x_i. Розбиття $C(X, f)$ множини X на класи що не перетинаються x_t, $t = 1, ..., m$ і є класифікацією.

1.4. Метричний простір об'єктів

Для класифікації об'єктів необхідним є поняття відстані d: X×X→R між об'єктами, які характеризуються багатьма різними ознаками. Двом об'єктам x_i та x_j, поставимо в відповідність деяке число $d(x_i, x_j)$. Природно задати відстань так, що для неї виконані стандартні властивості метрики, зокрема відстань об'єкта до самого себе повинна бути нульовою. Об’єкти, відстань між якими мала, слід віднести до однієї групи і навпаки, чим вони далі друг від друга, тим більш ймовірно, що вони належать різним групам.

В якості вхідних даних маємо вектор характеристик для кожного об'єкта. Найчастіше відстанню між об'єктами вважають евклідову відстань між векторами ознак.

1.5. Кластерний аналіз як пошук оптимальної класифікації

Критерій якості повинен відображати, наскільки об'єкти одного класу близькі один до одного, а об'єкти різних класів один від одного далекі. Класифікація, що оптимізує функціонал якості, таким чином, групує кожне скупчення об'єктів до відповідного класу. Отже, можна сформулювати наступне визначення: кластеризація - пошук класифікації, оптимальної щодо критерію або кількох критеріїв якості.

Процес мінімізації критерію якості вимагає знання всього простору об'єктів. Заміна кількох об'єктів в просторі може призвести

і до зміни оптимальної класифікації.

Кластеризація істотно залежить від заданої метрики та критерію якості. Але структура цієї залежності непередбачувана та універсальних алгоритмів кластеризації немає.

Окрім того, пошук глобального мінімуму критерію якості не завжди можливий, особливо, коли простір об'єктів великий. В такому випадку ведеться пошук локального мінімуму одним з оптимізаційних алгоритмів.

2. Алгоритми кластерного аналізу

2.1. Неявне завдання критеріїв якості в алгоритмі кластеризації

Головною задачею кластерного аналізу є пошук оптимальної класифікації. Існує чимало алгоритмів кластеризації, які вирішують цю задачу.

Виділяється як мінімум три цілі кластеризації:

1) описати структуру простору об'єктів, розбиваючи його на класи схожих об'єктів і досліджуючи області згущення;
2) скоротити обсяг даних, виділяючи найбільш типові представники кожного кластера;
3) виділити специфічні (або аномальні) об'єкти, що не належать до жодного з кластерів;

Алгоритми кластеризації, як правило, мають параметри регулювання, за допомогою яких ведеться пошук найбільш оптимального результату. Вид цих параметрів різний - це може бути число кластерів, швидкість збіжності чи інші константи в залежності від алгоритму.

2.2. Метод к-середніх

Один з найбільш вживаних алгоритмів кластеризації, метод k-середніх, ділить простір об'єктів на кластери, мінімізуючи внутрішньокластерне розсіювання - середньоквадратичне відхилення об'єктів кластера від відповідного центру мас. Перед початком алгоритму k-середніх необхідно визначити та задати число кластерів, яке є оптимальним з певних міркувань.

Средньоквадратичне відхилення характеризує, наскільки елементи одного кластера близькі між собою за комплексом ознак. Однак така метрика має недолік: при такій постановці передбачається, що кластери опуклі і ізотропні, тобто розподіл значень всередині кластера не залежить від координат. Деякі модифікації дозволяють послабити цю вимогу.

Принцип пошуку оптимального значення функціоналу якості схожий на дію алгоритму Форел. Розглядаються наближені значення центрів кластерів. Кожен об'єкт вибірки ставиться до того кластеру, до центру якого він ближче, на кожному етапі після формування кластерів значення центрів перераховуються.

Цей метод - один з найбільш вживаних в прикладних задачах, існує чимало модифікацій, що дозволяють врахувати тонкі настройки оптимізації або форми кластерів. Він чутливий до початкових наближень центрів кластерів, оскільки різні стартові позиції можуть привести до різних локальних мінімумів функціоналу якості. Існують різні способи боротьби з цією проблемою, один з яких - паралельний запуск декількох випадкових наближень. Інший метод, призначений для вирішення проблеми пошуку глобального мінімуму, називають схемою ініціалізації k-means. В якості апріорного наближення вибираються k найбільш віддалених одна від одної точок. Доведено, що таким шляхом

виходить явно краща кластеризація, ніж при генерації випадкових центрів.

2.3. Ієрархічна кластеризація

Окрему групу алгоритмів представляють собою ієрархічні методи кластеризації. Вони будують не один класифікатор всього простору, а цілу систему класифікаторів. В результаті роботи цього методу виходить ієрархічне дерево - дендрограмма, в корені якого кластери, які містять окремі елементи, а вершина - весь простір об'єктів, як єдиний кластер. Або навпаки: в корені - весь простір об'єктів, а на вершині - кожен об'єкт в окремому кластері.

Ієрархічні алгоритми бувають двох видів: дивізимні (розділювальні) та агломеративні (об’єднувальні). Перші ділять простір на все більш дрібні частини. Другі використовуються частіше і провадять операцію, зворотну традиційного поділу при кластеризації. Вони об'єднують кластери.

На першому кроці алгоритму агломеративної кластеризації кожен об'єкт вважається окремим кластером, між кластерами задається функція відстані. Об'єднання відбувається наступним чином: на кожному кроці два найближчих кластера об'єднуються в єдиний кластер.

3. Оцінка якості кластеризації

3.1 Основне про оцінку якості кластеризації

Якщо при тестуванні алгоритму відома реальна класифікація, до якої слід прагнути, то відповідна оціночна метрика повинна відображати, наскільки прогнозне розбиття близьке до еталонного. На практиці найчастіше така класифікація відома лише для

невеликого числа об'єктів, тому розумно використовувати інші критерії, які оцінюють якість класифікації виходячи з її внутрішньої структури. По суті, вони являють собою різновид тих самих критеріїв якості, які алгоритм кластеризації повинен оптимізувати.

3.2 Перевірка якості кластеризації

Після отримання результатів кластерного аналізу необхідно перевірити правильність кластеризації (тобто оцінити, наскільки кластери відрізняються один від одного).

Існують такі методи перевірки якості кластеризації:

- Встановлення контрольних точок та перевірка на отриманих кластерах
- Визначення стабільності кластеризації шляхом додавання до моделі нових змінних
- Створення та порівняння кластерів з використання різних методів

В цій роботі ми використаємо лише останній метод.

4. Опис розробленого програмного продукту

4.1 Обробка даних

Бібліотеки які використовувалися:

```
library(dplyr)
library(factoextra)
library(NbClust)
library(cluster)
library(parameters)
```

```
library(fpc)
library(FactoMineR)
library(devtools)
library(corrplot)
library(psych)
library(GPArotation)
```

Дані для роботи взято на сайті державної служби статистики України: http://www.ukrstat.gov.ua/

Зчитуємо дані з файлу:

```
life_data <- read.csv("Lifet.csv") %>%
  mutate(ind = paste(Year, Age, sep=' '))
```

Цей файл містить 25 позицій даних захворюваності в різних областях України в 2016 році за такими ознаками:

- хвороби ендокринної системи
- хвороби вуха
- хвороби ока та придаткового апарату
- хвороби органів дихання
- хвороби органів травлення
- хвороби сечостатевої системи

Змінюємо назву рядків та видаляємо колонку з даними про область:

```
row.names(life_data) <- life_data$ind
life_data <- life_data[-1]
```

Дивимося на дані (команда з бібліотеки dplyr) та виводимо підсумки по стовпцям:

```
glimpse(life_data)
```

Rows: 25

Columns: 6

```
$ endocrine             <int> 26062, 10382, 33026, 13395, 11830, 20389, 12967, …
$ eye_diseases          <int> 47377, 33027, 158758, 43436, 32261, 44227, 52608,…
$ ear_diseases          <int> 34658, 25074, 116143, 39485, 20548, 23539, 38930,…
$ respiratory_diseases  <int> 552633, 366305, 1200543, 450605, 381388, 346649, …
$ digestive_diseases    <int> 43546, 23103, 111601, 42879, 34647, 52799, 28741,…
$ genitourinary_diseases <int> 55446, 40701, 270413, 67605, 51127, 29900, 70898,…
```

```
summary(life_data)
```

```
   endocrine      eye_diseases     ear_diseases    respiratory_diseases
Min.   : 4088   Min.   : 15019   Min.   : 10386   Min.   : 168947
1st Qu.:10267   1st Qu.: 34850   1st Qu.: 23539   1st Qu.: 346649
Median :13395   Median : 43436   Median : 31510   Median : 381388
Mean   :15757   Mean   : 54854   Mean   : 40279   Mean   : 503269
3rd Qu.:20389   3rd Qu.: 55048   3rd Qu.: 39485   3rd Qu.: 559493
Max.   :33026   Max.   :158758   Max.   :116143   Max.   :1214962
digestive_diseases genitourinary_diseases
Min.   :  9379     Min.   : 18581
1st Qu.: 27036     1st Qu.: 37349
Median : 40118     Median : 53815
Mean   : 44583     Mean   : 70434
3rd Qu.: 52799     3rd Qu.: 68176
Max.   :111601     Max.   :270413
```

4.2 Матриця відстаней (неподібності) між елементами

Вибравши певний спосіб вимірювання відстані (міри несхожості) між об’єктами, варто подивитися на матрицю відстаней, елементами якої є відстані між всіма парами об’єктів.

Функція dist() базового пакету stats та get_dist() з пакету factoextra рахує матрицю відстаней між об’єктами (рядками вхідного датасету). За замовчуванням в обох функціях використовується евклідова відстань, проте доступні кілька основних методів підрахунку відстаней, а друга функція має

додатково методи, що рахують відстані на базі кореляції.

Функція fviz_dist() з пакету factoextra візуалізує матрицю відстаней.

```
life_data.dist <- get_dist(life_data, stand = TRUE)

fviz_dist(life_data.dist, gradient = list(low = "green2", mid = "white", high = "darkblue"))
```

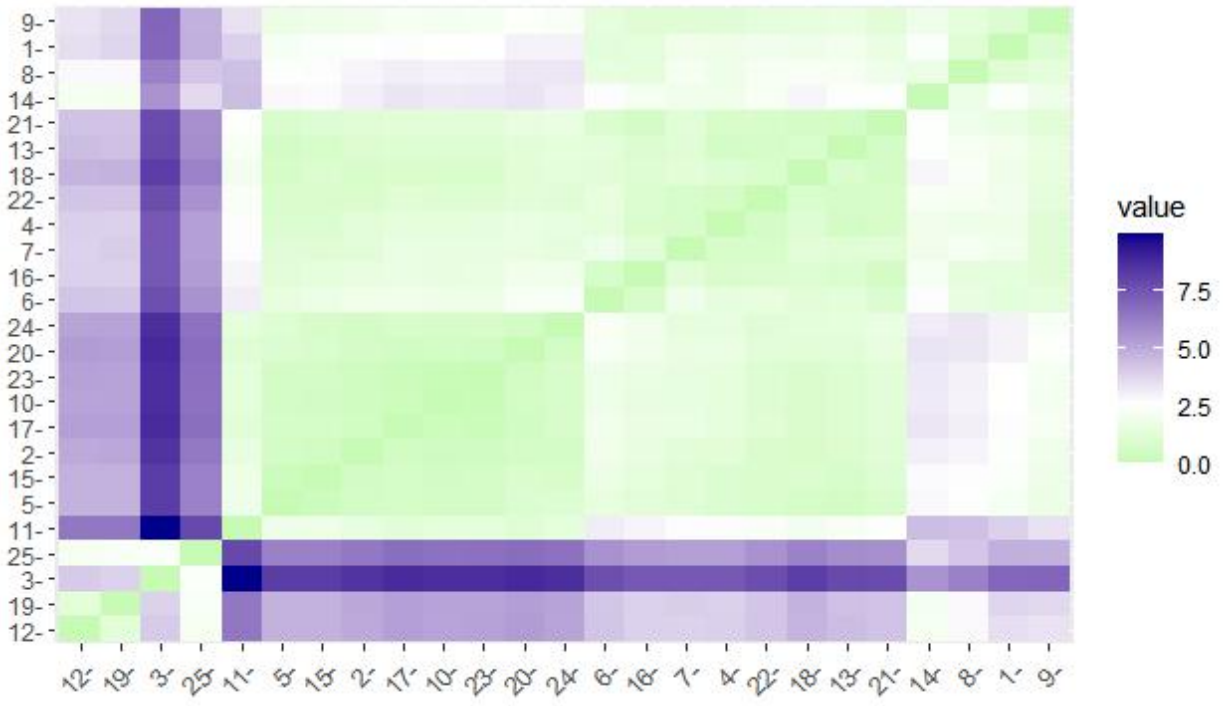

Рисунок 4.2.1- Матриця відстаней між елементами

Де:

1	Вінницька область
2	Волинська область
3	Дніпропетровська область
4	Донецька область
5	Житомирська область
6	Закарпатська область
7	Запорізька область

8	Івано-Франківська область
9	Київська область
10	Кіровоградська область
11	Луганська область
12	Львівська область
13	Миколаївська область
14	Одеська область
15	Полтавська область
16	Рівненська область
17	Сумська область
18	Тернопільська область
19	Харківська область
20	Херсонська область
21	Хмельницька область
22	Черкаська область
23	Чернівецька область
24	Чернігівська область
25	м.Київ

Отже можна сказати, що на даному етапі вже можна побачити виділення 2 кластерів.

4. 3 **Ієрархічна кластеризація**

Найбільш популярний метод. Простий, інтуїтивно зрозумілий, не вимагає задання початкових умов (кількості кластерів, центрів кластерів). Проте не ефективний по часу та об’єму використаної пам’яті.

Результат ієрархічної кластеризації зручно зображувати у вигляді дендрограми – діаграми спеціальної форми, яка демонструє процес об’єднання елементів в кластери. Базуючись на вигляді дендрограми, можна приймати рішення щодо кількості кластерів, на які буде розбита сукупність об’єктів, спираючись на бажаний рівень подібності між кластерами.

Функція hclust з пакету stats проводить ієрархічну кластеризацію. Основним аргументом має бути матриця відстаней (неподібностей) між об’єктами, аргумент method задає метод, за яким проводиться агломерація (метод визначення відстані між кластерами, за замовчуванням method = "complete").

На відміну від функції, що реалізує метод k-середніх, який повертає вектор, що містить інформацію щодо того, якому кластеру належить кожен об’єкт, функція hclust створює ієрархічну структуру (дендрограму), “дерево”, з якого ми можемо “відрізати” гілки, щоб отримати номери кластерів. Ця інформація може бути викликана функцією cutree().

Для зображення дендрограми можна скористатися універсальною функцією plot() або функцією fviz_dend() з пакету factoextra. Функція отримує на вхід в якості вхідного аргумента об’єкт відповідного типу (зокрема hclust) і має багато додаткових графічних параметрів.

Створюємо вибірку із датасету:

```
life_sample_ind <- sample(seq(1, nrow(life_data)), size = 25)
life_data_smpl <- life_data[life_sample_ind,]
head(life_data_smpl)
```

	endocrine	eye_diseases	ear_diseases	respiratory_diseases	digestive_diseases	genitourinary_diseases
17	10017	28380	21274	262000	25409	37349
5	11830	32261	20548	381388	34647	51127
1	26062	47377	34658	552633	43546	55446
10	9973	34850	18987	298407	27036	35004
4	13395	43436	39485	450605	42879	67605
18	15621	33505	25323	354097	29147	37258

Якщо ми попередньо не проведемо стандартизацію даних, то у нас на створення кластерів буде впливати лише частина даних (там, де більші значення). Стандартизація даних – процедура, за якої ми створюємо нові дані на основі наявних так, щоб середнє дорівнювало 0, а вибіркова дисперсія - 1.

```
scale_life <- scale(life_data_smpl)
```

4. 4 **Групування даних у вибірці**

Вертикальні риски дендрограми демонструють міжкластерну відстань (відстань між двома найближчими кластерами). Відповідно, з виду дендрограми ми можемо припустити, яка кількість кластерів є оптимальною за кластеризації конкретним методом.

Ми проведемо кластеризацію методом Варда.

Метод Варда мінімізує суму квадратів будь-яких двох кластерів, які можуть утворитися на кожному кроці, тобто квадрат використаної евклідової відстані між кластерами: $d_{ij} = d(\{X_i\},\{X_j\}) = \| X_i - X_j \|^2$ повинен бути найменшим.

Нехай X_{ijk} – значення k-тої змінної в j-му спостереженні, що належить і-тому кластеру. При цьому для реалізації даного методу ми маємо визначити наступне:

$ESS(X) = \sum_i \sum_j \sum_k |x_{ijk} - \bar{x}_{ijk}|^2$, тут підсумовуються всі змінні у всіх частинах кожного кластера і порівнюється окреме спостереження для кожної змінної із середньою змінною з кластера. Якщо ESS має малі значення, то дані близькі до середніх за кластером, у такому випадку ми вже маємо кластер, як одиницю аналізу. ESS - сума квадратів помилок. Математичний образ ESS описується наступним чином:

D(X, Y)=ESS(XY)-(ESS(X)+ESS(Y)), де XY - кластер отриманий в результаті злиття 2 кластерів.

```
life_data.hc <- hclust(life_data.dist, method = "ward.D2")  # кластеризація

fviz_dend(life_data.hc)
```

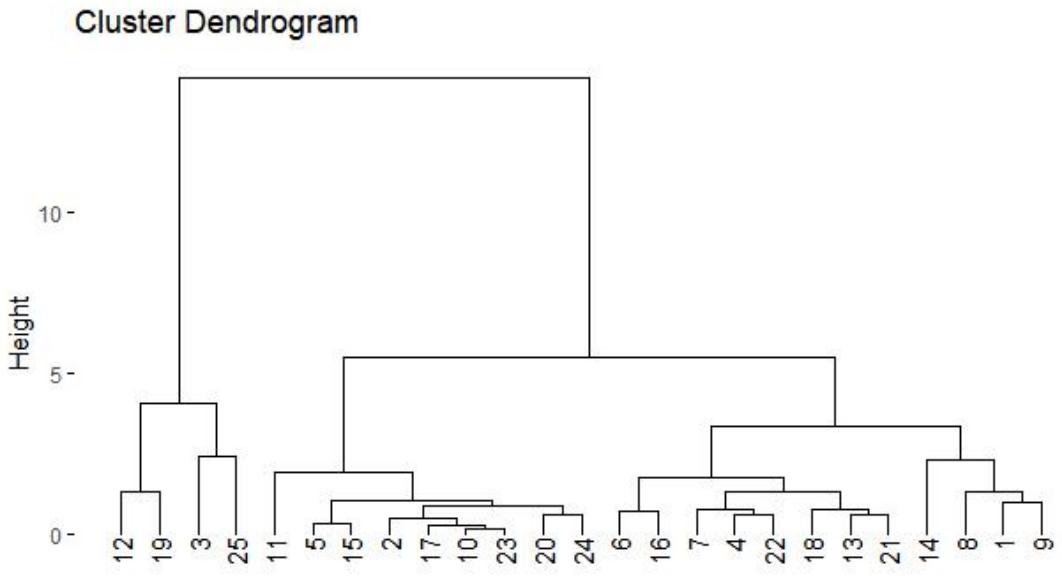

Рисунок 4.4.1- дендограма кластеризації методом Варда

Де:

1 Вінницька область

2	Волинська область
3	Дніпропетровська область
4	Донецька область
5	Житомирська область
6	Закарпатська область
7	Запорізька область
8	Івано-Франківська область
9	Київська область
10	Кіровоградська область
11	Луганська область
12	Львівська область
13	Миколаївська область
14	Одеська область
15	Полтавська область
16	Рівненська область
17	Сумська область
18	Тернопільська область
19	Харківська область
20	Херсонська область
21	Хмельницька область
22	Черкаська область
23	Чернівецька область
24	Чернігівська область
25	м.Київ

На останній дендрограмі оптимальною кількістю кластерів є 2 або 3.

```
fviz_dend(life_data.hc, k = 2, # розділяємо на 2 кластери
          cex = 0.5, # розмір шрифту для назв об'єктів
```

```
        k_colors = c("magenta", "turquoise3", "orange", "tomato3
"),
        color_labels_by_k = TRUE,
        rect = TRUE,
        main = "Дендрограма для кластеризації за методом В
арда")
)
```

Рисунок 4.4.3- дендограма кластеризації методом Варда на 2 кластери

Для поділу на три кластери за методом Варда маємо.

```
fviz_dend(life_data.hc, k = 3,
        cex = 0.5,
        k_colors = c("olivedrab", "black"),
        color_labels_by_k = TRUE,
```

```
            rect = TRUE,
main = "Дендрограма для кластеризації за методом Варда")
```

Рисунок 4.4.2- дендограма кластеризації методом Варда на 3 кластери

Інформацію, після кластеризації методом Варда на 3 кластери, щодо належності до певного кластеру додамо до базового датасету.

```
life_data.clust <- cbind(life_data, as.factor(life_data.clust2w))
head(life_data.clust)
```

```
   endocrine eye_diseases ear_diseases respiratory_diseases digestive_diseases
17     26062         47377        34658               552633              43546
5      10382         33027        25074               366305              23103
1      33026        158758       116143              1200543             111601
10     13395         43436        39485               450605              42879
4      11830         32261        20548               381388              34647
18     20389         44227        23539               346649              52799
   genitourinary_diseases as.factor(UA_data.clust2w)
17                  55446                           1
5                   40701                           1
1                  270413                           1
10                  67605                           1
4                   51127                           1
18                  29900                           1
```

4. 5 K-mean кластеризація (метод к-середніх)

Ітераційні (роздільні) алгоритми є методикою кластеризації, за якої множина об’єктів розділяється на k груп, причому кількість груп має бути вказана до початку кластеризації.

Найбільш популярним алгоритмом цієї групи є метод k-середніх (k-means clustering), в якому кожний кластер представлений його центром (центроїдом), координати якого є середнім арифметичним координат точок, які належать відповідному кластеру. За такого способу підрахунку координат центру метод стає чутливим до викидів. На вхід вказаних алгоритмів треба задати кількість кластерів, на які будуть розбиватися всі об’єкти, та початкові центри кластерів.

4. 5. 1 Метод ліктя

Для визначення оптимальної кількості кластерів розроблено більше 30 методів. Одним з основних є метод ліктя, який спирається на порівняння внутрішньокластерної суми квадратів відстаней (WSS), для зручності ця величина зображується як функція від кількості кластерів.

```
fviz_nbclust(life_data, kmeans, method = "wss") +
  geom_vline(xintercept = 2, linetype = 2) +
  labs(subtitle = "Метод ліктя")
```

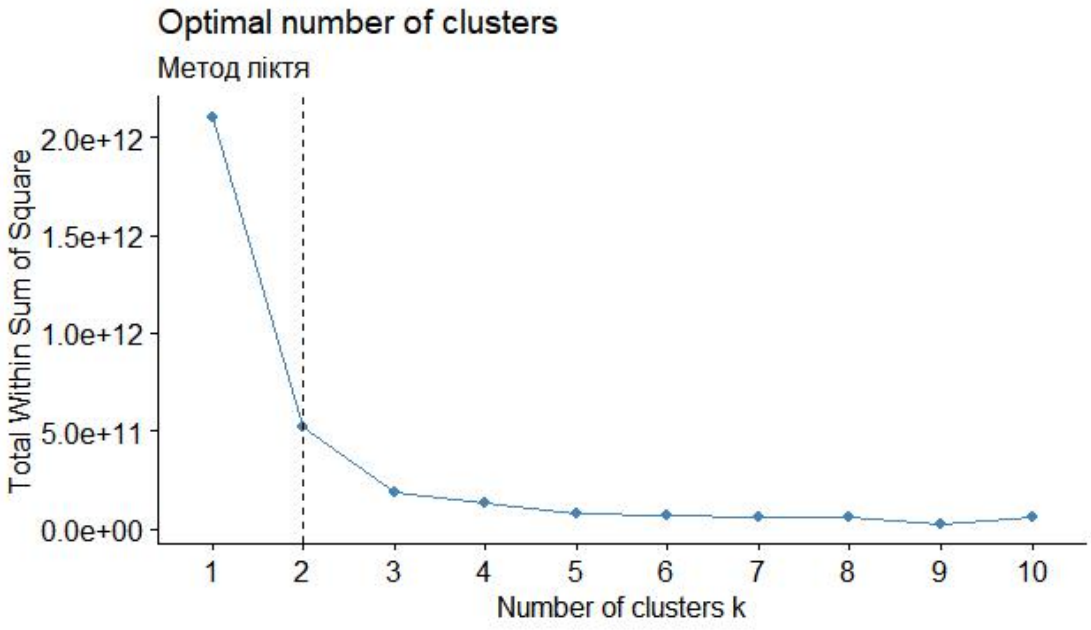

Рисунок 4.5.1.1- візуалізація методу ліктя

Точка “ліктя”, в який досягається останнє суттєве зменшення внутрішньокластерної суми квадратів, може бути оптимальною кількістю кластерів, оскільки подальше збільшення кількості кластерів не покращить суттєво розбиття.

В даному випадку оптимальною кількістю кластерів є 2.

4. 5. 2 Метод середнього силуету (Silhouette method)

Метод середнього силуету дещо інакше характеризує якість кластеризації, і визначає, наскільки “вдало” кожна точка розміщена в своєму кластері. Оптимальною кількістю кластерів є така, що максимізує середнє значення силуету.

```
fviz_nbclust(life_data, kmeans, method = "silhouette") +
  labs(subtitle = "Метод середнього силуету")
```

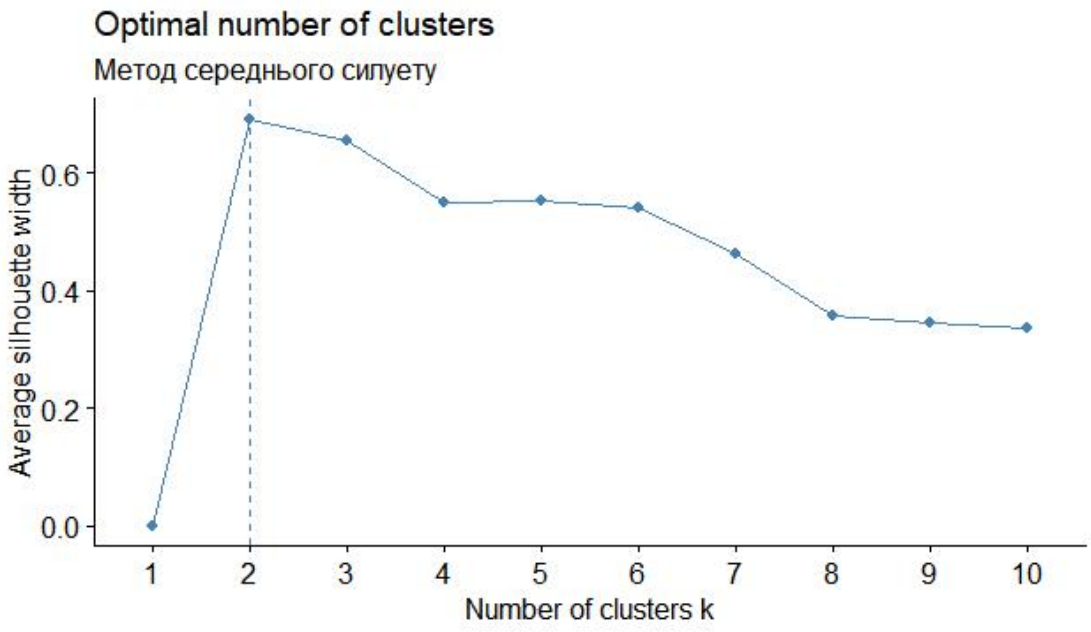

Рисунок 4.5.2.1- візуалізація методу середнього силуету

Згідно з методом середнього силуету оптимальною кількістю кластерів є 2.

Можна зобразити діаграму силуету (silhouette plot), яка зображує значення силуету для кожної точки та середнє значення силуету. Додатнє значення силуету означає, що об’єкт вдало розміщений (у “правильному” кластері), причому чим ближче значення до 1, тим краще він розміщений. Якщо значення силуету від’ємне, то об’єкт знаходиться не в тому кластері. Якщо значення силуету рівне нулю, то об’єкт знаходиться на межі між двома кластерами.

```
life_data.kmeans2 <- kmeans(life_data, centers = 2, nstart = 20)
sil2 <- silhouette(life_data.kmeans2$cluster, dist(life_data))
fviz_silhouette(sil2)
plot(sil2)
```

```
  cluster size ave.sil.width
1       1   20          0.76
2       2    5          0.43
```

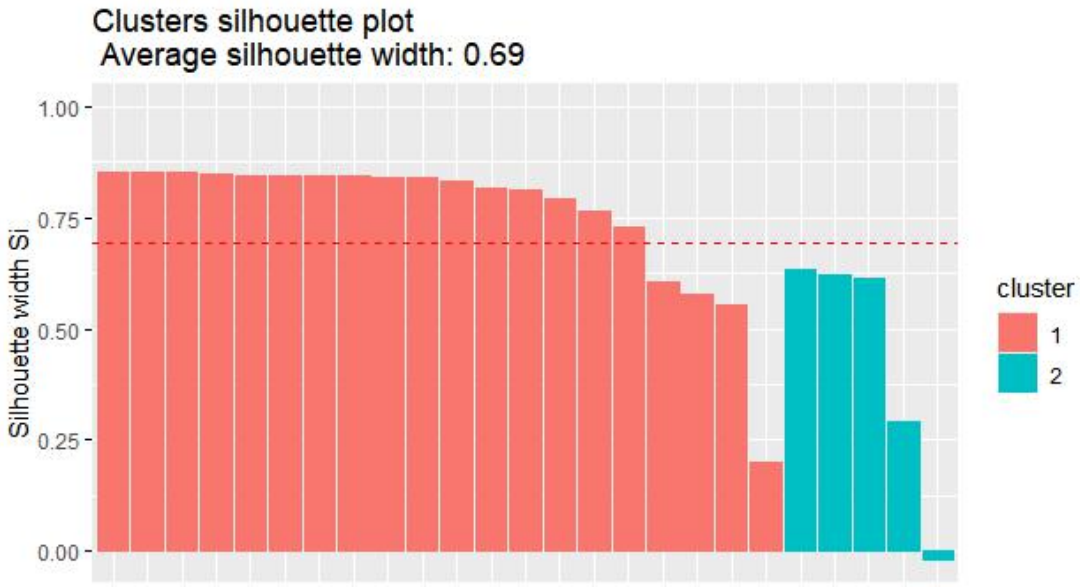

Рисунок 4.5.2.2- візуалізація діаграми силуету для 2 кластерів

Інформацію по застосованому методу силуету можна вивести наступним чином:

```
summary(sil2)
```

```
Silhouette of 25 units in 2 clusters from silhouette.default(x = UA_data.kmeans2$clus
ter, dist = dist(UA_data)) :
 Cluster sizes and average silhouette widths:
       20         5
0.7576302 0.4287022
Individual silhouette widths:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.01984  0.61258  0.81180  0.69184  0.84525  0.85430
```

Для трьох кластерів маємо:

```
life_data.kmeans3 <- kmeans(life_data, centers = 3, nstart = 20)

Sil3 <- silhouette(life_data.kmeans3$cluster, dist(life_data))

fviz_silhouette(sil3)
```

```
  cluster size ave.sil.width
1       1   16          0.72
2       2    6          0.50
3       3    3          0.63
```

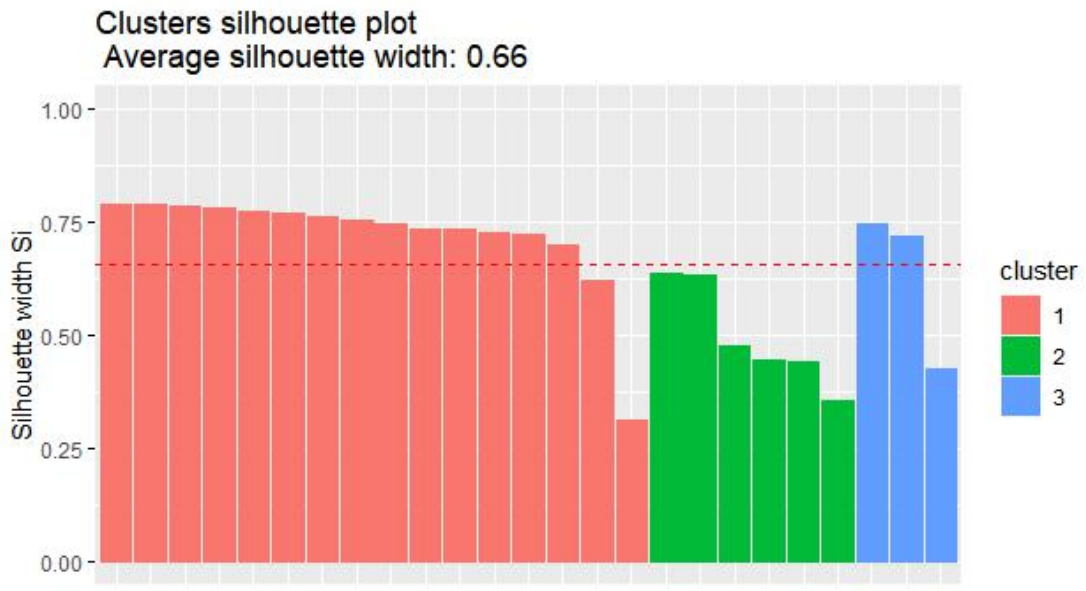

Рисунок 4.5.2.3- візуалізація діаграми силуету для 3 кластерів

Тут бачимо що діаграма силуету для 3 кластерів виглядає краще.

4. 5. 3 Алгоритм на основі консенсусу

Існує доволі багато різних методів та чисельних характеристик, які допомагають визначити оптимальну кількість кластерів.

Оскільки немає методу, який однозначно вірно визначає оптимальну кількість кластерів, хорошою ідеєю є запустити кілька з них та обрати кількість кластерів, яка найкраще узгоджується (тобто знайти консенсус).

```
n_clust <- n_clusters(life_data, package = c("easystats", "NbClust", "mclust"))

n_clust
```

```
The choice of 2 clusters is supported by 12 (40.00%) methods out of 30 (Elbow, Silhouette, Ch, CCC, DB, Ratkowsky, PtBiserial, Mcclain, Dunn, SDindex, Mixture (EVE), Mixture (VVE)).
```

```
plot(n_clust)
```

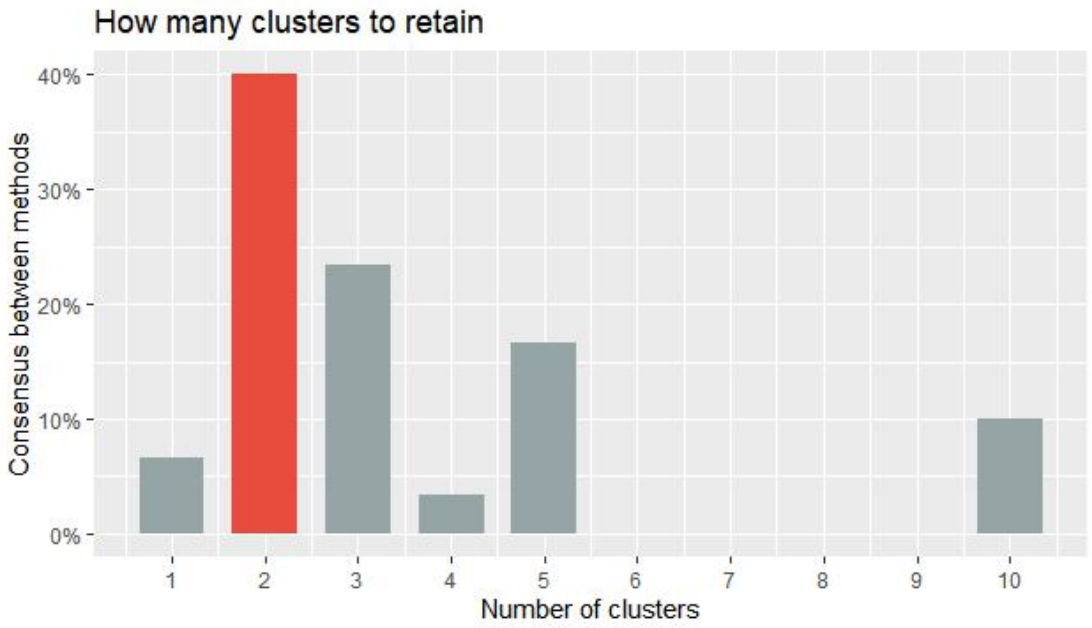

Рисунок 4.5.3.1- результат роботи алгоритму на основі консенсусу

Отже два, мабуть, буде найкращою кількістю кластерів для методу к-середніх.

4. 5. 4 **Процедура кластеризації методом k-середніх**

Сама процедура кластеризації реалізується в R функцією kmeans(), основним аргументом є датасет, аргументом center задається кількість кластерів, на які буде розбиватися сукупність об’єктів, або вручну задаються центри кластерів.

За замовчуванням алгоритм k-середніх використовує випадкові точки в якості центрів стартових кластерів. Як наслідок можемо мати різні результати кластеризації для різних запусків алгоритму та дещо різну якість кластеризації (яка визначається зокрема відношенням міжкластерної суми квадратів до загальної суми квадратів).

```
life_data.kmeans2 <- kmeans(life_data, centers = 2, nstart = 10)

100 * life_data.kmeans2$betweenss / life_data.kmeans2$totss
```

```
## [1] 75.17089
```

```
life_data.kmeans2$cluster
```

Вінницька область	1 кластер
Волинська область	1 кластер
Дніпропетровська область	2 кластер
Донецька область	1 кластер
Житомирська область	1 кластер
Закарпатська область	1 кластер
Запорізька область	1 кластер
Івано-Франківська область	1 кластер
Київська область	1 кластер
Кіровоградська область	1 кластер
Луганська область	1 кластер
Львівська область	2 кластер
Миколаївська область	1 кластер
Одеська область	2 кластер
Полтавська область	1 кластер
Рівненська область	1 кластер
Сумська область	1 кластер
Тернопільська область	1 кластер
Харківська область	2 кластер
Херсонська область	1 кластер
Хмельницька область	1 кластер
Черкаська область	1 кластер

Чернівецька область	1 кластер
Чернігівська область	1 кластер
м.Київ	2 кластер

```
life_data_clust <- data.frame(life_data, cluster = as.factor(life_data.kmeans2$cluster))

head(life_data_clust)
```

```
  endocrine eye_diseases ear_diseases respiratory_diseases digestive_diseases
1     26062        47377        34658               552633              43546
2     10382        33027        25074               366305              23103
3     33026       158758       116143              1200543             111601
4     13395        43436        39485               450605              42879
5     11830        32261        20548               381388              34647
6     20389        44227        23539               346649              52799
  genitourinary_diseases cluster
1                  55446       1
2                  40701       1
3                 270413       2
4                  67605       1
5                  51127       1
6                  29900       1
```

4. 6 **Візуалізація даних**

Проводимо кластеризацію методом k-середніх за допомогою функції eclust() та зображуємо отримані результати кластеризації в площині двох головних компонент. Ця функція може проводити кластеризацію багатьма методами.

```
life_data.eclust.km2 <- eclust(life_data, "kmeans", k = 2, nstart = 25)
```

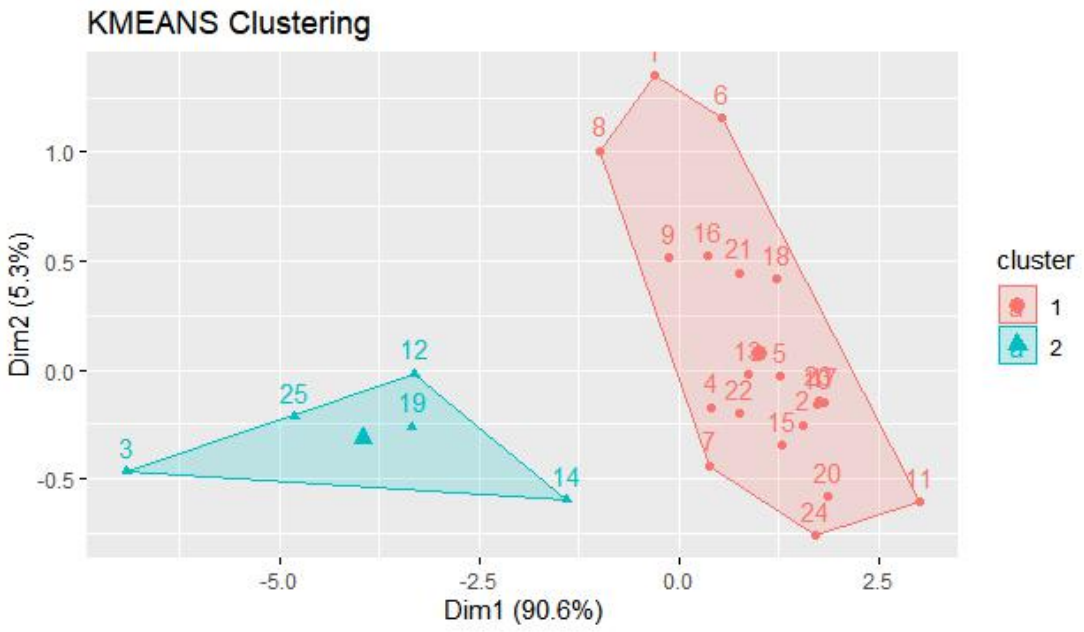

Рисунок 4.6.1- візуалізація кластеризації методом k-середніх для 2 кластерів

Спробуємо пояснити, чому утворилися саме такі кластери. Для цього ми виконаємо ще 3 команди:

```
df <-  as.data.frame(life_data.kmeans2$centers)

sapply(seq(1,6),function(x)  which.max(df[,x]))
```

[1] 2 2 2 2 2 2

```
sapply(seq(1,6),function(x)  which.min(df[,x]))
```

[1] 1 1 1 1 1 1

Перша команда визначає центроїди кластерів та створює на їхній основі дата фрейм. Друга команда визначає по кожній із характеристик той центроїд, який має максимальне значення. Це означає, що значення однієї з характеристик буде максимальним серед представників того кластеру. Аналогічно, третя команда визначає по кожній із характеристик мінімум. Отже :

- У представників першого кластеру, а саме Вінницька, Волинська, Донецька, Житомирська, Закарпатська, Запорізька, Івано-Франківська, Київська, Кіровоградська, Луганська, Миколаївська, Полтавська, Рівненська, Сумська, Тернопільська, Херсонська, Хмельницька, Черкаська, Чернівецька та Чернігівська області найвищі показники за всіма ознаками.
- У представників другого кластеру, а саме Дніпропетровська, Львівська, Одеська, Харківська області та м.Київ найнижчі показники за всіма ознаками.

4. 7 Якість кластеризації

Оскільки об’єкти всередині кожного кластера мають бути максимально близько один від одного та максимально далеко від об’єктів з інших кластерів, то саме ці два фактори так чи інакше оцінюються різними індексами для визначення якості кластеризації та утворюють групу внутрішніх показників якості.

- Компактність кластера показує, наскільки близько знаходяться об’єкти всередині кластерів. Індикатор хорошої компактності – внутрішньокластерна дисперсія. Показники, що характеризують компактність базуються на відстанях між об’єктами. Мають бути мінімальними.
- Роздільність кластерів визначає, наскільки добре розділяються кластери. Індекси використовують дистанції між центрами кластерів та/або попарні мінімальні дистанції між об’єктами з різних кластерів.

Більшість індексів так чи інакше використовують пропорцію між характеристиками компактності та роздільності з певними ваговими коефіцієнтами.

```
dd <- dist(life_data, method ="euclidean")
km_stats <- cluster.stats(dd, life_data.kmeans4$cluster)
km_stats$within.cluster.ss
```

[1] 522433499149 - сума квадратів кластерів

```
km_stats$clus.avg.silwidths
```

```
        1         2
0.7576302 0.4287022
```

```
km_stats
```

```
$n
[1] 25

$cluster.number
[1] 2

$cluster.size
[1] 20  5

$min.cluster.size
[1] 5

$noisen
[1] 0

$diameter
[1] 494533.1 520577.2

$average.distance
[1] 139442.8 310607.4

$median.distance
[1] 112856.6 280569.0

$separation
[1] 87987.76 87987.76

$average.toother
[1] 631475.2 631475.2

$separation.matrix
         [,1]     [,2]
[1,]     0.00 87987.76
[2,] 87987.76     0.00
```

```
$ave.between.matrix
          [,1]     [,2]
[1,]      0.0 631475.2
[2,] 631475.2      0.0

$average.between
[1] 631475.2

$average.within
[1] 173675.7

$n.between
[1] 100

$n.within
[1] 200

$max.diameter
[1] 520577.2

$min.separation
[1] 87987.76

$within.cluster.ss
[1] 522433499149

$clus.avg.silwidths
        1         2
0.7576302 0.4287022

$avg.silwidth
[1] 0.6918446

$g2
NULL


$g3
NULL

$pearsongamma
[1] 0.8069829

$dunn
[1] 0.1690196

$dunn2
[1] 2.033034

$entropy
[1] 0.5004024

$wb.ratio
[1] 0.2750317

$ch
[1] 69.63319

$cwidegap
[1] 101151.9 246311.9

$widestgap
[1] 246311.9

$sindex
[1] 87987.76

$corrected.rand
NULL

$vi
NULL
```

4. 8 **Критерії якості**

Мета – порівняти кластеризації, отримані за допомогою методу К-середніх та ієрархічної кластеризації.

```
life_data.km1 <- eclust(life_data, "kmeans", k = 2, nstart = 25, grap
h = FALSE)

table(life_data.clust2w$cluster, life_data.kmeans2$cluster)
```

```
w  1  2
1 18  5
2  2  0
```

```
clust_stats <- cluster.stats(d = dist(life_data), life_data.clust2w$cluste
r,
                                        life_data.kmeans2$cluster)

# Зкоригований індекс Ренда

clust_stats$corrected.rand
```

```
##[1] 0.7882728
```

```
      clust_stats$vi
```

```
##[1] 0.304878
```

Виправлений індекс Ренда забезпечує міру оцінки подібності між двома розділами, з поправкою на випадковість. Його діапазон становить від -1 (без узгодження) до 1 (ідеальна згода). Узгодження між типами видів і кластерним рішенням становить 0.788 з використанням індексу Ренда та 0.305 за допомогою критерію варіації інформації, він вимірює кількість інформації, втраченої та отриманої в результаті зміни методу кластеризації. Досягнутий рівень варіації інформації 0.305 означає, що варіація між двома вибраними кластерами низька, і вони мають гарну схожість.

Висновки.

У даній роботі було реалізовано процес кластеризації за даними з демографічної таблиці захворювань органів в Україні в 2016 році за такими ознаками: хвороби ендокринної системи, хвороби вуха, хвороби ока та придаткового апарату, хвороби органів дихання, хвороби органів травлення та хвороби сечостатевої системи.

Для проведення експериментів була написана комп'ютерна програма, що дозволяє проводити кластеризацію за вказаними ознаками.

Поставлена мета досягнута – демографічні дані кластерізовано. Отримано 2 кластери, кластерізовані за такими ознаками:

- У типового представника першого кластеру а саме Вінницької, Волинської, Донецької, Житомирської, Закарпатської, Запорізької, Івано-Франківської, Київської, Кіровоградської, Луганської, Миколаївської, Полтавської, Рівненської, Сумської, Тернопільської, Херсонської, Хмельницької, Черкаської, Чернівецької та Чернігівської областей найвищі показники за всіма ознаками.
- У типового представника другого кластеру а саме Дніпропетровської, Львівської, Одеської, Харківської областей та в м.Київ найнижчі показники за всіма ознаками. Однак варто зауважити, що рівень захворюваності в великих містах може бути нижчим не за рахунок хорошої екології, а за рахунок кращого медичного обслуговування з можливістю профілактики та ранньої діагностики.

Було використано ієрархічну кластеризацію методом Варда та метод к-середніх. Отже роботу виконано, дані було прокластеризовано. Виокремлено області України з високим а саме

Вінницька, Волинська, Донецька, Житомирська, Закарпатська, Запорізька, Івано-Франківська, Київська, Кіровоградська, Луганська, Миколаївська, Полтавська, Рівненська, Сумська, Тернопільська, Херсонська, Хмельницька, Черкаська, Чернівецька та Чернігівська області та більш низьким рівнем захворюваності а саме Дніпропетровська, Львівська, Одеська, Харківська області та м.Київ.

Список використаних джерел

1. Cluster Analysis,5th Edition,Brian S. Everitt . Sabine Landau,Morven Leese . Daniel Stahl,King’s College London, UK
2. Practical Guide To Cluster Analysis in R, Edition 1, Unsupervised Machine Learning
3. Classification via clustering for predicting final marks based on student participation in forums, M.I. López, J.M Luna, C. Romero, S. Ventura, Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain
4. The Use of Cluster Analysis in Typological Research on Community College Students, Peter Riley Bahr, Rob Bielby, Emily House
5. Мандель И.Д. Кластерный анализ — М.: Финансы и статистика, 1988 — 176 с.
6. Лагутин М. Б. Наглядная математическая статистика. — М.: П-центр, 2003.
7. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999.
8. Jain, Murty, Flynn Data clustering: a review. // ACM Comput. Surv. 31(3), 1999.
9. Журавлев Ю. И., Рязанов В. В., Сенько О. В. Распознавание. Математические методы. Программная система. Практические применения. — М.: Фазис, 2006.
10. Шуметов В. Г. Шуметова Л. В. Кластерный анализ: подход с применением ЭВМ.
11. Мандель И. Д. Кластерный анализ. — М.: Финансы и статистика, 1988.
12. B.S. Everitt, S. Landau, M. Leese (2001) Cluster Analysis
13. Kaufman, Leonard, and Peter J Rousseeuw. 2009. Finding Groups

in Data: An

14. Introduction to Cluster Analysis. Vol. 344. John Wiley & Sons.
15. Hair, Joseph F. 2006. Multivariate Data Analysis. Pearson Education India.
16. Kassambara Alboukadel, Practical Guide to Cluster Analysis in R. STHDA, 2017.
17. A. Broder, R. Kumar, F. Maghoul et al., “Graph structure in the web,” Computer Networks, vol. 33, no. 1–6, pp. 309–320, 2000.
18. A. R. Anaya and J. G. Boticario. Content-free collaborative learning modeling using data mining. User Modeling and User-Adapted Interaction, Springer, 2011.
19. Ammon, B. V., Bowman, J., and Mourad, R. “Who Are Our Students? Cluster Analysis as a Tool for Understanding Community College Student Populations.” Journal of Applied Research in the Community College, 2008.
20. Bahr, P. R. “The Bird’s Eye View of Community Colleges: A Behavioral Typology of First-Time Students Based on Cluster Analytic Classification.” Research in Higher Education, 2010.
21. Майборода Р. Комп’ютерна статистика: професійний старт - 2017 р.
22. Майборода Р., Сугакова О. “Аналіз даних за допомогою пакета R” 2015 р.
23. Ward 1963.

Додаток А. Текст програми

Текст програми

```
library(dplyr)
library(factoextra)
library(NbClust)
library(cluster)
library(parameters)
library(fpc)
library(FactoMineR)
library(devtools)
library(corrplot)
library(psych)
library(GPArotation)
life_data <- read.csv("Lifet.csv") %>%
   mutate(ind = paste(Year, Age, sep=' '))
row.names(life_data) <- life_data$ind
life_data <- life_data[-c(1,2,11)]
glimpse(life_data)

summary(life_data)

life_data.dist <- get_dist(life_data, stand = TRUE)
fviz_dist(life_data.dist, gradient = list(low = "green2", mid = "white",
high = "darkblue"))

life_sample_ind <- sample(seq(1, nrow(life_data)), size = 30)
life_data_smpl <- life_data[life_sample_ind,]
head(life_data_smpl)
```

```
scale_life <- scale(life_data_smpl)

life_data.hc <- hclust(life_data.dist, method = "ward.D2")    #
кластеризація
fviz_dend(life_data.hc)
fviz_dend(life_data.hc, k = 4, # розділяємо на 4 кластери
                cex = 0.5, # розмір шрифту для назв об'єктів
                k_colors = c("magenta", "turquoise3", "orange", "tomato3"),
                color_labels_by_k = TRUE,
                rect = TRUE,
                main = "Дендрограма для кластеризації за методом
Варда")
fviz_dend(life_data.hc, k = 2,
                cex = 0.5,
                k_colors = c("olivedrab", "black"),
                color_labels_by_k = TRUE,
                rect = TRUE,
                main = "Дендрограма для кластеризації за методом
Варда")
life_data.clust <- cbind(life_data, as.factor(life_data.clust2w))
head(life_data.clust)

fviz_nbclust(life_data, kmeans, method = "wss") +
   geom_vline(xintercept = 2, linetype = 2) +
   labs(subtitle = "Метод ліктя")
fviz_nbclust(life_data, kmeans, method = "silhouette") +
   labs(subtitle = "Метод середнього силуету")

life_data.kmeans2 <- kmeans(life_data, centers = 2, nstart = 20)
```

```
sil2 <- silhouette(life_data.kmeans2$cluster, dist(life_data))
fviz_silhouette(sil2)
plot(sil2)
summary(sil2)
life_data.kmeans3 <- kmeans(life_data, centers = 3, nstart = 20)
Sil3 <- silhouette(life_data.kmeans3$cluster, dist(life_data))
fviz_silhouette(sil3)

life_data.kmeans4 <- kmeans(life_data, centers = 4, nstart = 20)
sil4 <- silhouette(life_data.kmeans4$cluster, dist(life_data))
fviz_silhouette(sil4)
n_clust <- n_clusters(life_data, package = c("easystats", "NbClust",
"mclust"))
n_clust

plot(n_clust)
life_data.kmeans3 <- kmeans(life_data, centers = 3, nstart = 10)
100 * life_data.kmeans3$betweenss / life_data.kmeans3$totss
life_data.kmeans3$cluster

life_data_clust <- data.frame(life_data, cluster =
as.factor(life_data.kmeans3$cluster))
head(life_data_clust)
life_data.eclust.km2 <- eclust(life_data, "kmeans", k = 2, nstart = 25)
life_data.eclust.km3 <- eclust(life_data, "kmeans", k = 3, nstart = 25)
df <- as.data.frame(kmeans.red$centers)
sapply(seq(1,8),function(x) which.max(df[,x]))
sapply(seq(1,8),function(x) which.min(df[,x]))
sapply(seq(1,8),function(x) which.min(abs(df[,x])))
```

```
dd <- dist(life_data, method ="euclidean")
km_stats <- cluster.stats(dd,    life_data.kmeans4$cluster)
km_stats$within.cluster.ss
km_stats$clus.avg.silwidths
km_stats
life_data.km1 <- eclust(life_data, "kmeans", k = 2, nstart = 25, graph =
FALSE)
table(life_data.km1$cluster, life_data.pam$cluster)

clust_stats <- cluster.stats(d = dist(life_data), life_data.pam$cluster,
                                          life_data.km1$cluster)
# Зкоригований індекс Ренда
clust_stats$corrected.rand
     clust_stats$vi
head(var$cos2, 8)
corrplot(var$cos2, is.corr=FALSE)
corrplot(var$cos2[ ,c(1:3)])
fviz_cos2(my_life_data.pca, choice = "var", axes = 1:2)
fviz_pca_var(my_life_data.pca, col.var = "cos2",
                    gradient.cols = c("blue", "green", "red"),
                    repel = TRUE
)
head(var$contrib, 8)
corrplot(var$contrib, is.corr=FALSE)



fviz_contrib(my_life_data.pca, choice = "var", axes = 1)
fviz_contrib(my_life_data.pca, choice = "var", axes = 2)
fviz_contrib(my_life_data.pca, choice = "var", axes = 1:2)
```

```
fviz_pca_var(my_life_data.pca, col.var = "contrib",
                    gradient.cols = c("turquoise3", "yellow", "red") )
```

Додаток Б. Датасет

Датасет

Ukraine	endocrine	eye_diseases	ear_diseases	respiratory_diseases	digestive_diseases	genitourinary_diseases
Vinnytsia	26062	47377	34658	552633	43546	55446
Volyn	10382	33027	25074	366305	23103	40701
Dnipropetrovsk	33026	158758	116143	1200543	111601	270413
Donetsk	13395	43436	39485	450605	42879	67605
Zhytomyr	11830	32261	20548	381388	34647	51127
Transcarpathian	20389	44227	23539	346649	52799	29900
Zaporizhzhia	12967	52608	38930	559493	28741	70898
Ivano-Frankivsk	24038	63309	46003	533796	64963	68176
Kyiv_obl	19760	48853	36741	658832	40118	57898
Kirovohrad	9973	34850	18987	298407	27036	35004
Luhansk	4088	15019	10386	168947	9379	18581
Lviv	23753	115503	82063	1029572	75510	101134
Mykola	1293	35716	31510	320515	42562	63256

yiv	7					
Odesa	15052	83163	60090	721014	60298	99413
Poltava	9743	35436	23427	391180	33281	53815
Rivne	17708	55048	31292	368905	46552	49168
Sumy	10017	28380	21274	262000	25409	37349
Ternopil	15621	33505	25323	354097	29147	37258
Kharkiv	21299	111210	85563	785688	93828	129733
Kherson	7513	32436	25376	240623	21393	46604
Khmelnytsky	16324	36799	33790	348085	41386	44214
Cherkasy	12913	51117	35799	376775	32352	55672
Chernivtsi	10267	34915	19909	268416	25706	36949
Chernihiv	6256	37041	27968	382303	22191	36033
Kyiv	28622	107351	93094	1214962	86140	204499